INTRODUCTION TO PROGRAMMING USING VISUAL BASIC 2008, AN (W/VS2008 DVD) 7TH EDITION PDF, EPUB, EBOOK

Schneider, David, David I. | 0136060722 | 9780136060727

David I. Schneider: List of Books by Author David I. Schneider

Introduction to Programming Using Visual Basic 2008, An (w/VS2008 DVD) : United

States Edition

3.17 (12 ratings by Goodreads)

By (author) David I. Schneider

Description

Now in its Seventh Edition, Schneider's "tried and true" text - consistently praised by both students and instructors - is designed for students with no prior computer programming experience. Now updated for Visual Basic 2008, Schneider focuses on teaching problem-solving skills and sustainable programming skills. A broad range of real-world examples, section-ending exercises, case studies, and programming projects gives students significant hands-on experience.

Supplements:

Instructor: Solutions, Power Point Lecture Slides, Test Files

Student: Source Code

www.prenhall.com/schneider
show more

Product details

- Mixed media product 8

  ## PROGRAMMING IN VISUAL BASIC 2008

  Pemrograman merupakan salah satu mata kuliah dalam kurikulum Program Studi Manajemen Informatika AMIK Indonesia Banda Aceh. Mata kuliah ini bertujuan agar mahasiswa mampu memahami konsep dasar pemrograman windows dan mampu menerapkan konsep pemrograman Visual BASIC dalam mengembangkan perangkat lunak berbasis windows. Sesuai dengan tujuan tersebut, buku ajar ini tersusun atas beberapa bagian yakni dimulai dari pengenalan .NET dan Framerwok, kemudian pada bagian berikutnya dibahas mengenai Visual Studio 2015, dasar pemrograman Visual BASIC, basis data dalam pemrograman Visual BASIC, Microsoft Office Acces 2016, SQL Server 2015, Crytal Report dan Compiler. Buku ajar ini ditujukan kepada khalayak yang ingin memahami mengenai konsep pemrograman BASIC dan pengembangan perangkat lunak berbasis Windoes, terutama bagi mahasiswa Program Studi Manajemen Informatika AMIK Indonesia Banda Aceh. Buku ajar ini diharapkan dapat menjadi acuan untuk membantu mahasiswa dalam perkuliahan. Buku ajar ini dapat diselesaikan dengan baik berkat dukungan dari berbagai pihak. Penulis dalam kesempatan ini ingin menyampaikan ucapan terima kasih kepada semua pihak dan terutama kepada AMIK Indonesia Banda Aceh. Semoga buku ajar ini dapat bermanfaat bagi semua pihak yang membutuhkan. Penulis menyadari, dalam penulisan buku ajar ini masih banyak terdapat kekurangan. Penulis sangat mengharapkan saran yang bersifat membangun demi kesempuranaan buku ajar ini.

  Источник: https://www.academia.edu/11436009/PROGRAMMING_IN_VISUAL_BASIC_2008
  2 4 5 4 768 pages
- 203 x 254 x 30.48mm

e99.com Bookstore

Page 10 181-188 of 188 Back

An introduction to programming using Visual Basic 2008: with Visual Studio Expression edition DVD [7th ed] 9780136060727, 2008023532, 0136060722

Table of contents :

Citation preview

TO

AN INTRODUCTION PROGRAMMING USING ®

VISUAL BASIC 2008 with Visual Studio® Express Edition DVD

SEVENTH EDITION

This page intentionally left blank

TO

AN INTRODUCTION PROGRAMMING USING ®

VISUAL BASIC 2008 with Visual Studio® Express Edition DVD

SEVENTH EDITION

David I. Schneider University of Maryland

Upper Saddle River, NJ 07458

Library of Congress Cataloging-in-Publication Data Schneider, David I. An introduction programming using Visual Basic 2008 / David I. Schneider. —7th ed. p. cm. Includes index. ISBN 978-0-13-606072-7 1. Microsoft Visual BASIC. 2. BASIC (Computer program language) I. Title. QA76.73.B3S3336 2008 005.2'762—dc22 2008023532

Editorial Director, Computer Science and Engineering: Marcia J. Horton Executive Editor: Tracy Dunkelberger Editorial Assistant: Melinda Haggerty Director of Marketing: Margaret Waples Marketing Manager: Christopher Kelly Associate Editor: ReeAnne Davies Senior Managing Editor: Scott Disanno Production Editor: Irwin Zucker Art Director: Kenny Beck Cover Art: Franklin Hammond/Images.com Art Editor: Gregory Dulles Manufacturing Manager: Alan Fischer Manufacturing Buyer: Lisa McDowell

Pearson Education Ltd., London Pearson Education Singapore, Pte. Ltd. Pearson Education Canada, Inc. Pearson Education—Japan Pearson Education Australia PTY, Limited Pearson Education North Asia Ltd., Hong Kong Pearson Educación de Mexico, S.A. de C.V. Pearson Education Malaysia, Pte. Ltd. Pearson Education Inc., Upper Saddle River, New Jersey 10 9 8 7 6 5 4 3 2 1 ISBN 10: 0-13-606072-2 ISBN 13: 978-0-13-606072-7

CONTENTS

◆

◆

This page intentionally left blank

PREFACE

Since its introduction in 1991, Visual Basic has been the most widely used programming language in the world. The latest incarnation of Visual Basic, called Visual Basic 2008, brings the language into the Internet age by incorporating the .Net Framework. Visual Basic programmers are enthusiastically embracing the new features of the language. Likewise, students who are learning their first programming language will find VB 2008 the ideal tool to understand the development of computer programs. My objectives when writing this text were as follows: 1.

To develop focused chapters. Rather than covering many topics superficially, I concentrate on important subjects and cover them thoroughly.

2.

To use examples and exercises that students can relate to, appreciate, and feel comfortable with. I frequently use real data. Examples do not have so many embellishments that students are distracted from the programming techniques illustrated.

3.

To produce compactly written text that students will find both readable and informative. The main points of each topic are discussed first, and then the peripheral details are presented as comments.

4.

To teach good programming practices that are in step with modern programming methodology. Problem solving techniques and structured programming are discussed early and used throughout the book. The style follows object-oriented programming principles.

5.

To provide insights into the major applications of computers.

Unique and Distinguishing Featur es Exercises for Most Sections. Each section that teaches programming has an exercise set. The exercises both reinforce the understanding of the key ideas of the section and challenge the student to explore applications. Most of the exercise sets require the student to trace programs, find errors, and write programs. The answers to all the odd-numbered exercises in Chapters 2 through 8 and selected odd numbered exercises from Chapters 9, 10, and 11 are given at the end of the text. Practice Problems. Practice Problems are carefully selected exercises located at the end of a section, just before the exercise set. Complete solutions are given following the exercise set. The practice problems often focus on points that are potentially confusing or are best appreciated after the student has worked on them. The reader should seriously attempt the practice problems and study their solutions before moving on to the exercises. Programming Projects. Beginning with Chapter 3, every chapter contains programming projects. The programming projects not only reflect the variety of ways that computers are used in the business community, but also present some games and general-interest topics. The large number and wide range of difficulty of the programming projects provide the flexibility to adapt the course to the interests and abilities of the students. Some programming projects in later chapters can be assigned as end-of-the-semester projects.

Comments. Extensions and fine points of new topics are deferred to the “Comments” portion at the end of each section so that they will not interfere with the flow of the presentation. Case Studies. Each of the four case studies focuses on an important programming application. The problems are analyzed and the programs are developed with top-down charts and pseudocode. The programs can be found on the companion website at www.prenhall.com/schneider. Chapter Summaries. In Chapters 2 through 11, the key results are stated and the important terms are summarized at the end of the chapter. Arrays. Arrays are introduced gently in two sections. The first section presents the basic definitions and avoids procedures. The second section presents the techniques for manipulating arrays and shows how to pass arrays to procedures. “How To” Appendix. Appendix B provides a compact, step-by-step reference on how to carry out standard tasks in the Visual Basic environment. Appendix on Debugging. Placing of the discussion of Visual Basic’s sophisticated debugger in Appendix D allows the instructor flexibility in deciding when to cover this topic. Appendix on Converting from Visual Basic 6.0 to Visual Basic 2008. This appendix is included primarily to assist the instructor who is familiar with VB 6.0 and is new to using VB 2008. Companion Website. The companion website (http://www.prenhall.com/schneider) contains all the examples, case studies, and data files referenced in the book. (Details can be found on page xv). Instructor Supplements. In addition to the materials on the companion website, solutions to every exercise and programming project, a test item file for each chapter, and PowerPoint lecture slides for each chapter are available to instructors. Contact your local Pearson sales representative for information on how to access these resources.

What’s New in the Seventh Edition 1.

The version of Visual Basic has been upgraded from Visual Basic 2005 to Visual Basic 2008 and relevant new features of Visual Basic 2008 have been incorporated.

2.

Suggestions from students and reviewers have been incorporated as much as possible.

3.

Chapter 1 has been shortened considerably and combined with Chapter 2. (Some of the material from the previous Chapter 1 has been moved to the Appendices.)

4.

Chapter 3 has been split into two chapters.

5.

Chapter 4 (General Procedures) of the previous edition has been moved to follow the chapter on decision structures.

6.

The concept of scope is discussed earlier in the book.

7.

The real-life data in the examples and exercises have been updated and revised.

8.

Text files are dealt with inside the Visual Basic environment instead of with Notepad.

9.

All screen captures have been updated to show the Microsoft® Windows Vista operating system.

10.

Message boxes are invoked with MessageBox.Show instead of MsgBox.

ACKNOWLEDGMENTS

Many talented instructors, students, and programmers provided helpful comments and constructive suggestions during the preparation of this text. For their contributions to the quality of the first six editions of the book I extend my gratitude to A. Abonomah, University of Akron; Timothy Babbitt, Rochester Institute of Technology; William Barnett, Northwestern State University; Sherry Barriclow, Grand Valley State University; Robert Berman, Wayne State University; William Burrows, University of Washington; David Chao, San Francisco State University; Christopher Chisolm, University of Nebraska, Omaha; Robert Coil, Cincinnati State Technical and Community College; Gary Cornell, University of Connecticut; Ronit Dancis; John DaPonte, Southern Connecticut State University; Ward Deutschman, Briarcliff; Ralph Duffy, North Seattle Community College; Charles Fairchild; Pat Fenton, West Valley College; David Fichbohm, Golden Gate University; Robert Fritz, American River College; Matthew Goddard, New Hampshire Technical College; Mickie Goodro, Casper College; Wade T. Graves, Grayson Community College; Christine Griffin; Gary Haw, MIPS Software Development Inc.; Shelly Hawkins, Microsoft; Tom Janicki, Kent State University; Dana Johnson, North Dakota State University; Dan Joseph, Rochester Institute of Technology; Del Kimber, Clemson University; Wanda Kunkle, Rowan College; Paul Lecoq, San Francisco Community College; David Leitch, DeVry Institute; David Letcher, The College of New Jersey; Kieran Mathieson, Oakland University; Charlie Miri, Delaware Tech; George Nezlek, DePaul University; Ron Notes, Hebrew Academy of Greater Washington; Mike Paul, Berry University; T. S. Pennington, Maple Woods Community College; Arland Richmond, Computer Learning Center; David Rosser, Essex County College; Arturo Salazar, San Francisco State; Susanne Peterson, Microsoft; Janie Schwark, Microsoft; Mike Talber, Portland Community College; Steve Turek, DeVry Institute of Technology, Kansas City; Jac Van Deventer, Washington State University; Randy Weinberg, St. Cloud State University; Laurie Werner, Miami University; Melinda, White, Santa Fe Community College; Ronald Williams, Central Piedmont Community College; Amit Kalani, CIStems Solutions LLC; Priti Kalani, MobiEcast Corporation; Chris Panell, Heald College; Kevin Parker, Idaho State University; TJ Racoosin, rSolutions; and Bill Tinker, Aries Software. The current edition benefited greatly from the valuable comments of the following knowledgeable reviewers: Paul Bladek, Edmonds Community College (WA) Jason Blazzard, Lewis Clark State College & Washington State Douglas Bock, Southern Illinois University Joe Brady, University of Delaware Peter Chan, Canyon College (CA) Debra Chapman, University of South Alabama Chakib Chraibi, Barry University Ronald Greenberg, Loyola University of Chicago Dennis Higgins, SUNY- Oneonta Duke Hutchins, Bowling Green State University Stephen Klein, Ramapo College (NJ) Kathleen Lynch, University of Wisconsin (Oshkosh) Joe Mast, Eastern Mennonite University Tom McCollough, Hillsborough CC (FL) Jim McKeown, Dakota State University

Acknowledgments Josh Pauli, Dakota State University Markita Price, University of Missouri Paula Ruby, Arkansas State University Sara Rushinek, University of Miami Cheryl Schlitter, University of Colorado - Colorado Springs Syd Shewchuk, Heald College Debbie Tesch, Xavier University Sangsuree Vasupongayya, Portland State University Keith Weber, Idaho State University Many people are involved in the successful publication of a book. I wish to thank the dedicated team at Pearson whose support and diligence made this textbook possible, especially Marcia Horton, Editor-in-Chief, and Scott Disanno, Senior Managing Editor. I am grateful to Jeremy Schneider, Paul Mailhot, and Zane Lang for their careful proofreading. Production editor Irwin Zucker did a fantastic job producing the book and keeping it on schedule. I extend special thanks to my editor Tracy Dunkelberger. Her ideas and enthusiasm helped immensely with the preparation of the book.

THE COMPANION WEBSITE FOR STUDENTS AND INSTRUCTORS How to Access the Companion Website All the programs from the examples, case studies, text files for the exercises, database files and BMP images can be downloaded by students and instructors at www.prenhall.com/schneider.

Students The companion website, located at www.prenhall.com/schneider, contains all the programs from the examples and case studies set forth in this textbook, all the text files needed for the exercises, all databases needed for the exercises, and several picture files. All these files are contained in the folders "Ch02", "Ch03", "Ch04", and so on. Each chapter file contains a subfolder named "Text_Files_for_Exercises" which contains the text files needed for that chapter's exercises. The folder "Ch09" has a subfolder named "Pictures" that contains the picture files. The folder "Ch10" has a subfolder named "MajorDatabases" containing all the databases needed for the exercises. Each program is contained in a folder with a name in the form chapter–section–number. For instance, the program in Chapter 3, Section 1, Example 2 is contained in the folder "3-1-2". Many of the programs make use of a text file in the subfolder Debug of the program's folder named bin.

Instr uctors Instructor resources including solutions to the exercises, PowerPoint lecture slides, all the example programs and data files used by students throughout the book, and multiple-choice and true/false questions. Contact your local Pearson sales representative to gain access to this material.

Notice: This book contains many screen captures. When you run one of the programs downloaded from the website, what you see on your monitor might not look exactly like the screen capture shown in the book. To make them appear the same, you must check that your monitor is set to display 96 DPI (Dots Per Inch). To determine and/or change the DPI setting for your monitor, see the first item under "Utilize the Windows Environment" in Appendix B on pages 608-9.

USING THIS BOOK FOR A SHORT OR CONDENSED COURSE This book provides more than enough material for a complete semester course. The topics must be trimmed for courses lasting considerably less than a full semester. The following syllabus provides one possible way to present an abbreviated introduction to programming.

Chapter 1 An Introduction to Computers and Problem Solving 1.1 An Introduction to Computers 1.2 Windows, Folders, and Files Chapter 2 2.1 2.2 2.3

Visual Basic, Controls, and Events An Introduction to Visual Basic 2008 Visual Basic Controls Visual Basic Events

Chapter 3 3.1 3.2 3.3

Variables, Input, and Output Numbers Strings Input and Output

Chapter 4 4.1 4.2 4.3

Decisions Relational and Logical Operators If Blocks Select Case Blocks

Chapter 5 5.1 5.2 5.3

General Procedures Sub Procedures, Part I Sub Procedures, Part II Function Procedures

1

1.1

An Introduction to Computers

An Introduction to Programming Using Visual Basic 2008 is a book about problem solving using computers. The programming language used is Visual Basic 2008 (hereafter shortened to Visual Basic), but the principles taught apply to many modern programming languages. The examples and exercises present a sampling of the ways that computers are used in the real world. Here are some questions that you might have about computers and programming. Question: What are the main components of a personal computer? Answer: Hidden from view inside the system unit are several components, including the microprocessor, memory, and hard drive of the computer. The central processing unit (CPU), sometimes referred to as the microprocessor, can be thought of as the computer's brain, which carries out all of the computations. The memory, often referred to as random access memory (RAM), stores instructions and data while they are being used by the computer. When the computer's power is turned off, the contents of memory are lost. A hard disk drive is used to store instructions and data when they are not being used in memory and when the computer is turned off. Inside the system unit there are also device cards, such as a graphics card, sound card, network card, and a modem. A graphics card is used to send an image to the monitor, and a sound card is used to send audio to a set of speakers attached to the computer. Network cards can be used to connect to a local area network (LAN) of computers, while a modem uses a telephone line or cable to connect to another computer. The personal computer also has several input and output devices, which are used to communicate with the computer. Standard input devices include the keyboard and mouse. Standard output devices include the monitor and printer. Instructions are entered into the computer by typing them on the keyboard, clicking a mouse, or loading them from a file located on a disk drive or downloaded from a network. Information processed by the computer can be displayed on the monitor, printed on the printer, or recorded on a disk drive. Question: What are some topics covered in this text that students could use immediately? Answer: Computer files can be created to hold lists of names, addresses, and phone numbers, which can be alphabetized and displayed in their entirety or selectively. Mathematical computations can be carried out for science, business, and engineering courses. Personal financial transactions, such as bank deposits and loans, can be recorded, organized, and analyzed. Line charts, pie charts, and bar charts can be created to enhance the data in a term paper. Question: How do we communicate with the computer? Answer: Many languages are used to communicate with the computer. At the lowest level, there is machine language, which is understood directly by the microprocessor, but is awkward for humans. Visual Basic is an example of a higher-level language. It consists of instructions to which people can relate, such as Click, If, and Do. Question: How do we get computers to perform complicated tasks? Answer: Tasks are broken down into a sequence of instructions that can be expressed in a computer language. (This text uses the language Visual Basic.) This sequence of instructions is called a program. Programs can range in size from two or three instructions to millions of instructions. Instructions are typed on the keyboard, or read in from a file

on a disk, and stored in the computer's memory. The process of executing the instructions is called running the program. Question: Are there certain features that all programs have in common? Answer: Most programs do three things: take in data, manipulate them, and give desired information. These operations are referred to as input, processing, and output. The input data might be held in a portion of the program, reside on a disk drive, or be provided by the computer operator in response to requests made by the computer while the program is running. The processing of the input data occurs inside the computer and can take from a fraction of a second to many hours. The output data are either displayed on the monitor, printed on the printer, or recorded on a disk. As a simple example, consider a program that computes sales tax. An item of input data is the cost of the thing purchased. The processing consists of multiplying the cost by a certain percentage. An item of output data is the resulting product, the amount of sales tax to be paid. Question: What are the meanings of the terms “hardware” and “software?” Answer: Hardware refers to the physical components of the computer, including all peripherals, the central processing unit, disk drives, and all mechanical and electrical devices. Programs are referred to as software. Question: What are the meanings of the terms “programmer” and “user?” Answer: A programmer is a person who solves problems by writing programs on a computer. After analyzing the problem and developing a plan for solving it, he or she writes and tests the program that instructs the computer how to carry out the plan. The program might be run many times, either by the programmer or by others. A user is any person who uses a program. While working through this text, you will function both as a programmer and as a user. Question: What is meant by problem solving? Answer: Problems are solved by carefully reading them to determine what data are given and what outputs are requested. Then a step-by-step procedure is devised to process the given data and produce the requested output. This procedure is called an algorithm. Finally, a computer program is written to carry out the algorithm. Algorithms are discussed in Section 1.4. Question: What types of problems are solved in this text? Answer: Carrying out business computations, creating and maintaining records, alphabetizing lists, and displaying tabular data are some of the types of problems we will solve. Question: When did the concept of computers and programming first appear? Answer: The first general-purpose computer, called the Analytical Engine, was described by Charles Babbage in 1837. Although it was never built, its logical design was essentially modern. (In 1939, the first general-purpose computer was actually built by John Atanasoff.) In 1843, Augusta Ada Byron showed how to carry out certain complex mathematical computations with the Analytical Engine, thereby establishing herself as the first computer programmer. For a historical perspective of the evolution of modern computers and programming languages, see Appendix E, A Biographical History of Computing.

Question: How did Visual Basic 2008 evolve? Answer: In the early 1960s, two mathematics professors at Dartmouth College developed BASIC to provide their students with an easily learned language that could tackle complicated programming projects. As the popularity of BASIC grew, refinements were introduced that permitted structured programming, which increases the reliability of programs. Visual Basic 1.0 is a version of BASIC developed in 1991 by the Microsoft Corporation to allow easy, visual-oriented development of Windows applications. Visual Basic 2008 is a language similar to the original Visual Basic, but more powerful. It is targeted for what is known as the .NET run time, which is a program that executes Visual Basic 2008 as well as programs from other languages that are targeted for the .NET run time. This will ultimately allow programs written in Visual Basic to be run on devices other than computers, such as cell phones and handheld devices. Other features of Visual Basic 2008 include full object-oriented programming capabilities and the development of Web services. Object-oriented programming is discussed in Chapter 11. Question: Are there any prerequisites to learning Visual Basic 2008? Answer: Since Visual Basic is used to write Windows applications, you should be familiar with Windows and understand how folders and files are managed with Windows. The key concepts are presented succinctly in Section 1.2 and discussed in detail in Appendix F. Question: Will it matter whether I use Windows XP or Windows Vista as the operating system? Answer: Visual Basic runs fine with either Windows XP or Windows Vista. However, the screens will vary in appearance. Figure 1.1 shows how the appearance of a Visual Basic program differs with the operating system. In this book, all screens have the Windows Vista appearance.

(a) Windows XP FIGURE 1.1

1.2

(b) Windows Vista

A Visual Basic window.

Windows, Folders, and Files

This preliminary section presents some terms used in this book.

■ Windows and its Little Windows Windows gets its name from the way it organizes the screen into rectangular regions. When you run a program, the program runs inside a bordered rectangular box.

Unfortunately Windows jargon calls these windows, so there’s only a lowercase “w” to distinguish them from the operating system called Windows. Figure 1.2 shows the window that results from running one of the programs in this book. In Visual Basic terminology, such a window is also called a form.

Title bar

Minimize button

FIGURE 1.2

Maximize button

Close button

A Visual Basic window.

■ Mouse Actions Hover: Linger the mouse at a particular place and wait for a message (such as a tool tip) to appear. Drag an object: Move the mouse pointer until it is at the object, press the left mouse button and hold it down, move the mouse pointer until the object moves to where you want it to be, and finally, release the mouse button. (Sometimes this whole activity is called drag-and-drop.) Right-click: Press and release the right mouse button once. Click: Press and release the left mouse button once. (sometimes referred to as singleclick or left-click) Double-click: Click the left mouse button twice in quick succession. Note: An important Windows convention is that clicking selects an object so you can give Windows further directions about it, but double-clicking tells it to perform a default operation. For example, double-clicking on a folder will open that folder.

■ Files and Folders (A detailed discussion of files and folders can be found in Appendix F.) Disk: Either the hard disk, a diskette, a CD, or DVD. Each disk drive is identified by a letter followed by a colon.

◆

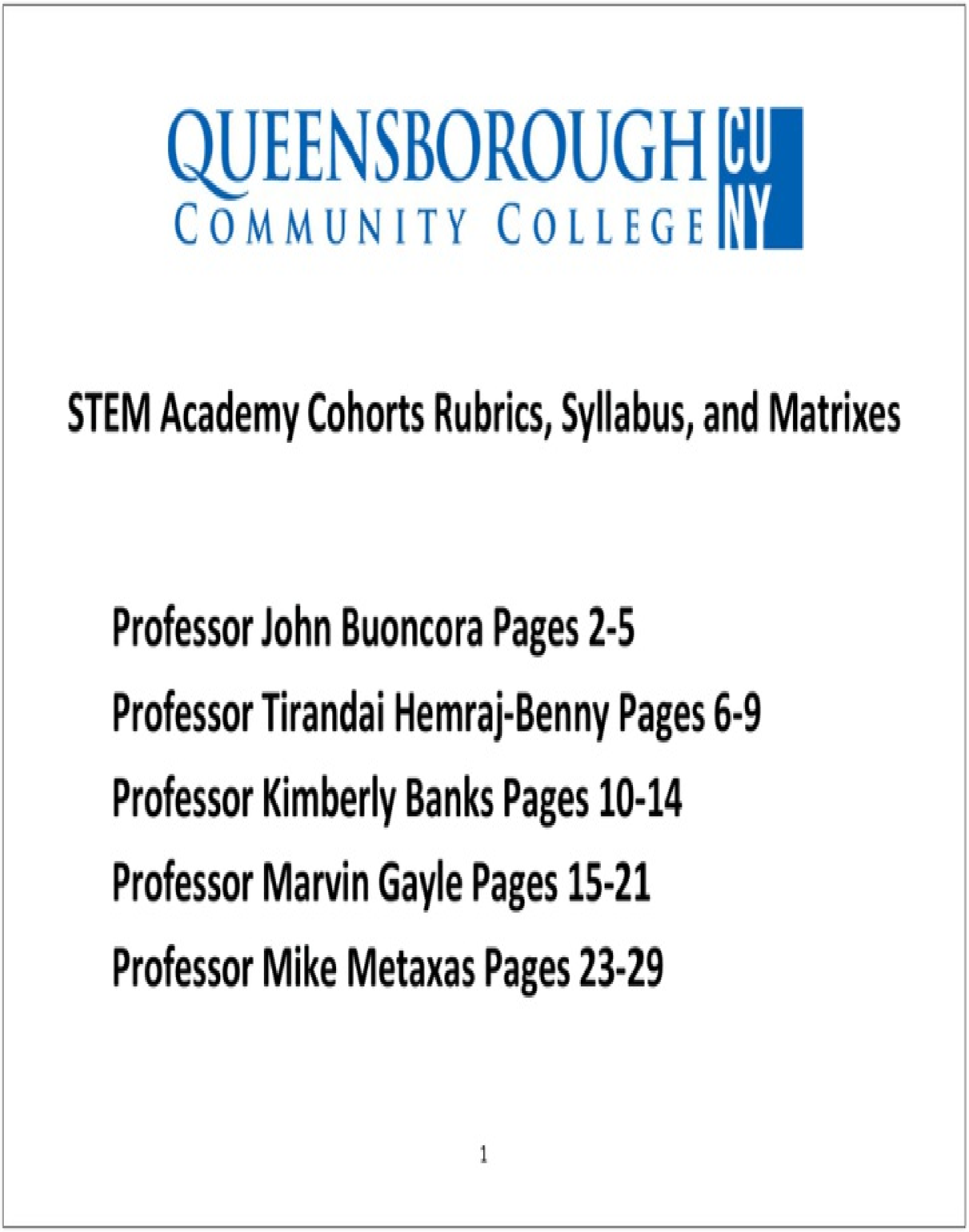
5

6

◆

Chapter 1 An Introduction to Computers and Problem Solving

File: Either a program file or a data file. Its name typically consists of letters, digits, and spaces. The name of the file is also called the base name. Extension of a file name: One or more letters, preceded by a period, that identify the type of file. For example, files created with Word have the extension doc or docx. Filename: The combination of the base name, the period, and the extension. The only characters that cannot be used in filenames are \, /, :, *, ?, , ", and 5 6 7



QUEENSBOROUGH COMMUNITY COLLEGE CUNY

STEM Academy Cohorts Rubrics, Syllabus, and Matrixes

Professor John Buoncora Pages 2-5

Professor Tirandai Hemraj-Benny Pages 6-9

Professor Kimberly Banks Pages 10-14

Professor Marvin Gayle Pages 15-21

Professor Mike Metaxas Pages 23-29

1

STEM A

Academmy Coh horts R Rubrics,, Syllab bus, an nd Matrrixes Professor John B Buonco ora Pagges 2-5 dai Hemmraj-Be enny Paages 6--9 Professor Tirand erly Ban nks Pagges 10--14 Professor Kimbe n Gayle e Pagess 15-211 Professor Marvin Metaxaas Page es 23-229 Professor Mike M 1 Assessment for STEM Academy – Scoring Rubrics Instructor: John Buoncora Course: ET 110 (Electrical Circuit Analysis I) General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Use the varied forms of mathematical communication: language, symbolic notation, graphs, charts. Assessment Plan and Scoring Rubrics: Assessment Plan &amp; Scoring Rubrics 0 (Lowest score) 1 2 3 4 (Highest Score) a: Only able to find the equivalent (total) resistance of the entire circuit of the problem. b: a) plus able to find the source current or voltage, and able to find "a few" of the resistor currents or voltages in the problem. c: b) plus able to find all of the currents, voltages and powers for the problem. d: c) and able to check the results using Kirchhoff's Laws and conservation of energy in the problem. ET 110 Exam 2 Problem 4: The student is unable to find the total resistance of Competency/ Exercise the circuit. Description: Analyze a series-parallel resistive circuit with a single DC voltage source using equivalent resistances, KVL, KCL, and Ohm's Law. 2 General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Application of mathematics to appropriate fields of study. Assessment Plan and Scoring Rubrics: Assessment Plan &amp; 0 (Lowest score) 1 2 3 4 (Highest Score) Scoring Rubrics ET 110 Exam 3 Problem The student is unable 1: to write any of the Competency/ mesh (loop) equations. Exercise Description: Use Mesh (Loop) analysis to analyze a circuit containing multiple voltage sources and/or current sources. a: Could only write the mesh (loop) equations in a partially correct manner for the problem. b: a) plus able to write all of the mesh equations correctly for the problem. c: b) plus able to partially solve the mesh equations and solve for at least one quantity correctly in the problem. d: c) plus able to completely solve the mesh equations and correctly solve for all quantities in the problem. Able to check all of the problem results on the original circuit diagram using KVL, KCL, and Ohms' Law. 3 General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make informed decisions. Student Learning Outcome/Competency: Differentiate between facts, assumptions, and conclusions in the formulation of a proposed solution or answer. Compare the way questions, issues or problems are formulated within various fields of study. Assessment Plan and Scoring Rubrics: Assessment Plan &amp; Scoring Rubrics ET 110 Lab 8 (Series- Parallel Circuits &amp; In- Circuit Resistance Measurement): Competency/ Exercise description: Construct Series-Parallel resistive circuits on a breadboard and apply a DC voltage source. Measure equivalent circuit resistances with power off and measure voltages and currents with power on. Calculate the theoretical values of all circuit quantities and simulate the circuit operation. 0 (Lowest score) 1 The student is unable a: Student was able to to construct any of construct only the the circuits. "simple" series-parallel circuit and the wiring may have contained errors. Assessment for STEM Academy – Summary Rubric Score Sheet 4 2 3 4 (Highest Score) b: a) without any errors and student was able to measure all circuit voltages with power applied. The student was also able to measure equivalent circuit resistances with power off. c: b) and the student was able to measure all circuit currents without blowing a fuse in the DMM. Student was also able to construct the "more complicated" series- parallel circuit. The student was able to determine if a circuit was operating within specifications as compared to the calculated values. d) and the student was able to measure all voltages and currents in "more complicated" series- parallel circuits without any issues. The student was able to calculate all circuit currents and voltages in both series-parallel circuits and simulate all circuits successfully. The student was able to compare measured and calculated values using %differences and explain the differences. Instructor: John Buoncora Course: ET 110 (Electrical Circuit Analysis I) Competency General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Use the varied forms of mathematical communication: language, symbolic notation, graphs, charts. Competency/Exercise Description: Analyze a series-parallel resistive circuit with a single DC voltage source using equivalent resistances, KVL, KCL, and Ohm's Law. General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Application of mathematics to appropriate fields of study. Competency/Exercise Description: Use Mesh (Loop) analysis to analyze a circuit containing multiple voltage sources and/or current sources. General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make informed decisions. Student Learning Outcome/Competency: Differentiate between facts, assumptions, and conclusions in the formulation of a proposed solution or answer. Compare the way questions, issues or problems are formulated within various fields of study. Competency/Exercise description: Construct Series-Parallel resistive circuits on a breadboard and apply a DC voltage source. Measure equivalent circuit resistances with power off and measure voltages and currents with power on. Calculate the theoretical values of all circuit quantities and simulate the circuit operation. 0 (Lowest Score) 0 The number of students in each category is indicated in the table above. Notes: The first and second rows correspond to an ET 110 Lecture sections, with 34 students assessed. The third row corresponds to an ET 110 Lab sections, with 17 students assessed. 5 0 0 1 2 3 2 4 3 1 3 2 9 8 5 4 (Highest Score) 19 20 9 Assessment for STEM Academy Instructor: Tirandai Hemraj-Benny Course: CH-121 – Fundamentals of Chemistry Laboratory General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Used the varied forms of mathematical communication: language, symbolic notation, graphs, charts, to formulate quantitative ideas and patterns. Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Lab Report #4: The student is unable to 1 2 The student correctly assigns each axis but The student is able to plot the data but cannot does not understand that draw complete conclusions. Axes are each axis should start correctly assigned and Student will perform assign each axis, does from zero and even divided but not labeled. an experiment to not label axes and does divisions should follow. investigate the not understand that each Thus, correct conclusions cannot be drawn. relationship between axis should start from zero and even divisions distance from a should follow. radioactive source and Properties of Natural Radioactivity Description: degree of radioactive signal. Student will represent their data on a graph and draw conclusions. plot the data and draw conclusions. Student does not correctly 6 3 4 The student is able to plot the data but cannot draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. The student is able to plot the data and draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. Quiz #2: Description Student will be given radioactive data to plot on a graph and draw conclusions after graded lab reports have been returned. The student is unable to plot the data and draw conclusions. Student does not correctly The student correctly assigns each axis but does not understand that each axis should start from zero and even assign each axis, does divisions should follow. not label axes and does Thus, correct conclusions not understand that each cannot be drawn. axis should start from zero and even divisions should follow. The student is able to plot the data but cannot draw complete conclusions. Axes are correctly assigned and divided but not labeled. The student is able to plot the data but cannot draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. Assessment Data: # of students The student is able to plot the data and draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. Lab Report 0 3 1 2 2 2 3 1 4 0 2 3 3 0 4 4 Lab Quiz # of students 0 1 1 0 7 Instructor: Tirandai Hemraj-Benny Course: CH-120 – Fundamentals of Chemistry Lecture General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Identify problems that need a mathematical solution, and use computational methods in the mathematics applicable in everyday life. Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Exam #4: The student is not aware Question #4 that mathematical calculation is necessary Description: Student will be asked and cannot determine the oxidation numbers of to assign the any element. oxidation number of each element in compounds 1 2 3 4 The student is aware that mathematical calculation is necessary but cannot determine the oxidation numbers of any of the elements correctly. The student is aware that mathematical calculation is necessary and determines the oxidation numbers of some of the elements correctly. The student is aware that mathematical calculation is necessary and determines the oxidation numbers of most of the elements correctly. The student is aware that mathematical calculation is necessary and determines the oxidation numbers of all elements correctly. Assessment Data: # of students 0 0 1 2 2 3 8 3 6 4 0 Instructor: Tirandai Hemraj-Benny Course: CH-120 – Fundamentals of Chemistry Lecture General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make an Informed decisions. Specific Objective (Outcome): Assessment Plan and Rubrics: Assessment Plan 0 &amp; Rubrics EXAM #4: The student does not identify the specific Question #3 topic or problem, Description: where they are unable Students will be to predict the products asked to provide formed and represent the molecular, ionic and net ionic the reaction in the equations when a three types of equations. carbonate and a nitrate compound are mixed together. Assessment Data: # of students 0 2 Distinguish the problem or question from a proposed solution. 1 2 4 The student identifies the specific topic or problem, where they are able to predict the products formed but cannot represent the reaction in the three types of equations. The student identifies the specific topic or problem, where they are able to predict the products formed and can represent the reaction in the three types of equations to some extent. The student identifies the specific topic or problem, where they are able to predict the products formed and can represent the reaction in the three types of equations with a few mistakes. The student identifies the specific topic or problem, where they are able to predict the products formed and represent the reaction in the three types of equations correctly. 1 1 2 3 9 3 3 5 4 0 Instructor: Course: Kimberly Banks EN-101 English Composition CUNY Proficiency Examination Task 1 Scoring Guide A. Develops an essay that presents a focused response to the writing assignment, making appropriate and coherent connections among all parts of the assignment. B. Demonstrates understanding of the readings through summary and explanation of relevant material. C. Incorporates, as support for own thoughts, references to the readings, identifying the sources formally or informally. 6 Addresses the writing assignment fully, analytically, and perhaps critically or imaginatively, with superior focus and coherence. 6 Demonstrates superior and perhaps critical understanding of readings through accurate summary, full explanation, and insightful analysis of relevant sections. 5 Addresses the writing assignment fully and analytically, with strong focus and coherence. 4 Addresses all parts of the writing assignment with adequate focus and coherence throughout. 5 Demonstrates strong understanding of readings through accurate summary, with appropriate explanation and analysis of relevant sections. 6 Makes insightful connections and distinctions between readings and own ideas; integrates references smoothly into own essay and identifies them consistently and correctly. 5 Makes analytical connections and perhaps distinctions between readings and own ideas; integrates references into own essay and identifies them consistently and correctly. 4 Demonstrates overall understanding of readings through appropriate summary and explanation, with some analysis. 4 Makes and explains appropriate connections between readings and own ideas; identifies references consistently and correctly. 6 Communicates with precision and enhanced expression through highly effective use of vocabulary and sentence variety; infrequent, if any, lapses in use of conventions. 5 Communicates effectively throughout the essay, with few lapses in use of conventions. 10 D. Communicates clearly and effectively, using appropriate conventions of language (e.g., grammar, spelling, punctuation). 4 Communicates clearly throughout the essay; sentences may contain some lapses in use of conventions, but these rarely impede comprehension. 3 Addresses all or most parts of the writing assignment adequately, but focus may lapse briefly or connections may be missing. 3 Demonstrates generally accurate understanding of readings although summary or explanation may be incomplete or not fully relevant. 2 Addresses some parts of the writing assignment or addresses all parts superficially; focus or coherence may break down at several points. 2 Demonstrates partial understanding of the readings through summary or explanation, but understanding is flawed or explanation is incomplete. 1 Shows little or no ability to address the writing assignment; may not link thoughts between paragraphs. 3 Makes some connections between readings and own ideas but they may not all be appropriate or adequately explained; identifies most references consistently and correctly. 3 Generally communicates clearly throughout the essay although lapses in use of conventions may at times impede comprehension or prove distracting. 2 Makes few or unwarranted connections between readings and own ideas; may identify references inconsistently or incorrectly. 2 Communicates clearly at times, showing some ability to use conventions, but whole sections are unclear or errors frequently impede comprehension. 1 Demonstrates little or no understanding of text. 1 Makes no reference to background reading or makes no distinctions between background reading and own ideas. 11 1 Communicates little because few sentences demonstrate appropriate use of conventions. CPE Quick Summary Initial Proficiencies Scores A 6 3 5 1 4 4 3 8 2 6 1 0 B 2 2 5 5 8 0 C 0 0 1 2 6 13 D 0 0 2 9 4 5 B 4 2 4 10 1 0 C 2 7 6 5 2 0 D 1 4 6 8 3 0 Final Proficiencies Scores A 6 3 5 5 4 5 3 6 2 2 1 0 12 Prof. Kimberly J. Banks EN 101 Composition I, section D13 11/23/09 In-Class Essay #2 Using the selection from Henry I. Miller and Gregory Conko's The Frankenfoood Myth: How Protest and Politics Threaten the Biotech Revolution and Carol Tucker Foreman's "Killing the 'Frankenfood' Monster: How People Can Love, Not Fear, Biotech Food," discuss the advantages and disadvantages associated with biotechnology. In your essay, summarize Miller and Conko's key points distinguishing between the myths and realities of biotechnology. Draw a strong relationship between Miller and Conko's thinking and what you have just read about the need to reassure the public in Tucker Foreman's article. In light of the reading selections, discuss your own knowledge of or attitudes towards biotechnology. What experiences have you had with biotechnology and have they made your life better or worse? Also discuss the degree to which your perspective reflects the ideas of either or both writers. 13 In-Class Essay #2 CONTENT (75%) To what extent does the student address the assignment? How focused and coherent is the answer? Points Earned To what extent does the student demonstrate understanding of the reading? At what level does the student summarize, explain, and analyze evidence? How regularly does the student distinguish between his/her ideas and those of the authors in the readings? How well are the references integrated into the essay? To what extent are the references identified? How clear and effective is the essay? How well are paragraphs and sentences structured? MECHANICS (25%) What is the size and appropriateness of the student's vocabulary? To what extent are words spelled appropriately? How effective is the student's use of grammar? How appropriate and adequate is the punctuation to effectively help convey meaning? Total 14 Points Possible 25 25 25 10 5 5 5 100 Assessment for STEM Academy Instructor: Marvin Gayle Course: ET-502 – Introduction to Programming General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Application of mathematics to appropriate fields of study. Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Lab Experiment:: The student is unable to start the Lab_#7 software tool and Description: setup the Visual Student write a Visual basic program that takes Basic Software environment IDE. two inputs Resistance and voltage and then develops a software program to calculate and output the current to and appropriate number of digits. 1 2 3 4 The student is able to start the software tool and setup the Visual Basic Software environment IDE but is unable to read the user inputs without an error. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to read the user inputs but unable to display the appropriate answer. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to read the user inputs, and able to display the appropriate answer, but user interface is not balanced and appears unbalanced The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to read the user inputs, and able to display the appropriate answer, but user interface is not balanced and appear well balanced. 15 Instructor: Marvin Gayle Course: ET-502 – Introduction to Programming General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make informed decisions. Specific Objective (Outcome): Differentiate between facts, assumptions, and conclusions in the formulation of a proposed solution or answer. Compare the way questions, issues or problems are formulated within various fields of study Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Lab Experiment:: The student is unable to Lab_#9 start the software tool and setup the Visual Description: Student writes a Visual basic program that will be used to display ONLY the numbers that are perfect Squares, within a given range of numbers. Basic Software environment IDE. 1 2 3 4 The student is able to start the software tool and setup the Visual Basic Software environment IDE but is unable to develop a correct algorithm to get to an end solution The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to develop a correct algorithm but unable to successfully implement it. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to develop a correct algorithm, is able to successfully implement it, but is unable to decide in the implementation which results to display at the end of the program. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to develop a correct algorithm, is able to successfully implement it, and is able to properly display the results to display at the end of the program. 16 P Professor Marvvin Gayle ET – 502 Introduction to Proggramming O Office Location: Technology Bu uilding Room T-2 20 TTelephone: 718 8-631-6207 EEmail: mgayle@ @qcc.cuny.edu C Course Descripttion: IIntroduction to the Visual BASIC P Programming lan nguage with application problems in electrical an d Computer Engineering Technollogy; with h hands-on experie ence in the Department's Computer Center. Hours Prerequisites Corequisites General Educatio on Objectives Course Objectives 1 Cred dit, 3 Laboratoryy Hours ET–501 Computer Applications Demonstrate masteryy of discipline-sp pecific knowledge, skills and toools required for entry into or advanceme ent in the job m market in their fi eld (caree er programs) Use an nalytical reasoning skills and ap pply logic to solvve problems This course introducees the student to fundamental prrogramming concep pts and structurees such as decision making and d loops. They w will need th hese skills for any a problem solv ving (algorithm)) and for future coursees like ET503 an nd ET560. Textbooks Introduction to Programming Using Visual Basic 2008" (w/VS2008 DVD D), 7th Edition Author David I. Schn neider Publish her Prenticce Hall ISBN ISBN-10: 013606 60722, ISBN-13: 978013 36060727 17 Department of Electrical and Computer Engineering Technology ET-502 – Introduction to Programming Fall 2009 ET_502 COURSE OUTLINE Prerequisite: ET-501 Week No Topic Reading Introduction to Blackboard. Overview of computers and computer languages: applications programming, high level/low level, compiled/interpreted. Problem solving and flowcharting. Class notes pp. 1 – 18 pp. 19 – 24 Appendix B 1 How to VB2008 Express Edition, saving and naming forms and projects. 2 Introduction to the Visual Basic Environment: Objects, Properties. Internal Documentation (The Rem statement). pp. 19 – 43 pp. 85 – 86 Visual Basic Events: Using code to change Properties of Objects. pp. 44 - 54 3 4 pp. 61 - 70 Arithmetic Operations: Level of Precedence, Numeric Variables, Print method, Numeric Functions. Declaring Variable Types. 5 Strings: String Variables, Concatenation, Scope of Variables, Using Text Boxes for input (Val and Str functions), Using the Format function with the Print method. pp. 77 - 88 pp 95 – 104 6 Mid-Term Examination Relational Operators and String Relationships And as time permits: [The ANSI (or ASCII) Character Set. String and String Related Functions (Left, Mid, Right, UCase, Trim, Len, and InStr)] 7 Using a Message Box for Output General Procedures, Sub Procedures, Variables and Expressions as 18 p. 121 – 126 pp. 104 – 106 Arguments, Passing Variables Back from Sub Procedures. 8 Review of Relational Opreators and String Relationships pp. 121 – 127 Controlling Program Flow: If Blocks, Logical Operators pp. 129 - 136 9 Decision Structures (Select Case Blocks) pp. 146 - 154 10 Looping Structures: Do Loops and For…Next Loops pp. 237 – 242 pp 267 - 275 11 Class Note PBASIC – FOR…NEXT loop and LEDs 12 Introduction to the BASIC Stamp: Controlling Outputs Class Notes PBASIC – FOR…NEXT loop and LEDs 13 Introduction to the BASIC Stamp: Reading Switch Inputs 14 Final Examination Class Notes Note: It is important to read the "Comments" appearing at the conclusion of each section. 19 Course Outcomes Outcomes Outcome 1 Description an appropriate mastery of the knowledge, techniques, skills and modern tools of their disciplines Outcome 2 an ability to identify, analyze and solve technical problems Outcome 3 an ability to communicate effectively Laboratory Experiments Lab Week_1 Week_2 Week_3 Week_4 Week_5 Week_6 Week_7 Week_8 Week_9 Week_10 Week_11 Week_12 &amp; Week_13 Week_14 Topics Overview of computers and computer languages:

applications/programming, high level/low level, compiled/interpreted. Problem solving, flow charts. Loading VB6 Working Edition, saving and naming forms and projects Introduction to the Visual Basic Environment: Objects, Properties Visual Basic Events: Using code to change Properties of Objects. REM statements. Arithmetic Operations: Level of Precedence, Numeric Variables Print into picture Box Declaring Variable Types, Val-and Str-functions pp Mid-Term Examination, Relational Operators Input from and InputBox, General Procedures, Passing Variables, Scope of Variables Controlling Program Flow: If Blocks, Logical Operators Controlling Program Flow, For…Next Loops Controlling Program Flow, Do Loops, error detection Basic Stamp by Parallax, programming a microcontroller with BASIC Final Exam 20 Grades Item 1 2 3 4 Description Weight There will be NO make up exam. A missed test or exam will count as a zero 25 % 25 % 25 % 25 % TOTAL 100 % Ten weekly online exams, each worth 2.5 % Ten weekly Lab assignments each worth 2,5 % One Mid-Term Exam One Final Exam Grade Scale A A- 100-96 95-90 B+ B B- C+ C C- D+ D D- F 89-87 86-84 83-80 79-77 76-74 73-70 69-67 66-64 63-60 59-0 21 Assessment for STEM Academy Instructor: Mike Metaxas Course: ET 710 (Building and Maintaining Web Sites) General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Use HTML and Notepad to create a functional Web page Assessment Plan and Rubrics: Assessment Plan &amp; 0 1 2 3 4 Rubrics Description: The student does not The student is able to use The students The student The student Notepad but does not Use Notepad to create have basic computer understands some of understands most of understands the skills and cannot use understand the HTML basics of HTML as a Web Page. the HTML but does not the HTML and the Notepad. concepts presented. Demonstrate a basic grasp the use of a use of a browser but presented including page structure and knowledge of HTML browser to view the is missing the tags and is able to including page web page just created concept of page create a functional structure and structure. Web page rudimentary HTML tags. View the Web page in a browser 22 General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Apply Microsoft Expression Web Software to Web page design. Assessment Plan and Rubrics: Assessment Plan &amp; 0 1 2 3 4 Rubrics Description: The student does not The student is able use The student is able to The student can use The student can use Use Microsoft's have basic computer the Expression Web create a Web page Expression web and Expression web and Expression Web skills and cannot use software but does not work with it to work with it to create a using Expression Web. Software to create a the Expression Web understand a WYSIWYG create a basic web web page but does not He can use most of Web page. Understand software. (what you see is what you page but does not understand publishing the basic features to what the software does get) editor. add unique own style personalize the web and using the server. and apply it to the web page. Can pages. appropriately. preview and easily Understand how to preview a page from within Expression Web and how to create a web site on a flash drive. Connect to and publish files to the web server make changes. Publishes to the server 23 Electrical and Computer Engineering Technology Department Queensborough Community College The City University of New York ET710 - Building and Maintaining Web Sites Hours: 3 Class hours, 3 Laboratory hours, 4 credits In order to pass the course a passing grade must be obtained in both the LECTURE and LABORATORY components of the course. Course Purpose/Objectives: Students will learn most of the important topics of Microsoft's Expression Web software. Building and Maintaining Web Sites teaches the student to make Web pages, Publish Web pages to a Web Server, and troubleshoot any and all problems associated with their Web site design. Prerequisites: It is expected that the student will have basic knowledge in the use of a personal computer and a basic understanding of the Vista operating system. In particular the student should know how to use Window Explorer to save, copy, delete and move files. No prior knowledge or experience in the use of a database is required. Text/Materials: Text: ET-710 Building and Maintaining Web Sites Laboratory Guide Second Edition by Professor Robert M. Kueper Software: Microsoft Expression Web, Internet Explorer 7.0 128 Bit encryption, Windows Vista. Teaching Methods: Lectures: Important material from the laboratory guide and PowerPoint presentations will be covered in class. Students should plan to take careful notes as not all material can be found in the readings. Discussion is encouraged as is student-procured, outside material relevant to topics being covered. 24 Demonstrations: Much of the class time will be used to demonstrate the practical use of the software for this course. Students are encouraged to carefully observe the demonstrations and also to go through the tutorial material in the laboratory guide. Assignments: Weekly assignments (which enforce the project based learning objectives) include web pages, summaries and answers to critical thinking questions. All written material will be graded according to Writing Intensive principals and feedback is given to the students using the "comments" feature in Blackboard. Summaries: Summaries should be a minimum of 1 page in length and contain an introductory paragraph, a paragraph containing the details of what was necessary to complete the assignment, and a conclusion. For example: John Smith April 20, 2009 Assignment Number 3 Introduction: Here you should write about the goals of this assignment. This section should be one or two sentences long. Details: Here you should write about how and what you did in order to complete the assignment. This section should be at least one paragraph and should contain the bulk of your writing. Conclusions: Here you should write your thoughts about the assignment and what conclusions you made. Page 1 Critical Thinking Questions: Question 1 Your answer 25 Question 2 Your answer Question 3 Your answer Question 4 Your answer Question 5 Your answer Page 2 Quizzes: Weekly quizzes will be given to help ensure students stay up to date with assigned material. Internet: All material will be distributed on the Internet (Blackboard). Class Syllabus, Quizzes and Final will be posted on Blackboard. Grading: Letter grades will be determined using a standard percentage point evaluation as outlined below. Grades MAY be curved after the total semester points have been tabulated Grade A AB+ B BC+ Percent 96%-100% 90%-95% 87%-89% 84%-86% 80%-83% 77%-79% 26 C CD+ D DF 74%-76% 70%-73% 67%-69% 64%-66% 60%-63% Below 60% The final grade will be computed using the following weights: Lab Assignments Weekly Quizzes Term Project Final Total 1100 Points 1200 Points 200 Points 300 Points 2800 Points 40% 43% 7% 10% 100% Course Policies: Missed Classes: The student is responsible for obtaining material, which may have been distributed on class days when he/she was absent. This can be done through contacting a classmate or logging into blackboard and download the assignments as needed. Missed or late quizzes cannot be made up under any circumstances but with good cause and adequate notice, an early quiz may be given. Any unexcused missed exam will result in a score of 0 for that exam. Assignments: All assignments are due at the beginning of class on the week following the lab assignment. Late submissions of lab assignments may be assessed a penalty of 10% per day. Lab assignments that are two weeks past due will be may be given a zero 0 for that assignment. Academic Dishonesty: Plagiarism and cheating are serious offenses and may be punished by failure on exam, paper, project, course and or expulsion from the University. For more information, refer to the &quot;Academic Dishonesty&quot; policy in the University Undergraduate Catalog. Need for Assistance: If you have any condition, such as a physical or learning disability, which will make it difficult for you to carry out the work as outlined or which will require academic accommodations, please notify the instructor as soon as possible. 27 Posting of Grades: Final grades will not be posted. If you wish to have your final grade sent to you, please bring a self-addressed, stamped envelope to the final exam. Instructor: Mike Metaxas Office Location: Technology Building Room T-20 Telephone: 718-631-6207 Email: mmetaxas@qcc.cuny.edu 28 Syllabus Subject Material Projects Project#1: Using Notepad to make a Web Page Pages 3-10, Sample quiz Project#2: Introduction to Expression Web Pages 11-15, Quiz #1 Project#3: Create and Modifying Web Pages Pages 17-21, Quiz #2 Project#4: Creating and Verifying Hyperlinks Pages 23-29, Quiz #3 Project#5: Creating and Modifying Expression Web Templates Pages 31-36, Quiz #4 Project#6: Defining Graphic on a Web Page Pages 37-44, Quiz #5 Project#7: Creating and Utilizing Tables Pages 45-50, Quiz #6 Project#8: Creating and Manipulating a Web Form Pages 51-58, Quiz #7 Project#9: Implementing Frames Pages 59-65, Quiz #8 Project#10: Introduction to Adobe Fireworks Pages 67-72, Quiz #9 Project#11: Advanced Web Page Development features Pages 73-78, Quiz #10 Project#12: Web Administration &amp; Troubleshooting Pages 79-87 Quiz #11 Term Project Pages 88-89 Review for final Quiz #12 29 30
Источник: https://studylib.net/doc/11104365/stem%C2%A0a-academ-my%C2%A0coh-horts%C2%A0r

ISBN 13: 9781259097096 Instructor Solutions Manual Sundaram

skyusbook

unread,
Aug 12, 2018, 5:48:14 AM8/12/18
□

Reply to author

Sign in to reply to author

Forward

Sign in to forward

Delete

You do not have permission to delete messages in this group

Link

Report message as abuse

Sign in to report message as abuse

Show original message

Either email addresses are anonymous for this group or you need the view member email addresses permission to view the original message

to

[Sanjiv Das Rangarajan Sundaram] Derivatives: Principles and Practice 1st Edition McGraw Hill, 2013 Instructor Solutions Manual ISBN 13: 9781259097096

The solutions manuals to huge list of educational books are ready for use. They all have full and precise solutions. Both instructors and students can use these solutions manuals.
To get the solutions manual you need, feel free to send an email with the title.

NOT free

Contact email: skyusbook"@"gmail.com (@ with no quotations marks)

To search by author's name use Ctrl + F

Finite Math and Its Application (9th Ed., Larry J Goldstein, Schneider & Martha J. Siegel)
[L. D. Field, H. L. Li, A. M. Magill] Organic Structures from 2D NMR Spectra, Kindle Edition Set 2016 ISBN: 978-1-119-07370-3
Modeling in Transport Phenomena: A Conceptual Approach (2nd Ed.,Ismail Tosun)
Batch Distillation: Simulation, Optional Design and Control (Diwekar)
Mechanics of Flight (Warren F. Phillips)
Francis Carey, Neil Allison - Solutions Manual to accompany Organic Chemistry, Seventh Edition (2007, McGraw-Hill Science_Engineering_Math)
Calculus with Applications for the Life Sciences (Raymond N. Greenwell, Nathan P. Ritchey & Margaret L. Lial)
Principles of Combustion (2nd Ed., Kenneth Kuan-yun Kuo)
[Fulay, Donald R. Askeland, Pradeep P.] The Science , Engineering of Materials 5th Edition Thomson; 5th ISBN-13: 978-0534553968
[Mark J. Schulz, Ajit D. Kelkar, Mannur J. Sundaresan] Nanoengineering of Structural Functional and Smart Materials 1st Edition CRC Press; 1st ISBN-13: 978-0849316531
[Boas, Mary L.] Mathematical Methods in the Physical Sciences 3rd Edition Wiley; 3 ISBN-13: 978-0471198260
Computer Numerical Control: Operation and Programming (3rd Ed., Stenerson & Curran)
Introductory Algebra (8th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
Essentials of Stochastic Processes (Rick Durrett)
Customer Relationship Management: The Bottom Line to Optimizing Your ROI (Anton & Petouhoff)
[Elements of Chemical Reaction Engineering by Fogler hubbard, hamman , johnson , 3rd edition
[Baker,R. Jacob] CMOS: Mixed-Signal Circuit Design, Second Edition 2nd Edition Wiley-IEEE Press; 2 ISBN-13: 978-0470290262
Elements of Statistical Mechanics: With an Introduction to Quantum Field Theory and Numerical Simulation (Ivo Sachs & Siddhartha Sen)
Prealgebra (5th Ed., Marvin L. Bittinger, David J. Ellenbogen & Barbara L. Johnson)
[Gary Chartrand, Albert D. Polimeni, Ping Zhang] Mathematical Proofs: A Transition to Advanced Mathematics 2nd Edition Pearson; 2nd 2007 ISBN-13: 978-0321390530
Process Control Instrumentation Technology (8th Ed., Curtis Johnson)
[Richard E. Baker, Valdean C. Lembke, Thomas E. King] Advanced Financial Accounting 5th Edition Mcgraw-Hill College; 5th ISBN-13: 978-0072444124
Construction Contracts (3RD Ed., Jimmie Hinze)
[Godfrey Beddard] Applying Maths in the Chemical and Biomolecular Sciences: An Example-Based Approach 1st Edition Oxford University Press; 1st ISBN-13: 978-0199230914
Applied Statistics and Probability for Engineers (4th Ed., Douglas Montgomery & George Runger)
[Jimmy T. Arnold, Lee W. Johnson, R. Dean Riess] Introduction to Linear Algebra 5th Edition Pearson; 5th 2001 ISBN-13: 978-0201658590
Introduction to Mathematical Structures and Proofs (Larry J. Gerstein)
[David R. Gaskell Introduction to the Thermodynamics of Materials 3rd Edition Taylor & Francis ISBN 13: 9781560324324
[J. David Logan Applied Mathematics 3rd Edition 2006 Wiley-Interscience ISBN-13: 978-0471746621
A Graphical Approach to Precalculus (4th Ed., John Hornsby, Lial & Rockswold)
Donald R. Askeland (auth.) - The Science and Engineering of Materials_ Solutions Manual (1991, Springer Netherlands)
[John D. Cutnell, Kenneth W. Johnson] Physics, 9th Edition John Wiley and Sons;(2012) ISBN: 978-0-470-87952-8
Design of Analog CMOS Integrated Circuits, 1st Edition, by Razavi Douglas C. Montgomery
Single Variable Calculus (3rd Ed., Monty J. Strauss, Gerald L. Bradley & Karl J. Smith)
Tony Gaddis - Starting Out with Java_ From Control Structures through Objects, Instructor Solution Manual (2012, Addison-Wesley)
[Francis A. Carey, Robert M. Giuliano] Organic Chemistry, 8th Edition 2010 McGraw-Hill ISBN-13: 978-0077354770
CMOS VLSI Design: A Circuits and Systems Perspective, 3/E., Neil Weste, David Harris
Simulation Modeling and Analysis (4th Ed., Averill Law)
Oracle 10g Programming: A Primer (Rajshekhar Sunderraman)
[Curtis Howard] Orbital Mechanics For Engineering Students 1st Edition Butterworth-Heinemann 2004 ISBN: 9780080470542

Intermediate Algebra: Graphs & Models (3rd Ed., Marvin L. Bittinger, David J. Ellenbogen & Barbara L. Johnson)
Aircraft Structures for Engineering Students (4th Ed., T.H.G. Megson)
Java Software Structures: Designing and Using Data Structures (2nd Ed., Lewis, Chase)
A Quantum Approach to Condensed Matter Physics (Philip L. Taylor & Olle Heinonen)
File Structures: An Object-Oriented Approach with C++ (3rd Ed., Folk, Zoellick, Riccardi)
[Howard Curtis] Orbital Mechanics for Engineering Students Kindle 2nd Edition Butterworth-Heinemann 2009 ISBN: 9780080887845
[Paul M. Fischer, William J. Taylor and Rita H. Cheng] Advanced Accounting 10th edition Publisher: South-Western Publishing Co.2009 ISBN13: 9780324379051
Leslie D. Field, Sev Sternhell, John R. Kalman - Organic Structures from Spectra_ Solutions Manual (2007, John Wiley and Sons Ltd)
[Rousseau, Richard M. Felder, Ronald W.] Elementary Principles of Chemical Processes 3rd Edition John Wiley & Sons; 3rd 2004 ISBN-13: 978-0471375876
Principles of Managerial Finance plus MyFinanceLab Student Access Kit, 12th Ed., Lawrence J. Gitman)
Developmental Mathematics (K. Elayn Martin-Gay)
[Edward B. Saff, Arthur David Snider] Fundamentals of Complex Analysis with Applications to Engineering, Science, and Mathematics 3rd Edition Pearson; 3rd 2003 ISBN-13: 978-0139078743
Biostatistics for the Health Sciences (R. Clifford Blair & Richard Taylor)
Concrete Structures (Mehdi Setareh & Robert M. Darvas)
Willem Conradie, Valentin Goranko, Claudette Robinson - Logic and Discrete Mathematics_ A Concise Introduction, Solutions Manual (2015, Wiley)
[Falconer, Kenneth J.] Fractal Geometry: Mathematical Foundations and Applications 2nd Edition Wiley, 2003 ISBN 13: 9780470848623
Prealgebra & Introductory Algebra (2nd Ed., K. Elayn Martin-Gay)
Data Structures and Algorithm Analysis in C (2nd Ed., Mark Allen Weiss)
T. W. Graham Solomons, Craig B. Fryhle, Robert G. Johnson - Organic Chemistry_ Study Guide and Solutions Manual (2008, Wiley)
McCabe_ Julian C Smith_ Peter Harriott - Unit operations of chemical engineering [SOLUTIONS MANUAL] (1985, McGraw-Hill) [McGraw-Hill chemical engineering series] Warren L
[Robert L. Carter] Molecular Symmetry and Group Theory Wiley ISBN-13: 978-0471149552
[Goodman D.Yates, David J.Roy] Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers 2nd Edition Wiley; 2nd 2004 ISBN-13: 978-0471272144
[David C. Lay] Linear Algebra and Its Applications,Books a la Carte edition 4th Edition Pearson;2011 ISBN-13: 978-0321385178
[Michael R. Cullen Dennis G. Zill] Advanced Engineering Mathematics: 3rd Edition Jones & Bartlett Learning, LLC 2006 ASIN: B0089AEAJK
Introduction to Engineering Experimentation (2nd Ed., Wheeler & Ganji)
[Jordan, Stephen A. Ross, Randolph Westerfield, Bradford D.] Essentials of Corporate Finance 6th Edition McGraw-Hill; 6th ASIN: B0041W2KEC
[Rosen, Kenneth H.] Elementary Number Theory 5th Edition Addison Wesley; 5th 2004 ISBN-13: 978-0321237071
[Stephen Ross, Randolph Westerfield, Jeffrey Jaffe] Corporate Finance, 8th Edition McGraw-Hill/Irwin; 8th 2006 ISBN-13: 978-0073337180
[Douglas C. Montgomery] Introduction to Statistical Quality Control 4th Edition John Wiley & Sons 2007 ISBN-13: 978-0471381853
Intermediate Algebra with Applications and Visualization (2nd Ed., Gary K. Rockswold & Terry A. Krieger)
[Sundem, Stratton, Burgstahler, Schatzberg Horngren] Introduction to Management Accounting:Pearson Education (2008) ISBN-13: 978-8120335684
C++ Programming Today, 2/E, (Barbara Johnston)
ELEMENTARY NUMBER THEORY AND ITS APPLICATIONS, (5TH EDITION, Bart Goddard, Kenneth H. Rosen)
[Mark A. Weiss Data structures and Algorithm Analysis in C 2nd Edition 1996 Pearson ISBN-13: 978-0201498400
Starting Out with Java: From Control Structures through Data Structures, by Gaddis, Muganda)
[International Trade 2nd Ed by Robert Feenstra, Alan Taylor
[Lightfoot, R. Byron Bird, Warren E. Stewart, Edwin N.] Transport Phenomena Revised 2nd Edition John Wiley & Sons, Inc.; 2nd 2006 ISBN-13: 978-0470115398
[Jerry Banks, John S. Carson, Barry L. Nelson , David M. Nicol] Discrete-Event System Simulation 3rd Edition Prentice Hall 2000 ISBN-13: 978-0130887023
Robert J. Schilling, Sandra L. Harris - Instructor Solution Manual - Fundamentals of Digital Signal Processing Using MATLAB
Elementary Statistics: Picturing the World (4TH Ed., Larson & Farber)
American Radio Relay League - The ARRL Instructor's Manual for Technician and General License Courses (2008, Amer Radio Relay League)
Principles of Highway Engineering and Traffic Analysis (3/E, Fred Mannering, Kilareski & Washburn)
[T. W. Graham Solomons, Craig B. Fryhle Organic Chemistry, 10th Edition 2010 Wiley ISBN-13: 978-0470478394
Finite Mathematics and Calculus with Applications (8th Ed.,L. Lial, Greenwell & Ritchey)
Experiments with Economic Principles, 2/E, by Bergstrom, Miller
[Johnson,Curtis D.] Process Control Instrumentation Technology 8th Edition Pearson; 8th 2005 ISBN-13: 978-0131194571
C. Henry Edwards, David E. Penney - Instructor's Solutions Manuals for Calculus Early Transcendentals 7th Edition (2007)
Design and Analysis of Experiments, 7/E, by Montgomery
[Douglas C. Montgomery] Introduction to Statistical Quality Control, 7th Edition, + Kindle Edition Wiley;(2009) ISBN-13: 978-1118146811
[Richard Baker] Advanced Financial Accounting 8th edition Richard D. Irwin, Inc. 2009 ISBN13: 9780073526911
[Thiele, Warren L. Stutzman, Gary A.] Antenna Theory and Design 2nd Edition Wiley; 2nd 1997 ISBN-13: 978-0471025900
Systems for Planning and Control in Manufacturing (D. K. Harrison & D. J. Petty)
Applied Numerical Analysis, 7th Edition, by Gerald, Wheatley
[C. T. Sun] Mechanics of Aircraft Structures 2nd Edition Kindle Wiley; 2nd 2006 ISBN-13: 978-0471699668
[Fitzgerald, A. E.Charles Kingsley, JR., Stephen D. Umans] Electric Machinery 6th Edition McGraw-Hill; 6th 2005 ISBN-13: 978-0071230100
[Adel S. Sedra, Kenneth C. Smith] Microelectronic Circuits 5TH Edition McGraw-Hill ISBN-13: 978-0195142525
Mathematics with Applications (9th Ed., Margaret L. Lial, Hungerford & John Holcomb)
Precalculus with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
[Richard C. Dorf and James A. Svoboda] Introduction to Electric Circuits 7th Edition John Wiley & Sons, Inc.7th 2006 ISBN13: 9780471730422
Prealgebra (3rd Ed., Margaret L. Lial & Diana L. Hestwood)
Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5/e, Luger
[Kenneth M. Leet Emeritus, Chia-Ming Uang, Joel Lanning] Fundamentals of Structural Analysis 3rd Edition McGraw-Hill ASIN: B0086HY1O0
[Principles of Highway Engineering and Traffic Analysis 4 ED (US Units) by Mannering,Washburn,Kilareski
[Baker, Richard E.; Lembke, Valdean C.; King, Thomas E.] Advanced Financial Accounting 5th Edition Mcgraw-Hill ASIN: B008BWV85A
Starting Out with Java 5: Control Structures to Objects (Tony Gaddis)
College Algebra and Trigonometry (4TH Ed., Margaret L. Lial, John Hornsby & David I. Schneider)
[Wesley G. Crawford] Construction Surveying and Layout 2nd Edition Creative Construction ISBN 13: 9780964742109
[Ravi Anupindi, Sunil Chopra, Sudhakar D. Deshmukh, Jan Van Mieghem] Managing Business Process Flows, Principles of Operations Management 2nd Edition Prentice Hall; 2nd 2005 ISBN-13: 978-0130675460
Heating, Ventilating and Air Conditioning Analysis and Design (6th Ed., McQuiston)
[Frederick S. Hillier, Gerald J. Lieberman] Introduction to Operations Research 7th Edition McGraw Hill ISBN-13: 978-0072461213
Calculus and its Applications (11th Ed., Larry J Goldstein, Schneider, Lay & Asmar)
[Walter T. Harrison Jr., Charles T. Horngren , Bill Thomas] Financial Accounting 8th Edition Pearson; 8th 2009 ISBN-13: 978-0136108863
PIC Microcontroller (Muhammad Ali Mazidi, Rolin McKinlay & Danny Causey)
Concrete Abstract Algebra: From Numbers to GrÃ¶bner Bases (Niels Lauritzen)
[Carey, Francis A., Sundberg, Richard J.] Advanced Organic Chemistry Part A: Structure and Mechanisms Academic Press, 1996 ISBN 978-0-387-44899-2

Calculus, Early Transcendentals (7th Ed., C. Henry Edwards & David E. Penney)
Fundamentals of Probability, with Stochastic Processes (3rd Ed., Saeed Ghahramani)
[Cengage Learning studyguide 1] Ron Larson - solutions manual _ Precalculus with limits (2014, Cengage Learning)
Algebra and Trigonometry with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
Trigonometry (9th Ed., Margaret L. Lial, John Hornsby & David I. Schneider)
Ultrasonic Nondestructive Evaluation Systems: Models and Measurements (Lester W. Schmerr & Sung-Jin Song)
[James M. Reeve, Jonathan E. Duchac Carl S. Warren] Accounting 25th Edition Latest version included Loose Leaf Cengage Learning (2014) ISBN-13: 978-1285914794
[Ken A. Dill, Sarina Bromberg] Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience, 2nd Edition Garland Science; 2nd ISBN-13: 978-0815344308
Orbital Mechanics for Engineering Students (Howard Curtis)
James Wilson - A Hungerford's Algebra Solutions Manual (2002, University of Oregon, Portland State University.)
[John R. Hubbard, Anita Huray] Data Structures with Java United States Ed Edition Prentice Hall; ISBN-13: 978-0130933744
Advanced Engineering Mathematics,3/E, by Zill, Cullen
Introductory and Intermediate Algebra (3rd Ed., argaret L. Lial, John Hornsby & Terry McGinnis)
Calculus with Applications (9th Ed., Margaret L. Lial, Greenwell & Nathan P. Ritchey)
Intermediate Algebra (5TH Ed., K. Elayn Martin-Gay)
Mechanics of Aircraft Structures (2nd Ed, C. T. Sun)
[Randolph W Westerfield Robert R. Dockson Deans Chair in Bus. Admin] Essentials of Corporate Finance, 8th Edition McGraw-Hill Education; 8th 2013 ISBN-13: 978-0078034756
[Margaret L. Lial , Raymond N. Greenwell, Nathan P. Ritchey] Calculus with Applications 8th Edition Addison-Wesley; 8th 2004 ISBN-13: 978-0321238436
Customer Service: A Practical Approach (4th Ed., Harris)
[Endre Süli, David F. Mayers] An Introduction to Numerical Analysis 1st Edition Cambridge University Press;2003 ISBN-13: 978-0521007948
Technical Mathematics (2th Ed., Dale Ewen, Joan S. Gary & James E. Trefzger)
The Cosmic Perspective, 5/E, Jeffrey O. Bennett, Megan Donahue, Nicholas Schneider, Mark Voit
[Warren McCabe, Julian Smith, Peter Harriott emeritus] Unit Operations of Chemical Engineering 7th Edition McGraw-Hill Education; 7 2004 ISBN-13: 978-0072848236
[Lawrence J. Gitman] Principles of Managerial Finance 4th Edition Pearson Education (2005) ISBN 13: 9781740911016
[Sanjiv Das Rangarajan Sundaram] Derivatives: Principles and Practice 1st Edition McGraw Hill, 2013 ISBN 13: 9781259097096
Organic Chemistry, 7/E., by Carey, Atkins
[Curtis F. Gerald, Patrick O. Wheatley] Applied Numerical Analysis 7th Edition Pearson; 7 2003 ISBN-13: 978-0321133045
Thomas E. Copeland J. Fred Weston Kuldeep Shastri - Financial Theory and Corporate Policy FOURTH EDITION SOLUTIONS MANUAL
[Paul D. Kimmel, Jerry J. Weygandt , Donald E. Kieso] Accounting Principles 9th Edition for SouthWestern Illinois College-Belleville 9th Edition Wiley; 9th 2009 ISBN-13: 978-0470539118
Developmental Mathematics: Basic Mathematics and Algebra (Margaret L. Lial, John Hornsby, Terry McGinnis, Stanley A. Salzman & Diana L. Hestwood)
[David R. Anderson, Dennis J. Sweeney, Thomas A. Williams, Jeffrey D. Camm, R. Kipp Martin] An Introduction to Management Science: Quantitative Approaches to Decision Making, Revised 13th Edition South-Western; 13th 2011 ISBN-13: 978-1111532222
Elementary Math Modeling Updated (2nd Ed., Mary Ellen Davis & C. Henry Edwards)
[Kenneth S. Krane] Modern Physics, 3rd Edition, + Kindle Edition Wiley; (2012) ISBN-13: 978-1118061145
Calculus, An Applied Approach, 8/E, by Larson
[Ralph H. Petrucci, William S. Harwood, Geoffrey Herring] General Chemistry: Principles and Modern Applications 8th Edition Prentice Hall; 8th 2001 ISBN-13: 978-0130143297
Marilyn E. Noz, Gerald Q. Maguire - Radiation Protection in the Health Sciences_ With Problem Solutions Manual (2007)
A Graphical Approach to Algebra and Trigonometry (4th ed., John Hornsby, Margaret L. Lial & Gary K. Rockswold)
[Rotman, Joseph J] Journey Into Mathematics: An Introduction to Proofs Dover Publications Inc. 2006 ISBN-13: 978-0486453064
[Clough, R.W.] Dynamics of Structures International 2nd Revised Edition, 1993 ISBN-13: 978-0071132411
An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation (Desmond Higham)
[Douglas B. West] Introduction to Graph Theory 2nd Edition Pearson; 2nd ISBN-13: 978-0130144003
[Behzad Razavi] Fundamentals of Microelectronics 2nd Edition Wiley; 2nd 2013 ISBN-13: 978-1118156322
[Keown Arthur J., John H. Martin, John W. Petty, David F. Scott] Financial Management: Principles and Applications 10th Edition Prentice Hall; 10 2004 ISBN-13: 978-0131450653
Oracle 9i Programming: A Primer (Rajshekhar Sunderraman)
[Arthur Nilson , David Darwin , Charles Dolan] Design of Concrete Structures 14th Edition McGraw-Hill Education; 14th ISBN-13: 978-0073293493
[T W Graham Solomons; Craig B Fryhle; S A Snyder; Robert G Johnson; Jon Antilla] Organic chemistry, 11th Edition Wiley, [2014] ISBN: 9781118147900
Multivariate Data Analysis (6th Ed., Hair, Black, Babin, Anderson & Tatham)
[Randall D. Knight, Brian Jones, Stuart Field] College Physics: A Strategic Approach Volume 1 2nd Edition Addison-Wesley;(2009) ISBN-13: 978-0321611147
[Samuel Karlin, Howard M. Taylor] An Introduction to Stochastic Modeling, 3rd Edition Academic Press; 3rd 1998 ISBN-13: 978-0126848878
[Welty, James, Wicks, Charles E., Wilson, Robert E., Rorrer] Fundamentals of Momentum, Heat, and Mass Transfer 4th edition Wiley; 4th 2000 ISBN-13: 978-0471381495
Understanding UNIX/LINUX Programming: A Guide to Theory and Practice(Bruce Molay)
Introduction to Statistical Quality Control (6th Ed., Douglas C. Montgomery)
The 8051 Microcontroller (4th Ed., I. Scott MacKenzie, Raphael Chung-Wei Phan)
[Charles H. Gibson] Financial Reporting and Analysis - Using Financial Accounting Information 10th Edition Thomson/SouthWestern; 10th 2006 ASIN: B0058SMY1M
Modern Electronic Communication (9th Ed., Jeff Beasley, Gary M.Miller)
[Kenneth Kuttler] Calculus Multivariable: Theory and Applications World Scientific Pub Co Inc 2011 ISBN-13: 978-9814335713
Jerry D. Wilson, Cecilia A. Hernández - Instructor's Manual to Physics Laboratory Experiments (2014, Cengage Learning)
[Lawrence E. Spence, Stephen H. Friedberg, Arnold J. Insel] Linear Algebra, 4th Edition Fourth Pearson; 4th 2002 ISBN-13: 978-0130084514
Advanced Engineering Mathematics (2nd Ed, Michael Greenberg)
Essentials of College Algebra with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
Beginning and Intermediate Algebra (4th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
[Faye C. McQuiston, Jerald D. Parker, Jeffrey D. Spitler] Heating, Ventilating and Air Conditioning Analysis and Design 6th Edition Wiley; 6th 2004 ISBN-13: 978-0471470151
[James K. Wight; James G. MacGregor] Reinforced Concrete: Mechanics and Design 5th Edition Prentice Hall, 2008 ISBN 13: 9780132281416
[Donald R. Askeland, Pradeep P. Fulay] The Science & Engineering of Materials, 5th Edition Thomson; 5th 2005 ISBN-13: 978-0534553968
An Introduction to Numerical Analysis (Endre Suli and David Mayers)
[C. Henry Edwards, David E. Penney Calculus, Early Transcendentals 6th Edition Kindle Prentice Hall;2002 ISBN-13: 978-0130084071
Essentials of College Algebra, Alternate Edition (Margaret L. Lial, Hornsby & Schneider)
[Raphael C.-W. Phan I. Scott Mackenzie] The 8051 Microcontroller 4th International Edition Pearson ASIN: B001KR5G5G
Saber Elaydi - Instructor's Solutions Manual to An Introduction to Difference Equations (2005, Springer)

Fundamentals of Complex Analysis with Applications to Engineering, Science, and Mathematics (3rd Ed., E. Saff & Arthur Snider)
[Michael Spivak] Calculus On Manifolds: A Modern Approach To Classical Theorems Of Advanced Calculus Westview Press; ISBN-13: 978-0805390216
Elementary Linear Algebra (2nd Ed., Spence, Insel & Friedberg)
[Tony Chan Carusone, David Johns, Kenneth Martin] Analog Integrated Circuit Design 2nd Edition Wiley; 2 2011 ISBN-13: 978-0470770108
Data Structures and Problem Solving Using C++ (2nd Ed., Mark Allen Weiss)
[Woolridge, Jeffrey M.] Introductory Econometrics: A Modern Approach 3rd Edition Thomson South-Western; ISBN-13: 978-7302143123
Algebra A Combined Approach (3rd Ed., K. Elayn Martin-Gay)
Construction Contracts (2nd Ed., Hinze)*
Discrete Mathematics (7th Ed., Richard Johnsonbaugh)
Stephen A. Ross, Randolph W. Westerfield, Jeffrey Jaffe - Instructor's Manual to accompany Corporate Finance, Sixth Edition (2001, McGraw-Hill Inc.,US)
[Kenneth W. Whitten, Raymond E. Davis, Larry Peck, George G. Stanley] Chemistry 10th Edition Cengage Learning 2013 ISBN-13: 978-1133610663
Francis A. Carey, Richard J. Sundberg - Solutions Manual for Advanced Organic Chemistry Part B_ Reactions and Synthesis (2007)
Probability and Statistics in Engineering (4th Ed., Hines, Montgomery, Goldsman & Borror)
[Pru Marriott, J R Edwards , Howard J Mellett] Introduction to Accounting 3rd Edition, Kindle Edition ISBN-13: 978-0761970378
Prealgebra and Introductory Algebra (2nd Ed., Margaret L. Lial, John Hornsby, Terry McGinnis & Diana L. Hestwood)
[Robert J. Schilling , Sandra L Harris, Fundamentals of Digital Signal Processing Using MATLAB 2nd Edition 2011 Cengage Learning ISBN-13: 978-0840069092
[R. Jacob Baker] CMOS Circuit Design, Layout, and Simulation, Second Edition 2nd Edition Wiley-IEEE Press; 2 ISBN-13: 978-0471700555
[Margaret L. Lial, Raymond N. Greenwell, Nathan P. Ritchey] Finite Mathematics 8th Edition Addison Wesley; 8th 2004 ISBN-13: 978-0321228260
[Bradford D. Jordan , Stephen A. Ross, Randolph Westerfield] Fundamentals of Corporate Finance, 8th Edition, Irwin/McGraw-Hill; 2008 ISBN-13: 978-0073282121
[Lawrence J. Gitman . Chad J.Zutter] Principles of Managerial Finance: Global Edition Pearson Education; 2011 ISBN-13: 978-0273754282
Tony Gaddis, Diane Christie - Starting Out with Java_ From Control Structures through Objects, Lab Manual Solutions (2012, Addison-Wesley)
[Ron Larson, Bruce H. Edwards] Calculus 10th Edition Kindle Cengage Learning; 10th ISBN-13: 978-1285057095
Electric Machinery (6th Ed., Fitzgerald)
Supply Chain Management (3rd Ed., Sunil Chopra & Peter Meindl)
Data Structures and Other Objects Using Java (3rd Ed., Michael Main)
[Athanasios Papoulis, S. Unnikrishna Pillai] Probability, Random Variables and Stochastic Processes 4th Edition McGraw-Hill Europe; 4th ISBN-13: 978-0071226615
Fundamentals of Investing plus MyFinanceLab Student Access Kit and OTIS Student Access Kit (10th Ed., Lawrence J. Gitman & Michael D.Joehnk)
Introduction to Programming Using Visual Basic 2008 (David I. Schneider), An, 7/E
[Kenneth Kuttler] Elementary Linear Algebra 2014 ISBN: 978-87-403-1425-0
Finite Math with Applications (9th Ed., Margaret L. Lial, Hungerford & John Holcomb)
Modeling and Simulation of Dynamic Systems (Woods, Lawrence)
[Montgomery, Douglas C.] Design and Analysis of Experiments 6th Edition John Wiley and Sons ISBN-13: 978-0470088104
Java Foundations: Introduction to Program Design and Data Structures, Lewis, DePasquale, Chase
Data Structures and Problem Solving Using Java (3rd Ed., Mark Allen Weiss)
Daniel C. Harris - Quantitative Chemical Analysis, Solutions Manual (2010, W.H. Freeman & Company)
Fluency with Information Technology, Brief Edition (Lawrence Snyder)
Econometrics of Financial Markets, by Adamek, Cambell, Lo, MacKinlay, Viceira
[T.H.G. Megson] Aircraft Structures for Engineering Students 4th Edition (Elsevier Aerospace Engineering) Butterworth-Heinemann; 4th ISBN-13: 978-0750667395
[Michael D. Greenberg] Advanced Engineering Mathematics 2nd Edition Dorling Kindersley Pvt Ltd;(2006) ISBN-13: 978-8177585469
[Theodore Bergstrom, John Mille] Experiments with Economic Principles: Microeconomics 2nd Edition McGraw-Hill/Irwin; 2nd 1999 ISBN-13: 978-0072295184
Walter Gautschi - Instructor's Solutions Manual to Numerical Analysis (2012, Springer)
Operating Systems (3rd Ed., Gary Nutt)
An Introduction to Statistical Signal Processing (Robert M. Gray)
[Daniel C. Harris] Quantitative Chemical Analysis 8th Edition 2010 W.H. Freeman ISBN-13: 978-1429264846
Introduction to Programming Using Visual Basic 2005 (6th Ed., David I.Schneider)
Prealgebra (5th Ed., K. Elayn Martin-Gay)
Applied Multivariate Statistical Analysis (6th Ed., Johnson & Wichern)
Introduction to Management Science with Student CD (9th Ed., Bernard W. Taylor)
Modern Welding Technology (6th Ed., Cary & Helzer)
[Sung-Mo Kang, Yusuf Leblebici] CMOS Digital Integrated Circuits Analysis & Design 3rd Edition McGraw-Hill Education; 2002 ISBN-13: 978-0072460537
[Daryl L. Logan] A First Course in the Finite Element Method 5th Edition CL Engineering;2011 ISBN-13: 978-0495668251
Data Structures in Java: From Abstract Data Types to the Java Collections Framework, by Gray
Heating and Cooling of Buildings: Design for Efficiency (2nd Ed., Jan F. Kreider, Peter S. Curtiss & Ari Rabl)
[Nicholas J. Garber, Lester A. Hoel] Traffic & Highway Engineering, 3rd Edition Brooks / Cole Thomson Learning 2002 ISBN 13: 9780534387433
[Ross Westerfield Jordan] Essentials of Corporate Finance 7th Edition McGraw-Hill; 7th ISBN-13: 978-0077550073
Pointers on C (Kenneth Reek)
Fundamentals of Database Systems/Oracle 9i Programming (5th Ed., Ramez Elmasri, Shamkant B. Navathe & Rajshekhar Sunderraman)
[South-Western Federal Taxation 2012 - Corporations, Partnerships, Estates and Trusts, 35th Ed by Hoffman, Maloney
[Richard L. Burden, J. Douglas Faires] Numerical Analysis 8th Edition Brooks Cole; 8th ISBN-13: 978-0534392000
Intermediate Algebra (10th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
Data Structures and Algorithm Analysis in Java (2nd Ed., Mark Allen Weiss)
L. D. Field, H. L. Li, A. M. Magill - Instructors Guide and Solutions Manual to Organic Structures from 2D NMR Spectra (2015, Wiley)
Introductory Statistics for Engineering Experimentation (Peter Nelson, Copeland & Coffin)
Discrete Mathematics (5th ed., John Dossey, Albert Otto, Spence & Charles Vanden Eynden)
The Finite Element Method: Its Basis and Fundamentals (6th Ed., Zienkiewicz, Taylor Zhu)
[Robert C. Feenstra, Alan M. Taylor] International Trade 2nd Edition Worth Publishers; 2010 ISBN-13: 978-1429241045
[Birger Bergersen, Michael Plischke, Birger Bergersen] Equilibrium Statistical Physics 2nd Edition World Scientific Publishing Company ISBN-13:9789810216429
Leslie D. Field, Sev Sternhell, John R. Kalman - Organic Structures from Spectra_ Solutions Manual (2014, John Wiley and Sons Ltd)
[Douglas C. Montgomery] Introduction to Statistical Quality Control 5th Edition Wiley; 5th ISBN-13: 978-0471656319
Professionalism: Real Skills for Workplace Success (Anderson & Bolt)
[Francis A. Carey, Richard J. Sundberg] Advanced Organic Chemistry, Part A: Structure and Mechanisms 5th Edition Springer; 5th 2008 ISBN-13: 978-0387683461
Transport Phenomena in Multiphase Systems (Faghri & Zhang)
Probability Random Variables, and Stochastic Processes, 4th Ed., by Papoulis, Pillai
Francis A. Carey, Richard J. Sundberg - Solutions Manual for Advanced Organic Chemistry Part A_ Structure and Mechanisms (2007)

[Fred L. Mannering, Scott S. Washburn, Walter P. Kilareski] Principles of Highway Engineering and Traffic Analysis 4th Edition Wiley;4th ISBN-13: 978-8126531653
[Kassimali, Aslam] Structural Analysis 3rd Edition CL-Engineering 2004 ISBN-13:9780534391683
Essentials of Geometry for College Students (2nd Ed., Margaret L. Lial, Barbara A. Brown, Arnold R. Steffenson & L. Murphy Johnson)
[Roy D. Yates, David J. Goodman] Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers 2nd Edition Wiley; 2nd 2004 ISBN-13: 978-0471272144
John West-Burnham, Ingrid Bradbury - Performance Management Manual Pack_ Creating a Culture for Sustainable High Performance (Schools Management Solutions)
Introductory Quantum Optics (Christopher Gerry & Peter Knight)
[Warren Nagourney] Quantum Electronics for Atomic Physics Oxford University Press ISBN-13 9780199665488
Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers,2/E, by Roy D. Yates , David J. Goodman
Engineering Statistics (4th Ed., Douglas Montgomery, George Runger & Norma Faris Hubele)
[T.H.G. Megson] Structural and Stress Analysis 2nd Edition Butterworth-Heinemann 2005 ISBN: 9780080455341
Dynamics of Structures (3rd Ed., Chopra)
Electronic Devices - Conventional Current Version (8th Ed., Thomas L.Floyd)
Technical Calculus (5th Ed., Dale Ewen, Joan S. Gary & James E. Trefzger)
[Warren Nagourney] Quantum Electronics for Atomic Physics Oxford University Press ISBN-13: 978-0199532629
Introductory Circuit Analysis (11th Ed., Robert L. Boylestad)
Reinforced Concrete Design (7th Ed., Chu-Kia Wang, Charles G. Salmon & JosÃ© A. Pincheira)
Structures (6th Ed., Daniel Lewis Schodek & Martin Bechthold)
[Lay, David C.] Linear Algebra and Its Applications, 3rd Updated 3rd Edition Addison Wesley; 2005 ISBN-13: 978-0321287137
[Charles T. Horngren, Walter T. Harrison Jr., M. Suzanne Oliver] Accounting, 8th Edition Prentice Hall; 8th 2008 ISBN-13: 978-0136072973
[Steven R. Lay] Analysis: With an Introduction to Proof (4th Edition) 4th Edition Pearson; 4th 2004 ISBN-13: 978-0131481015
Introduction to Finance (Lawrence J. Gitman & Jeff Madura)
[Rosen Kenneth H] Discrete Mathematics and Its Applications Seventh Edition 7th Edition Kindle Canada McGraw-Hill Education, 7th 2011 ISBN 13: 9780073383095
Douglas A. Skoog, Donald M. West, F. James Holler, Stanley R. Crouch - Instructor's Solutions Manual to Fundamentals of Analytical Chemistry (2013, Cengage Learning)
OSP: An Environment for Operating System Projects (Michael Kifer & Scott A. Smolka)
Construction Mathematics (Surinder Virdi & Roy Baker)
Reinforced Concrete Design (6th Ed., George F. Limbrunner & Abi Aghayere)
Basic College Mathematics (7th Ed., Margaret L. Lial, Stanley A. Salzman & Diana L. Hestwood)
[Gibson, Charles H.] Financial Reporting Analysis : Using Financial Accounting Information 10th Edition South-Western Publishing Co. 2007 ISBN13: 9780324304459
Basic College Mathematics with Early Integers (K. Elayn Martin-Gay)
[Logan, Daryl L.] A First Course In The Finite Element Method 4th Edition Nick Book House (Fresno, CA, U.S.A.) ISBN 13: 9780534552985
[Ghahramani, Saeed] Fundamentals of Probability, with Stochastic Processes 3rd Edition Pearson; 3rd 2004 Chapman and Hall/CRC 2015 ISBN-13: 978-0131453401
Design and Analysis of Experiments (7th Ed., Douglas Montgomery)
Physics,7/E, by John Cutnell
Calculus (3rd Ed., Monty J. Strauss, Gerald L. Bradley & Karl J. Smith)
Elementary Structures for Architects and Builders (5th Ed., Ronald E. Shaeffer)
Linear Algebra with Applications 3rd Ed by Otto Bretscher
[Steven R. Lay] Analysis: With an Introduction to Proof 4th Edition Pearson; 4th 2004 ISBN-13: 978-0131481015
[Jr. Edwards C.H., David E. Penney] Calculus With Analytic Geometry 4th Edition Prentice Hall College Div; 4th ISBN-13: 978-0134579122
Elementary and Intermediate Algebra: Graphs & Models (3RD Ed., Marvin L. Bittinger, David J. Ellenbogen & Barbara L. Johnson)
[David R. Anderson , Dennis J. Sweeney, Thomas A. Williams , R. Kipp Martin] An Introduction to Management Science: A Quantitative Approach to Decision Making (Book Only) 12th Edition South-Western College 2007 ISBN-13: 978-0324399790
Process Control Instrumentation Technology (8th Ed., Johnson)
[Clifford A. Shaffer] Practical Introduction to Data Structures and Algorithm Analysis 2nd Edition Prentice Hall ISBN 13: 9780130284464
[John D. Cutnell, Kenneth W. Johnson] Physics 8th Edition Wiley; 8th 2009 ISBN-13: 978-0470223550
Discrete Mathematics (Sherwood Washburn, Thomas Marlowe & Charles T.Ryan)
[James F. Sepe, Mark W. Nelson J. David Spiceland] Intermediate Accounting 2013 McGraw-Hill ISBN-13: 978-1259122064
[John M. Prausnitz, Rudiger N. Lichtenthaler, Edmundo Gomes de Azevedo Molecular Thermodynamics of Fluid-Phase Equilibria 3rd Edition Prentice Hall 3rd edition 1998 ISBN-13: 978-0139777455
Douglas A. Skoog, Donald M. West, F. James Holler - Solutions Manual for Skoog _ West Fundamentals of Analytical Chemistry, 9th Edition (2013, Brooks Cole)
Hazardous Waste Management (2nd Ed., Michael D. LaGrega, Phillip L Buckingham & Jeffrey C Evans)*
[Walter Nicholson Microeconomic Theory: Basic Principles and Extensions 9th Edition South-Western College Pub; 9th 2004 ISBN-13: 978-0324270860
Probability Random Variables, and Stochastic Processes, 4th Edition by Papoulis, Pillai
Analysis and Design of Analog Integrated Circuits, 4th Ed., by Gray,Hurst, Lewis, Meyer
Introduction to Linear Algebra (5th ed., Lee W. Johnson, R. Dean Riess & Jimmy T. Arnold)
[Kassimali Aslam] Matrix Analysis of Structures 2nd Edition Kindle Cengage Learning; 2nd ISBN-13: 978-1111426200
[Dilip Kondepudi, Ilya Prigogine] Modern Thermodynamics: From Heat Engines to Dissipative Structures 1st Edition John Wiley & Sons; 1st ISBN-13: 978-0471973942
[Behzad Razavi] Design of Analog CMOS Integrated Circuits 1st Edition McGraw-Hill Education; 1st 2000 ISBN-13: 978-0072380323
Electronic and Optoelectronic Properties of Semiconductor Structures (Jasprit Singh)
Linear Algebra and Its Applications with CD-ROM, Update, 3/E, David C. Lay
Introductory Chemistry, 3/E, Nivaldo J. Tro
Calculus (6th Ed., Henry Edwards & David E. Penney)
[Douglas A. Skoog, F. James Holler, Stanley R. Crouch] Principles of Instrumental Analysis 6th Edition Brooks Cole 2006 ISBN-13: 978-0495012016
College Physics (6th Ed., Jerry D Wilson, Anthony J Buffa & Bo Lou)
Technical Mathematics with Calculus (2th Ed., Dale Ewen, Joan S. Gary James E. Trefzger)
[Peter Smith, Peter Watts Jones] Stochastic Processes: An Introduction Chapman and Hall ISBN-13: 978-1420099607
[Otto Bretscher Linear Algebra with Applications, 4th Edition US Pearson;(2008) ISBN-13: 978-0136009269
[Douglas C. Montgomery] Design and Analysis of Experiments 8th Edition, Kindle Wiley;(2012) ISBN-13: 978-1118146927
[Folland, Gerald B.] Advanced Calculus 1st Edition Pearson; 1st 2001 ISBN-13: 978-0130652652
[David R. Anderson, Dennis J. Sweeney, Thomas A. Williams] Statistics for Business and Economics, 9th Edition South-Western College Pub; 9th 2005 ISBN-13: 978-0324380255
Simulation Modeling and Analysis with Expertfit Software (4th Ed.,Averill Law)
A Graphical Approach to Precalculus with Limits: A Unit Circle Approach (4th Ed., John Hornsby, Margaret L. Lial & Gary K. Rockswold)

Analysis and Performance of Fiber Composites (3rd Ed., Bhagwan Agarwal, Lawrence Broutman & K. Chandrashekhara)
[Maurice Obstfeld, Kenneth Rogoff] Foundations of International Macroeconomics (The MIT Press) 1996 ISBN-13: 978-0262150477
Physics for Scientist and Engineers, 2/E by Knight
[Eugene F. Brigham, Michael C. Ehrhardt] Financial Management: Theory & Practice 12th Edition South-Western Publishing Co. ISBN13: 9780324422696
Starting Out with Java: From Control Structures through Objects (3rd Ed., Tony Gaddis)
[Gregory L. Baker, James A. Blackburn] The Pendulum: A Case Study in Physics 1st Edition Oxford University Press 2005 ISBN-13: 978-0198567547
[Johnson, Curtis D.] Process Control Instrumentation Technology 8th Edition Pearson; 8th 2005 ISBN-13: 978-0131194571
Digital Fundamentals (10th Ed., Thomas L. Floyd)
A Graphical Approach to College Algebra (4th Ed., John Hornsby, Margaret L. Lial & Gary K. Rockswold)
[Chip Design for Submicron VLSI: CMOS Layout and Simulation, 1st Edition Cengage Learning; 1st ISBN13: 978-0-534-46629-9
[Margaret L. Lial, Raymond N. Greenwell, Nathan P. Ritchey Finite Mathematics 8th Edition Addison Wesley;(2004) ISBN-13: 978-0321228260
[Phillip E. Allen, Douglas R. Holberg] CMOS Analog Circuit Design 2nd Edition Oxford University Press, Incorporated ISBN 13: 9780195107203
[Curtis Howard D.] Fundamentals of Aircraft Structural Analysis 1st Edition Richard d Irwin 1996 ISBN-13: 978-0256192605
The 8051 Microcontroller and Embedded Systems (2nd Ed., Mazidi, Mazidi & McKinlay)
Statistical Thermodynamics and Microscale Thermophysics (Van P. Carey)
[Douglas C. Montgomery, George C. Runger] Applied Statistics and Probability for Engineers Kindle Edition + 6th Edition Wiley;(2013) ISBN-13: 978-1118539712
Principles of Money, Banking, and Financial Markets (11th Ed.,Lawrence S. Ritter, William L. Silber & Gregory F. Udell)
Ross, Westerfield, Jaffe - Solutions Manual Corporate Finance (0)
fracture mechanics ; fundamentals and applications, 3/E, by T.L. Anderson
Theory of Applied Robotics: Kinematics, Dynamics and Control (Reza N. Jazar)
Mathematics for Elementary School Teachers (4th Ed., Phares O'Daffer, Randall Charles, Thomas Cooney, John A. Dossey & Jane Schielack)
[David C. Lay] Linear Algebra and Its Applications 3rd Edition Addison Wesley; 3rd ISBN-13: 978-0201709704
[with Student Solutions Manual CD-ROM] Gary Haggard, John Schlipf, Sue Whitesides - Discrete Mathematics for Computer Science (2006, Brooks)
[William W. Hines, Douglas C. Montgomery, David M. Goldsman, Connie M. Borror] Probability and Statistics in Engineering 4th Edition Wiley; 4th 2003 ISBN-13: 978-0471240877
Financial Accounting, 7/E, by Harrison
[Bradford D. Jordan, Thomas Miller] Fundamentals of Investments 5th Edition McGraw-Hill College; 5th ISBN-13: 978-0073382357
[Randall D. Knight (Professor Emeritus)] Physics for Scientists and Engineers: A Strategic Approach with Modern Physics and MasteringPhysics 2nd Edition Benjamin Cummings; 2nd 2007 ISBN-13: 978-0321513335
[Otto Bretscher] Linear Algebra with Applications 3rd Edition Prentice Hall; 3rd 2004 ISBN-13: 978-0131453340
Ann G. Ryan, Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining - Solutions Manual to Accompany Introduction to Linear Regression Analysis (2013, Wiley)
Beginning Algebra with Applications & Visualization, 2/E, Gary K. Rockswold, Terry A. Krieger
[Paul M. Fischer, Rita H. Cheng , William J. Tayler] Advanced Accounting (8th edition) Kindle USASouth-Western College Pub ISBN 13: 9780324058789
The Essential Cosmic Perspective Media Update, 4/E, Jeffrey O. Bennett, Megan Donahue, Nicholas Schneider, Mark Voit
[Ray W Clough, Joseph Penzien] Dynamics of Structures Computers and Structures inc; 2nd Revised edition (2010) ISBN-13: 978-0923907518
[Douglas C. Montgomery] Introduction to Linear Regression Analysis, Set 5th Edition Wiley;2013 ISBN-13: 978-1118780572
[Carl S. Warren, James M. Reeve, Jonathan Duchac] Accounting, 23th Edition South-Western College Pub (2009) ISBN 13: 9780324664287
[Neil Weste, David Harris] CMOS VLSI Design: Circuits and Systems Perspective 3rd Edition ddison-Wesley Longman, Inc.2005 ISBN13: 9780321149015
Alexander McFarlane Mood, Franklin A. Graybill, Duane C. Boes - Introduction to the Theory of Statistics_ Solutions Manual (1974)
Applied Physics (9th Ed., Dale Ewen, Ronald Nelson, Neill Schurter & Erik Gundersen)
[Frederick S. Hillier, Gerald J. Lieberman] Introduction to Operations Research with Student Access Card 9th Edition McGraw-Hill Science/Engineering/Math;(2009) ISBN-13: 978-0077298340
Algebra for College Students (6th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
Data Structures and Algorithm Analysis in C++ (3rd Ed., Mark Allen Weiss)
Mathematical Proofs: A Transition to Advanced Mathematics (Gary Chartrand, Polimeni & Zhang)
[Bradford D. Jordan, Thomas Miller] Fundamentals of Investments 6th Edition 2011 McGraw-Hill College ISBN-13: 978-0073530710
[Adel S. Sedra, Kenneth C. Smith] Microelectronic Circuits 5TH Edition McGraw-Hill (2004) ISBN-13: 978-0195142525
Kinematics, Dynamics, and Design of Machinery, 2nd Ed., Waldron & Kinzel
[Douglas C. Montgomery, George C. Runger, Norma F. Hubele] Engineering Statistics 4th edition John Wiley & Sons, Inc.2007 ISBN13: 9780471735571
[John A. Dossey, Albert D. Otto , Lawrence E. Spence] Discrete Mathematics 5th Edition Pearson;(2005) ISBN-13: 978-0321305152
Business Mathematics, 11th Ed. (Charles D. Miller, Stanley A. Salzman Gary Clendenen)
[Stuart Warren, Jonathan Clayden, Nick Greeves] Organic Chemistry 2nd Edition Kindle Canada Oxford University Press; 2nd 2012 ISBN-13: 978-0199270293
Nelson D.L., Cox M.M. - Solutions Manual for Lehninger Principles of Biochemistry 5 ed. (Freeman, 2008)
Elementary Principles of Chemical Processes, 3rd Ed., by Felder,Rousseau
Probability and Statistics for Engineers (7th Ed., Johnson, Miller, Freund)
Essentials of College Algebra (Margaret L. Lial, John Hornsby & David Schneider)
[Dilip Kondepudi, Ilya Prigogine] Modern Thermodynamics: From Heat Engines to Dissipative Structures 1st Edition John Wiley & Sons; 1st 1998 ISBN-13: 978-0471973942
[C. Henry Edwards, David E. Penney] Calculus, Early Transcendentals 7th Edition Pearson; 7th ISBN-13: 978-0131569898
The Science and Engineering of Materials, 5/E, by Donald R.Askeland, Pradeep P. Phule
[Lawrence Revsine; Daniel W. Collins; W. Bruce Johnson] Financial Reporting and Analysis 3rd Edition Pearson ISBN-13:9780131430211
[Bauer, Wolfgang, Westfall, Gary] University Physics with Modern Physics 2nd Edition McGraw-Hill Education; (2013) ISBN-13: 978-0073513881
[Carl D. Meyer] Matrix Analysis and Applied Linear Algebra 1st Edition SIAM: ISBN-13: 978-0898714548
System Dynamics: Modeling and Simulation of Mechatronic Systems (4th Ed., Karnopp, Rosenberg)
Kenneth W. Whitten, Raymond E. Davis, M. Larry Peck, George G. Stanley, Wendy Keeney-Kennicutt - Solutions Manual for Whitten's Chemistry 10th (2013, Cengage Learning)
Applied Multivariate Statistical Analysis (6th Ed., Johnson & Wichern)
Beginning Algebra (10th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
Beginning and Intermediate Algebra with Applications & Visualization, 2/E, Gary K. Rockswold,Terry A. Kriege
[David Halliday, Kenneth S. Krane Robert Resnick] Physics, Volume 1, 5th Edition, Kindle Edition Wiley; 5th ISBN-13: 978-0471320579
Aircraft Control and Simulation (2nd Ed., Brian Stevens & Frank Lewis)
[Arthur Nilson, David Darwin , Charles Dolan] Design of Concrete Structures 13th Edition McGraw-Hill (2003) ISBN-13: 978-0070581999
Brief Calculus and Its Applications (11/e, Larry J Goldstein, Schneider, Lay & Asmar)
[Ron Larson, Robert P. Hostetler, Bruce H. Edwards] Calculus 8th Edition Brooks Cole; 8th 2005 ISBN-13: 978-0618502981
Engineering Materials Vol. 2 : An Introduction to Microstructures, Processing and Design (3rd Ed., Michael Ashby & David R H Jones)
Linear Algebra, 4th Ed, by Stephen H. Friedberg , Arnold J. Insel , Lawrence E. Spence
[N. Felder M. Felder Kenny, Gary] Mathematical Methods in Engineering and Physics 1st Edition Kindle Canada Wiley; 1st 2015 ISBN-13: 978-1118449608
Statics: Analysis and Design of Systems in Equilibrium (Sheppard & Tongue)
[Margaret L. Lial, Raymond N. Greenwell, Nathan P. Ritchey] Books a la Carte Edition for Calculus with Applications 10th Edition Pearson;(2011) ISBN-13:

978-0321757876

Kinematics and Dynamics of Machinery (3rd Ed., Wilson & Sadler)

[Kenneth H. Rosen] Elementary Number Theory and Its Application, 6th Edition Pearson; 6th 2010 ISBN-13: 978-0321500311

General, Organic and Biological Chemistry: Structures of Life with , 2nd ed. (Karen C. Timberlake)

[John Wilson, Tim Hunt] Molecular Biology of the Cell, 5th Edition Routledge (2008) ISBN 13: 9780815341109

Flight Performance of Fixed and Rotary Wing Aircraft (Antonio Filippone)

[Knight (Professor Emeritus), Randall D.] Physics for Scientists and Engineers: A Strategic Approach with Modern Physics (3rd Edition) Pearson, 2012 ISBN 13: 9780321740908

[Razavi, Behzad] Fundamentals of Microelectronics 1st Edition Wiley; 1st 2008 ISBN-13: 978-0471478461

[Ron Larson, Bruce H. Edwards] Calculus: Early Transcendental Functions (MindTap) 6th Edition Cengage Learning;(2014) ISBN-13: 978-1285774770

Kernel Projects for Linux (Gary Nutt)

Introduction to Graph Theory, 2nd Ed, by Douglas B. West

[David Halliday, Robert Resnick, Kenneth S. Krane] Physics, Volume 2 5th Edition John Wiley & Sons; 5th ISBN-13: 978-0471401940

[Shreve, Steven] Stochastic Calculus for Finance The Binomial Asset Pricing Model Springer; 2005 ISBN-13: 978-0387249681

[Bergersen, Birger Bergersen, Michael Plischke, Birger] Equilibrium Statistical Physics 2nd Edition World Scientific Publishing Edition 2 ISBN-13:9789810216429

Starting Out with C++: From Control Structures through Objects (5th Ed., Tony Gaddis)

Managing Business Process Flows: Principles of Operations Management(2nd Ed., Anupind, Chopra, Deshmukh, et al)

Data Structures in Java (Thomas A. Standish)

Theory of Applied Robotics: Kinematics, Dynamics and Control (Reza N.Jazar)

Principles of Managerial Finance Brief plus MyFinanceLab Student Access Kit (5th Ed., Lawrence J. Gitman)

[H. Scott Fogler] Elements of Chemical Reaction Engineering 4th Edition Prentice Hall 2005 ISBN-13: 978-0130473943

[Stephen A. Ross, Randolph Westerfield, Bradford D. Jordan] Essentials of Corporate Finance 6th Edition McGraw-Hill; 6th 2007 ASIN: B0041W2KEC

[Aslam Kassimali] Structural Analysis, SI Edition, 4th Edition, Kindle Edition CL Engineering; 4th 2010 ISBN-13: 978-0495295679

[Anil K. Chopra] Dynamics of Structures 3rd Edition TBS, Pearson, 3rd 2006 ISBN 13: 9788131713297

[Kenneth J. Waldron, G. L. Kinzel] Kinematics, Dynamics, and Design of Machinery 2nd Edition Wiley; 2 2003 ISBN-13: 978-0471244172

Elements of Chemical Reaction Engineering (4th Ed., Fogler)

Process Dynamics: Modeling, Analysis and Simulation (Wayne Bequette)

Dennis G. Zill, Michael R. Cullen, Warren S. Wright, Carol D. Wright - Solutions Manual to Accompany Advanced Engineering Mathematics (2006, Jones & Bartlett)

Hoffmann, Laurence D._ Rosen, Kenneth H. - Solutions Manual For Use With Applied Calculus For Business, Economics, And The Social And Life Sciences, Expanded. (2004, McGraw-Hill Science En

Introduction to Programming Using Visual Basic 2008 (w/VS2008 DVD), An, 7/E David I. Schneider

[Carter Robert L.] Molecular Symmetry and Group Theory Wiley 1997 ISBN-13: 978-0471149552

[Pru Marriott, J R Edwards, Howard J Mellett] Introduction to Accounting 3rd Edition SAGE Publications Ltd; 3rd 2002 ISBN-13: 978-0761970385

[Warren L. Stutzman, Gary A. Thiele] Antenna Theory and Design 2nd Edition Wiley; 2nd 1997 ISBN-13: 978-0471025900

[Taylor, Samuel Karlin, Howard M.] An Introduction to Stochastic Modeling 3rd Edition Academic Press; 3rd 1998 ISBN-13: 978-0126848878

Precalculus (4TH Ed., Margaret L. Lial, John Hornsby & David I. Schneider)

George B. Arfken, Hans J. Weber, Frank E. Harris - Instructor's Manual_ Mathematical Methods for Physicists (2014, Elsevier)

[Barry L. Nelson] Stochastic Modeling: Analysis and Simulation 2010 Dover Publications ISBN-13: 978-0486477701

Chemistry: The Molecular Nature of Matter and Change (4th Ed., Martin Silberberg)

Phase Equilibria, Phase Diagrams and Phase Transformations – Their Thermodynamic Basis (2nd Ed., Mats Hillert)

Reinforced Concrete: Mechanics and Design (5th Ed., James G. MacGregor & James K. Wight)

[Gary L. Gray, Francesco Costanzo, Michael E. Plesha] Engineering Mechanics: Dynamics 2007 McGraw-Hil ISBN-13: 978-0073358253

[Howard Curtis] Orbital Mechanics for Engineering Students 2nd Edition Butterworth-Heinemann 2nd 2009 ISBN: 9780080887845

[Stephen A. Ross, Randoloh W. Westerfield, Jeffrey Jaffe] Corporate Finance, 9th Edition McGraw-Hill/Irwin; 9th 2009 ISBN-13: 978-0073382333

College Algebra (10th Ed., Margaret L. Lial, John Hornsby & David I. Schneider)

[Stephen A. Ross, Randolph W. Westerfield, Jeffrey Jaffe] Corporate Finance, 9th Edition McGraw-Hill/Irwin; 9th ISBN-13: 978-0077398309

Margaret L. Lial ,Raymond N. Greenwell ,Nathan P. Ritchey - Instructor's Solutions Manual to Calculus with Applications (2012, Pearson)

[Dennis G. Zill, Warren S. Wright] Advanced Engineering Mathematics 5th Edition Kindle Jones & Bartlett Learning; 5th 2012 ISBN-13: 978-1449691721

Equilibrium and Non-Equilibrium Statistical Thermodynamics (Michel Le Bellac, Fabrice Mortessagne & G. George Batrouni)

Practical Interfacing in the Laboratory: Using a PC for Instrumentation, Data Analysis and Control (Stephen E. Derenzo)

Introductory Algebra (3rd Ed., K. Elayn Martin-Gay)

[Martin H. Sadd Elasticity: Theory, Applications, and Numerics 2nd Edition 2009 Academic Press ISBN-13: 978-0123744463

The Management of Construction: A Project Lifecycle Approach (F. Lawrence Bennett)

[Analysis and Design of Analog Integrated Circuits 5th Ed (vol.1) ch1-4 by Gray, Meyer

Elementary Principles of Chemical Processes (3rd Ed., Felder & Rousseau)

Applying PIC18 Microcontrollers: Architecture, Programming, and Interfacing using C and Assembly (Barry B. Brey)

Financial Reporting and Analysis,3/E Lawrence Revsine

College Algebra with Modeling and Visualization (3rd Ed., Gary K. Rockswold)

[Paul M. Fischer, William J. Taylor, Rita H. Cheng] Fundamentals of Advanced Accounting 1st Edition Cengage Learning; 1st 2007 ISBN-13: 978-0324378900

ADVANCED CALCULUS Gerald B. Folland

[Jeffry D. Madura, Carey Bissonnette, Ralph H. Petrucci, F. Geoffrey Herring] General Chemistry: Principles and Modern Applications 10th Edition Kindle Pearson Prentice Hall; 10th 2010 ISBN-13: 978-0132064521

[Harris Randy] Modern Physics 2nd Edition Pearson; 2nd 2007 ISBN-13: 978-0805303087

Advanced Topics in Finite Element Analysis of Structures: With Mathematica and MATLAB Computations (Asghar Bhatti)

Finite Mathematics, 9/E, by Margaret L. Lial , Raymond N. Greenwell , Nathan P. Ritchey

Structural Concrete: Theory and Design (3rd Ed., M. Nadim Hassoun & Akthem Al-Manaseer)

[Frank K. Reilly, Keith C. Brown] Investment Analysis and Portfolio Management South-Western; 8th 2005 ISBN-13: 978-0324289053

Aircraft Digital Electronic and Computer Systems: Principles, Operation and Maintenance (Mike Tooley)

[Jonathan Clayden, Nick Greeves, Stuart Warren, Peter Wothers] Organic Chemistry 1st Edition Oxford University Press; 1st ISBN-13: 978-0198503460

heste...@gmail.com

unread,
Sep 2, 2018, 5:57:13 AM9/2/18
□

8 3 2 . Filenames are not case sensitive. Folder: A container holding files and other folders. Folders also are known as directories. Subfolder: A folder contained inside another folder. Path: A sequence of folders, separated by backslashes (\), where each folder is a subfolder of the folder preceding it. Paths are used to identify the locations of folders and files. An example is: Programs\Ch07\Text_Files_for_Exercises

Filespec: An abbreviation of file specification, it is the combination of a drive letter followed by a colon, a path, and a filename. An example is: C:\Programs\Ch07\Text_Files_for_Exercises\USPRES.TXT

In practice, you rarely have to type a filespec, since both Windows and Visual Basic provide Browse facilities that locate files and folders for you. Windows Explorer: A program used to view, organize, and manage the folders and files on your disks. The details are presented in Appendix B in the section Manage Files and Folders with Windows Explorer. To invoke Windows Explorer, right-click the Windows Start button and click on Explore in the context menu that appears. Displaying File Extensions: By default, Windows shows only the base names of files. The following steps get Windows to also display the extensions. (In this book we assume that extensions are always shown). Windows XP 1. From Windows Explorer, press Alt/T/O to display the Folder Options dialog box. 2. Click on the View tab in the dialog box. (The dialog box in Figure 1.3 will appear.) 3. If there is a check mark in the box next to “Hide extensions for known file types,” click on the box to remove the check mark. 4. Click on the OK button to close the Folder Options dialog box. Windows Vista 1. Click on the Start button. 2. Type Folder Options into the Start Search box (just above the Start button) and press the Enter key. (The Folder Options dialog box appears.) 3. Click on the View tab in the Folder Options dialog box. (The dialog box will appear as in Figure 1.4.)

1.2

FIGURE 1.3

Windows, Folders, and Files

XP Folder Options dialog box.

4. If there is a check mark in the box next to “Hide extensions for known file types,” click on the box to remove the check mark. 5. Click on the OK button to close the Folder Options dialog box.

FIGURE 1.4

Vista Folder Options dialog box.

◆

7

8

◆

Chapter 1 An Introduction to Computers and Problem Solving

1.3

Program Development Cycle

We learned in Section 1.1 that hardware refers to the machinery in a computer system (such as the monitor, keyboard, and CPU) and software refers to a collection of instructions, called a program, that directs the hardware. Programs are written to solve problems or perform tasks on a computer. Programmers translate the solutions or tasks into a language the computer can understand. As we write programs, we must keep in mind that the computer will only do what we instruct it to do. Because of this, we must be very careful and thorough with our instructions. Note: A program is also known as a project, application, or solution.

■ Performing a Task on the Computer The first step in writing instructions to carry out a task is to determine what the output should be—that is, exactly what the task should produce. The second step is to identify the data, or input, necessary to obtain the output. The last step is to determine how to process the input to obtain the desired output, that is, to determine what formulas or ways of doing things can be used to obtain the output. This problem-solving approach is the same as that used to solve word problems in an algebra class. For example, consider the following algebra problem: How fast is a car traveling if it goes 50 miles in 2 hours? The first step is to determine the type of answer requested. The answer should be a number giving the speed in miles per hour (the output). (Speed is also called velocity.) The information needed to obtain the answer is the distance and time the car has traveled (the input). The formula speed = distance/time is used to process the distance traveled and the time elapsed in order to determine the speed. That is, speed = 50 miles/2 hours = 25 miles/hour A pictorial representation of this problem-solving process is

Input

Processing

Output

We determine what we want as output, get the needed input, and process the input to produce the desired output. In the following chapters we discuss how to write programs to carry out the preceding operations. But first we look at the general process of writing programs.

1.3

Program Development Cycle

■ Program Planning A baking recipe provides a good example of a plan. The ingredients and the amounts are determined by what is to be baked. That is, the output determines the input and the processing. The recipe, or plan, reduces the number of mistakes you might make if you tried to bake with no plan at all. Although it's difficult to imagine an architect building a bridge or a factory without a detailed plan, many programmers (particularly students in their first programming course) try to write programs without first making a careful plan. The more complicated the problem, the more complex the plan may be. You will spend much less time working on a program if you devise a carefully thought out stepby-step plan and test it before actually writing the program. Many programmers plan their programs using a sequence of steps, referred to as the program development cycle. The following step-by-step process will enable you to use your time efficiently and help you design error-free programs that produce the desired output. 1. Analyze: Define the problem. Be sure you understand what the program should do, that is, what the output should be. Have a clear idea of what data (or input) are given and the relationship between the input and the desired output. 2. Design: Plan the solution to the problem. Find a logical sequence of precise steps that solve the problem. Such a sequence of steps is called an algorithm. Every detail, including obvious steps, should appear in the algorithm. In the next section, we discuss three popular methods used to develop the logic plan: flowcharts, pseudocode, and top-down charts. These tools help the programmer break a problem into a sequence of small tasks the computer can perform to solve the problem. Planning also involves using representative data to test the logic of the algorithm by hand to ensure that it is correct. 3. Choose the interface: Select the objects (text boxes, buttons, etc.). Determine how the input will be obtained and how the output will be displayed. Then create objects to receive the input and display the output. Also, create appropriate buttons and menus to allow the user to control the program. 4. Code: Translate the algorithm into a programming language. Coding is the technical word for writing the program. During this stage, the program is written in Visual Basic and entered into the computer. The programmer uses the algorithm devised in Step 2 along with a knowledge of Visual Basic. 5. Test and debug: Locate and remove any errors in the program. Testing is the process of finding errors in a program, and debugging is the process of correcting errors that are found. (An error in a program is called a bug.) As the program is typed, Visual Basic points out certain types of program errors. Other types of errors will be detected by Visual Basic when the program is executed; however, many errors due to typing mistakes, flaws in the algorithm, or incorrect use of the Visual Basic language rules can be uncovered and corrected only by careful detective work. An example of such an error would be using addition when multiplication was the proper operation.

6. Complete the documentation: Organize all the material that describes the program. Documentation is intended to allow another person, or the programmer at a later date, to understand the program. Internal documentation (comments) consists of statements in the program that are not executed, but point out the purposes of various parts of the program. Documentation might also consist of a detailed description of what the program does and how to use the program (for instance, what type of input is expected). For commercial programs, documentation includes an instruction manual and on-line help. Other types of documentation are the flowchart, pseudocode, and hierarchy chart that were used to construct the program. Although documentation is listed as the last step in the program development cycle, it should take place as the program is being coded.

1.4

Programming Tools

This section discusses some specific algorithms and develops three tools used to convert algorithms into computer programs: flowcharts, pseudocode, and hierarchy charts. You use algorithms every day to make decisions and perform tasks. For instance, whenever you mail a letter, you must decide how much postage to put on the envelope. One rule of thumb is to use one stamp for every five sheets of paper or fraction thereof. Suppose a friend asks you to determine the number of stamps to place on an envelope. The following algorithm will accomplish the task. 1. Request the number of sheets of paper; call it Sheets. 2. Divide Sheets by 5. 3. Round the quotient up to the next highest whole number; call it Stamps. 4. Reply with the number Stamps.

(input) (processing) (processing) (output)

The preceding algorithm takes the number of sheets (Sheets) as input, processes the data, and produces the number of stamps needed (Stamps) as output. We can test the algorithm for a letter with 16 sheets of paper. 1. 2. 3. 4.

Request the number of sheets of paper; Sheets = 16. Dividing 5 into 16 gives 3.2. Rounding 3.2 up to 4 gives Stamps = 4. Reply with the answer, 4 stamps. This problem-solving example can be pictured by

Input (16)

Processing (formulas)

Output (4)

1.4

Programming Tools

Of the program design tools available, three popular tools are the following: Flowcharts: Graphically depict the logical steps to carry out a task and show how the steps relate to each other. Pseudocode: Uses English-like phrases with some Visual Basic terms to outline the task. Hierarchy charts: Show how the different parts of a program relate to each other.

■ Flowcharts A flowchart consists of special geometric symbols connected by arrows. Within each symbol is a phrase presenting the activity at that step. The shape of the symbol indicates the type of operation that is to occur. For instance, the parallelogram denotes input or output. The arrows connecting the symbols, called flowlines, show the progression in which the steps take place. Flowcharts should "flow" from the top of the page to the bottom. Although the symbols used in flowcharts are standardized, no standards exist for the amount of detail required within each symbol.

Symbol	Name	Meaning
	Flowline	Used to connect symbols and indicate the flow of logic.
	Terminal	Used to represent the beginning (Start) or the end (End) of a task.
	Input/Output	Used for input and output operations, such as reading and displaying. The data to be read or displayed are described inside.
	Processing	Used for arithmetic and data-manipulation operations. The instructions are listed inside the symbol.
	Decision	Used for any logic or comparison operations. Unlike the input/ouput and processing symbols, which have one entry and one exit flowline, the decision symbol has one entry and two exit paths. The path chosen depends on whether the answer to a question is "yes" or "no."
	Connector	Used to joint different flowlines.
	Offpage Connector	Used to indicate that the flowchart continues to a second page.
	Predefined Process	Used to represent a group of statements that perform one processing task.
	Annotation	Used to provide additional information about another flowchart symbol.

Start

Read sheets

input

Set stamps sheets / 5

processing

Round stamps up to next whole number

processing

Display stamps

output

End FIGURE 1.5

Flowchart for the postage stamp problem.

The table of the flowchart symbols shown on the previous page has been adopted by the American National Standards Institute (ANSI). Figure 1.5 shows the flowchart for the postage stamp problem. The main advantage of using a flowchart to plan a task is that it provides a pictorial representation of the task, which makes the logic easier to follow. We can clearly see every step and how each is connected to the next. The major disadvantage with flowcharts is that when a program is very large, the flowcharts may continue for many pages, making them difficult to follow and modify.

■ Pseudocode Pseudocode is an abbreviated plain English version of actual computer code (hence, pseudocode). The geometric symbols used in flowcharts are replaced by English-like statements that outline the process. As a result, pseudocode looks more like computer code than does a flowchart. Pseudocode allows the programmer to focus on the steps required to solve a problem rather than on how to use the computer language. The

programmer can describe the algorithm in Visual Basic-like form without being restricted by the rules of Visual Basic. When the pseudocode is completed, it can be easily translated into the Visual Basic language. The following is pseudocode for the postage stamp problem:

```
Program: Determine the proper number of stamps for a letter
Read Sheets
Set the number of stamps to Sheets / 5
Round the number of stamps up to the next whole number
Display the number of stamps
```

(input) (processing) (processing) (output)

Pseudocode has several advantages. It is compact and probably will not extend for many pages as flowcharts commonly do. Also, the plan looks like the code to be written and so is preferred by many programmers.

■ Hierarchy Chart The last programming tool we'll discuss is the hierarchy chart, which shows the overall program structure. Hierarchy charts are also called structure charts, HIPO (Hierarchy plus Input-Process-Output) charts, top-down charts, or VTOC (Visual Table of Contents) charts. All these names refer to planning diagrams that are similar to a company's organization chart. Hierarchy charts depict the organization of a program but omit the specific processing logic. They describe what each part, or module, of the program does and they show how the modules relate to each other. The details on how the modules work, however, are omitted. The chart is read from top to bottom and from left to right. Each module may be subdivided into a succession of submodules that branch out under it. Typically, after the activities in the succession of submodules are carried out, the module to the right of the original module is considered. A quick glance at the hierarchy chart reveals each task performed in the program and where it is performed. Figure 1.6 shows a hierarchy chart for the postage stamp problem.

Postage stamp program

Read sheets

Calculate stamps

Set stamps sheets / 5 FIGURE 1.6

Display stamps

Round stamps up to next whole number

Hierarchy chart for the postage stamp problem.

The main benefit of hierarchy charts is in the initial planning of a program. We break down the major parts of a program so we can see what must be done in general. From this point, we can then refine each module into more detailed plans using flowcharts or pseudocode. This process is called the divide-and-conquer method. The postage stamp problem was solved by a series of instructions to read data, perform calculations, and display results. Each step was in a sequence; that is, we moved from one line to the next without skipping over any lines. This kind of structure is called a sequence structure. Many problems, however, require a decision to determine whether a series of instructions should be executed. If the answer to a question is "Yes," then one group of instructions is executed. If the answer is "No," then another is executed. This structure is called a decision structure. Figure 1.7 contains the pseudocode and flowchart for a decision structure.

No

```
If condition is true Then
    Process step(s) 1
Else
    Process step(s) 2
End If
```

FIGURE 1.7

Is condition true?

Yes

Process step(s) 2

Process step(s) 1

Pseudocode and flowchart for a decision structure.

The sequence and decision structures are both used to solve the following problem.

■ Direction of Numbered NYC Streets Algorithm Problem: Given a street number of a one-way street in New York, decide the direction of the street, either eastbound or westbound. Discussion: There is a simple rule to tell the direction of a one-way street in New York: Even-numbered streets run eastbound. Input: Street number Processing: Decide if the street number is divisible by 2. Output: “Eastbound” or “Westbound” Figures 1.8 through 1.10 show the flowchart, pseudocode, and hierarchy chart for the New York City numbered streets problem. The solution to the next problem requires the repetition of a series of instructions. A programming structure that executes instructions many times is called a loop structure. We need a test (or decision) to tell when the loop should end. Without an exit condition, the loop would repeat endlessly (an infinite loop). One way to control the number of times a loop repeats (often referred to as the number of passes or iterations)

Start

Get street

No

Is street even?

Display westbound

Yes

Display eastbound

End

FIGURE 1.8

Flowchart for the New York City numbered streets problem.

Program: Determine the direction of a numbered NYC street. Get street If street is even Then Display Eastbound Else Display Westbound End If FIGURE 1.9

Pseudocode for the New York City numbered streets problem.

Street direction program

Get street number FIGURE 1.10

Decide whether street number is even or odd

Display direction

Hierarchy chart for the New York City numbered streets problem.

Is condition true? Do While condition is true Process step(s) Loop

No

Yes

Process step(s)

FIGURE 1.11

Pseudocode and flowchart for a loop.

is to check a condition before each pass through the loop and continue executing the loop as long as the condition is true. See Figure 1.11.

■ Class Average Algorithm Problem: Calculate and report the grade-point average for a class. Discussion: The average grade equals the sum of all grades divided by the number of students. We need a loop to read and then add (accumulate) the grades for each student in the class. Inside the loop, we also need to total (count) the number of students in the class. See Figures 1.12 to 1.14. Input: Student grades Processing: Find the sum of the grades; count the number of students; calculate average grade = sum of grades / number of students. Output: Average grade

■ Comments 1. Tracing a flowchart is like playing a board game. We begin at the Start symbol and proceed from symbol to symbol until we reach the End symbol. At any time, we will be at just one symbol. In a board game, the path taken depends on the result of spinning a spinner or rolling a pair of dice. The path taken through a flowchart depends on the input. 2. The algorithm should be tested at the flowchart stage before being coded into a program. Different data should be used as input, and the output checked. This process is known as desk checking. The test data should include nonstandard data as well as typical data.

Start

Initialize counter and sum to 0

Is there more data?

counter and sum start at 0

No

Yes Get next grade

read next grade

Increment counter

add 1 to counter

Add grade to sum

accumulate sum of grades

Set average to sum/counter

find the average

Display average

display the answer

End FIGURE 1.12

Flowchart for the class average problem.

3. Flowcharts, pseudocode, and hierarchy charts are universal problem-solving tools. They can be used to construct programs in any computer language, not just Visual Basic. 4. Flowcharts are used throughout this text to provide a visualization of the flow of certain programming tasks and Visual Basic control structures. Major examples of pseudocode and hierarchy charts appear in the case studies.

Program: Calculate and report the average grade of a class. Initialize Counter and Sum to 0 Do While there is more data Get the next Grade Increment the Counter Add the Grade to the Sum Loop Compute Average = Sum/Counter Display Average FIGURE 1.13

Pseudocode for the class average problem.

Class average problem

Get grade

Compute sum and number of grades FIGURE 1.14

Calculate average

Display average

Hierarchy chart for the class average problem.

5. There are four primary logical programming constructs: sequence, decision, loop, and unconditional branch. Unconditional branch, which appears in some languages as GoTo statements, involves jumping from one place in a program to another. Structured programming uses the first three constructs but forbids the fourth. One advantage of pseudocode over flowcharts is that pseudocode has no provision for unconditional branching and thus forces the programmer to write structured programs. 6. Flowcharts are time consuming to write and difficult to update. For this reason, professional programmers are more likely to favor pseudocode and hierarchy charts. Because flowcharts so clearly illustrate the logical flow of programming techniques, however, they are a valuable tool in the education of programmers. 7. There are many styles of pseudocode. Some programmers use an outline form, whereas others use a form that looks

almost like a programming language. The pseudocode appearing in the case studies of this text focuses on the primary tasks to be performed by the program and leaves many of the routine details to be completed during the coding process. Several Visual Basic keywords, such as, If, Else, Do, and While, are used extensively in the pseudocode appearing in this text. 8. Many people draw rectangles around each item in a hierarchy chart. In this text, rectangles are omitted in order to make hierarchy charts easier to draw and thereby to encourage their use.

2 Visual Basic, Controls, and Events 2.1

An Introduction to Visual Basic 2008 20 ◆ ◆

2.2

Why Windows and Why Visual Basic? The Different Versions of Visual Basic

◆

How You Develop a Visual Basic Program

Visual Basic Controls 22 Starting a New Visual Basic Program ◆ A Text Box Walkthrough ◆ A Button Walkthrough ◆ A Label Walkthrough ◆ A List Box Walkthrough ◆ The Name Property ◆ Two Help Walkthroughs ◆ Fonts ◆ Auto Hide ◆ Positioning and Aligning Controls ◆ Setting Tab Order ◆

2.3

Visual Basic Events 44 An Event Procedure Walkthrough ◆ Properties and Event Procedures of the Form ◆ The Header of an Event Procedure ◆ Context-Sensitive Help ◆ Text Files ◆ Viewing All Possible Event Procedures for a Control ◆ Opening a Program ◆

Summary 60

2.1

An Introduction to Visual Basic 2008

Visual Basic 2008 is the latest generation of Visual Basic, a language used by millions of software developers. Visual Basic was designed to make user-friendly programs easier to develop. Prior to the creation of Visual Basic, developing a friendly user interface usually required a programmer to use a language such as C or C++, often requiring hundreds of lines of code just to get a window to appear on the screen. Now the same program can be created with much less time and fewer instructions.

■ Why Windows and Why Visual Basic? What people call graphical user interfaces, or GUIs (pronounced “gooies”), have revolutionized the computer industry. Instead of the confusing textual prompts that earlier users once saw, today’s users are presented with such devices as icons, buttons, and drop-down lists that respond to mouse clicks. Accompanying the revolution in how programs look was a revolution in how they feel. Consider a program that requests information for a database. Figure 2.1 shows how a program written before the advent of GUIs got its information. The program requests the six pieces of data one at a time, with no opportunity to go back and alter previously entered information. Then the screen clears and the six inputs are again requested one at a time. Enter name (Enter EOD to terminate): Mr. President Enter Address: 1600 Pennsylvania Avenue Enter City: Washington Enter State: DC Enter Zipcode: 20500 Enter Phone Number: 202-456-1414 FIGURE 2.1

Input screen of a pre-Visual Basic program to fill a database.

Figure 2.2 shows how an equivalent Visual Basic program gets its information.The boxes may be filled in any order. When the user clicks on a box with the mouse, the cursor moves to that box. The user can either type in new information or edit the existing information. When the user is satisfied that all the information is correct, he or she just clicks on the “Write to Database” button. The boxes will clear, and the data

FIGURE 2.2

Input screen of a Visual Basic program to fill a database.

for another person can be entered. After all names have been entered, the user clicks on the Exit button. In Figure 2.1, the program is in control; in Figure 2.2, the user is in control!

■ How You Develop a Visual Basic Program One of the key elements of planning a Visual Basic program is deciding what the user sees—in other words, designing the screen. What data will he or she be entering? How large a window should the program use? Where will you place the buttons the user clicks on to activate actions by the program? Will the program have places to enter text (text boxes) and places to display output? What kind of warning boxes (message boxes) should the program use? In Visual Basic, the responsive objects a program designer places on windows are called controls. Two features make Visual Basic different from traditional programming tools: 1. You literally draw the user interface, much like using a paint program. 2. Perhaps more important, when you’re done drawing the interface, the buttons, text boxes, and other objects that you have placed in a blank window will automatically recognize user actions such as mouse movements and button clicks. That is, the sequence of procedures executed in your program is controlled by “events” that the user initiates rather than by a predetermined sequence of procedures in your program. In any case, only after you design the interface does anything like traditional programming occur. Objects in Visual Basic recognize events like mouse clicks; how the objects respond to them depends on the instructions you write. You always need to write instructions in order to make controls respond to events. This makes Visual Basic programming fundamentally different from conventional programming. Programs in traditional programming languages ran from the top down. For these programming languages, execution started from the first line and moved with the flow of the program to different parts as needed. A Visual Basic program works differently. Its core is a set of independent groups of instructions that are activated by the events they have been told to recognize. This event-driven methodology is a fundamental shift. The user decides the order in which things happen, not the programmer. Most of the programming instructions in Visual Basic that tell your program how to respond to events like mouse clicks occur in what Visual Basic calls event procedures. Essentially, anything executable in a Visual Basic program either is in an event procedure or is used by an event procedure to help the procedure carry out its job. In fact, to stress that Visual Basic is fundamentally different from traditional programming languages, Microsoft uses the term project, rather than program, to refer to the combination of programming instructions and user interface that makes a Visual Basic program possible. Here is a summary of the steps you take to design a Visual Basic program: 1. Design the appearance of the window that the user sees. 2. Determine the events that the controls on the window should recognize. 3. Write the event procedures for those events.

Now here is what happens when the program is running: 1. Visual Basic monitors the controls in the window to detect any event that a control can recognize (mouse movements, clicks, keystrokes, and so on). 2. When Visual Basic detects an event, it examines the program to see if you've written an event procedure for that event. 3. If you have written an event procedure, Visual Basic executes the instructions that make up that event procedure and goes back to Step 1. 4. If you have not written an event procedure, Visual Basic ignores the event and goes back to Step 1. These steps cycle continuously until the program ends. Usually, an event must happen before Visual Basic will do anything. Event-driven programs are reactive more than active—and that makes them more user friendly.

■ The Different Versions of Visual Basic Visual Basic 1.0 first appeared in 1991. It was followed by version 2.0 in 1992, version 3.0 in 1993, version 4.0 in 1995, version 5.0 in 1997, and version 6.0 in 1998. VB.NET, initially released in February 2002, was not backward compatible with the earlier versions of Visual Basic. It incorporated many features requested by software developers, such as true inheritance and powerful Web capabilities. Visual Basic 2005, released in November 2005, and Visual Basic 2008, released in November 2007 are significantly improved versions of VB.NET.

2.2

Visual Basic Controls

Visual Basic programs display a Windows-style screen (called a form) with boxes into which users type (and in which users edit) information and buttons that they click to initiate actions. The boxes and buttons are referred to as controls. In this section, we examine forms and four of the most useful Visual Basic controls.

■ Starting a New Visual Basic Program For our purposes, Visual Basic programs are also known as applications, solutions, or projects. Each program is saved (as several files and subfolders) in its own folder. Before starting a new program, you should use Windows Explorer to create a folder to hold the folders for your programs. The process for invoking Visual Basic varies slightly with the edition of Visual Basic installed on the computer. To invoke Visual Basic from a computer that has Visual Basic Express installed, click the Windows Start button, hover over All Programs, and then click on Microsoft Visual Basic 2008 Express Edition. With the other editions of Visual Basic, hover over All Programs, hover over Microsoft Visual Studio 2008, and then click on Microsoft Visual Studio 2008 in the short list that is revealed. The window that appears after Visual Basic is invoked is called the Start Page and is similar to the window shown in Figure 2.3. (The information in the large section in the middle of the screen comes from the Microsoft Web site and will differ each time Visual Basic is invoked.) The first item on the Start Page menu bar is "File." Click on File, and then click on New Project to produce a New Project dialog box. Figure 2.4 shows the

Menu bar

FIGURE 2.3

The Visual Basic 2008 Express Edition Start Page.

New Project dialog box produced by Visual Basic Express. The "Windows Forms Application" icon should be selected as the installed template. If this is not the case, click on "Windows Forms Application" to select it. (The other editions of Visual Basic contain a pane identifying a Project type. You should select "Visual Basic" as the Project type.) Note: The number of project types and icons showing will vary depending on the version of Visual Basic you are using. Figure 2.4 was created from the Express Edition.

FIGURE 2.4

The Visual Basic Express New Project dialog box.

The name of the program, initially set to WindowsApplication1, can be specified at this time. Since we will have a chance to change the name later, let's just call the program WindowsApplication1 for now. Click on the OK button to invoke the Visual Basic programming environment See Figure 2.5. The Visual Basic programming environment is referred to as the Integrated Development Environment or IDE. Very likely, your screen will look different than Figure 2.5.

The IDE is extremely configurable. Each window in Figure 2.5 can have its location and size altered. New windows can be displayed in the IDE, and any window can be closed or hidden behind a tab. For instance, in Figure 2.5 the Toolbox window is hidden behind a tab. The View menu is used to add additional windows to the IDE. If you would like your screen to look exactly like Figure 2.5, click on "Reset Windows Layout" in the Windows menu, and then click on "Yes." The Menu bar of the IDE displays the commands you use to work with Visual Basic. Some of the menus, like File, Edit, View, and Window, are common to most Windows applications. Others, such as Project, Build, and Debug, provide commands specific to programming in Visual Basic. The Toolbar holds a collection of icons that carry out standard operations when clicked. For example, you use the fifth icon, which looks like a stack of three diskettes, to save the files associated with the current program. To reveal the purpose of a Toolbar icon, hover the mouse pointer over the icon for a few seconds. The little information rectangle that pops up is called a tooltip. The Main area currently holds the rectangular Form window, or form for short. The form becomes a Windows window when a program is executed. Most information displayed by the program appears on the form. The information usually is displayed in Start Page tab

Form Designer tab

Main area

Properties window

Solution Explorer window

Menu bar Toolbar Toolbox tab

Form

Sizing handle

FIGURE 2.5

The Visual Basic integrated development environment in Form Designer mode.

2.2

Visual Basic Controls

controls that the programmer has placed on the form.You can change the size of the form by dragging one of its sizing handles. The Properties window is used to change how objects look and react. The Solution Explorer window displays the files associated with the program and provides access to the commands that pertain to them. (Note: If the Solution Explorer or the Properties window is not visible, click on it in the View menu.) The Toolbox holds icons representing objects (called controls) that can be placed on the form. If your screen does not show the Toolbox, hover the mouse over the Toolbox tab at the left side of the screen. The Toolbox will slide into view. Then click on the pushpin icon in the title bar at the top of the Toolbox to keep the Toolbox permanently displayed in the IDE. (Note: If there is no tab marked Toolbox, click on Toolbox in the View menu.) The controls in the Toolbox are grouped into categories such as All Windows Forms and Common Controls. Figure 2.6 shows the Toolbox after the plus sign to the left of "Common Controls" has been clicked. Nearly all the controls discussed in this text can be found in the list of common controls. (You can obtain a description of a control by hovering the mouse over the control.) The four controls discussed in this chapter are text boxes, labels, buttons, and list boxes. In order to see all the group names, click on each of the minus signs appearing to the left of a group name.

Pushpin

Group names

FIGURE 2.6

The Toolbox's common controls.

◆

◆

Text boxes: You use a text box to get information from the user, referred to as input, or to display information produced by the program, referred to as output. Labels: You place a label near a text box to tell the user what type of information is displayed in the text box. Buttons: The user clicks a button to initiate an action. List boxes: In the first part of the book, we use list boxes to display tables or several lines of output. Later, we use list boxes to make selections.

■ A Text Box Walkthrough 1. Double-click on the text box icon in the Common Controls portion of the Toolbox. A rectangle with two small squares, called sizing handles, appears at the upper left corner of the form. (You can alter the width of the text box by dragging one of the sizing handles.) Move the mouse arrow to any point of the text box other than a sizing handle, hold down the left mouse button, and drag the text box to the center of the form. See Figure 2.7. Note: The Tasks button will be discussed in Chapter 3. Tasks button

Sizing handles FIGURE 2.7

A text box with sizing handles.

2. Click anywhere on the form outside the rectangle to deselect the text box. 3. Click on the rectangle to restore the handles. An object showing its handles is said to be selected. A selected text box can have its width altered, location changed, and other properties modified. 4. Move the mouse arrow to the handle in the center of the right side of the text box. The cursor should change to a double arrow 142. Hold down the left mouse button, and move the mouse to the right. The text box is stretched to the right. Similarly, grabbing the text box on the left side and moving the mouse to the left stretches the text box to the left. You also can use the handles to make the text box smaller. Steps 1 and 4 allow you to place a text box of any width anywhere on the form. Note: The text box should now be selected; that is, its sizing handles should be showing. If not, click anywhere inside the text box to select it. 5. Press the delete key, Del, to

remove the text box from the form. Step 6 gives an alternative way to place a text box of any width at any location on the form. 6. Click on the text box icon in the Toolbox. Then move the mouse pointer to any place on the form. (When over the form, the mouse pointer becomes a pair of crossed thin lines.) Hold down the left mouse button, and drag the mouse on a diagonal to generate a rectangle. Release the mouse button to obtain a selected

text box. You can now alter the width and location as before. Note: The text box should now be selected. If not, click anywhere inside the text box to select it. 7. Press F4 to activate the Properties window. (You also can activate the Properties window by clicking on it, clicking on the Properties window icon in the right part of the Toolbar, selecting Properties Window from the View menu, or clicking on the text box with the right mouse button and selecting Properties.) See Figure 2.8. The first line of the Properties window (called the Object box) reads “TextBoxl etc.” TextBoxl is the current name of the text box. The first two buttons below the Object box permit you to view the list of properties either grouped into categories or alphabetically. Use the up- and down-arrow keys (or the up- and down-scroll arrows) to move through the list. The left column gives the property names, and the right column gives the current settings of the properties. We discuss four properties in this walkthrough.

Object box

Description pane

Categorized view

FIGURE 2.8

Alphabetic view

Properties window icon

Text box Properties window.

Note 1: The third and fourth buttons below the Object box, the Properties button and the Events button, determine whether properties or events are displayed in the Properties window. Normally the Properties button is highlighted. If not, click on the Properties button. Note 2: If the Description pane is not visible, right-click on the Properties window, and then click on “Description.” The Description pane describes the currently highlighted property.

8. Move to the Text property with the up- and down-arrow keys. (Alternatively, scroll until the property is visible, and click on the property.) The Text property, which determines the words displayed in the text box, is now highlighted. Currently, there is no text displayed in the Settings box on the right. 9. Type your first name. Then press the Enter key, or click on another property. Your name now appears in both the Settings box and the text box. See Figure 2.9.

FIGURE 2.9

Setting the Text property to David.

10. Click at the beginning of your name in the Text Settings box, and add your title, such as Mr., Ms., or The Honorable. (If you mistyped your name, you can easily correct it now.) Then, press Enter. 11. Use the up-arrow key or the mouse to move to the ForeColor property. This property determines the color of the information displayed in the text box. 12. Click on the down arrow in the right part of the Settings box, and then click on the Custom tab to display a selection of colors. See Figure 2.10. Click on one of the

FIGURE 2.10

Setting the ForeColor property.

colors, such as blue or red. Notice the change in the color of your name. (Note: The sixteen white boxes at the bottom of the grid are used to create custom colors. See item J under “Design a Form” in Appendix B for details.) 13. Highlight the Font property with a single click of the mouse. The current font is named Microsoft Sans Serif. 14. Click on the ellipsis 1 Á 2 box in the right part of the Settings box to display a dialog box. See Figure 2.11. The three lists give the current name (Microsoft Sans Serif), current style (Regular), and current size (8) of the font. You can change any of these attributes by clicking on an item in its list or by typing into the box at the top of the list. Click on Bold in the style list, and click on 12 in the size list. Now click on the OK button to see your name displayed in a larger bold font. The text box will be longer so that it can accommodate the larger font.

FIGURE 2.11

The Font dialog box.

15. Click on the text box and resize it to be about 3 inches wide. Visual Basic programs consist of three parts: interface, values of properties, and code. Our interface consists of a form with a single object, a text box. We have set a few properties for the text box—the text (namely, your name), the foreground color, the font style, and the font size. In Section 2.3, we discuss how to place code into a program. Visual Basic endows certain capabilities to programs that are independent of any code we will write. We will now run the existing program without adding any code in order to experience these capabilities.

16. Press F5 to run the program. (Alternatively, a program can be run from the menu by pressing Alt/D/S or by clicking on the Start Debugging icon, the fourteenth or fifteenth icon on the Toolbar.) After a brief delay, a copy of the form appears that has neither the form nor the text box selected. 17. Your name is highlighted. Press the End key to move the cursor to the end of your name. Now type in your last name, and then keep typing. Eventually, the words will scroll to the left. 18. Press Home to return to the beginning of the text. You have a miniature word processor at your disposal. You can place the cursor anywhere you like to add or delete text. You can drag the cursor across text to select a block, place a copy of the block in the Clipboard with Ctrl+C, and then duplicate it elsewhere with Ctrl+V. 19. To end the program, press Alt + F4. Alternatively, you can end a program by clicking on the form's Close button at the right corner of the title bar. 20. Select the text box, activate the Properties window, select the ReadOnly property, click on the down-arrow button, and finally click on True. Notice that the background color is now gray. 21. Run the program, and try typing into the text box. You can't. Such a text box is used for output. Only code can display information in the text box. Note: In this textbook, whenever a text box will be used only for the purpose of displaying output, we will always set the ReadOnly property to True. 22. End the program. 23. Let's now save the program on a disk. Click on the Save All icon to save the work done so far. (The Save All icon is the fifth or sixth icon on the Toolbar. It shows three fanned diskettes. Alternately, you can click on Save All in the File menu.) You will be prompted for the name of the program and the path to the folder where you want the program to be saved. Type a name, such as "VBdemo". You can either type a path or use Browse to locate a folder. (This folder will automatically be used the next time you click on the Save All icon.) The files for the program will be held in a subfolder of the selected folder. Important: If the "Create directory for solution" check box is checked, then click on the check box to uncheck it. Finally, click on the Save button. 24. Create a new program as before by clicking on "New Project" on the File menu. (Or, you can click on the New Project icon, the first icon on the Toolbar.) A New Project dialog box will appear. 25. Give a name to the project, such as My Program, and then click on the OK button. 26. Place three text boxes on the form. (If you use the double-click technique, move the text boxes so that they do not overlap.) Notice that they have the names TextBox1, TextBox2, and TextBox3. 27. Run the program. Notice that the cursor is in TextBox1. We say that TextBox1 has the focus. (This means that TextBox1 is the currently selected object and any keyboard actions will be sent directly to this object.) Any text typed will display in that text box.

28. Press Tab once. Now, TextBox2 has the focus. When you type, the characters appear in TextBox2. 29. Press Tab several times, and then press Shift+Tab a few times. With Tab, the focus cycles through the objects on the form in the order the objects were created. With Shift+Tab, the focus cycles in the reverse order. 30. End the program you created. 31. We would now like to return to the first program. Click on "Open Project" from the File menu. An Open Project dialog box will appear stating that "You must choose to either save or discard changes in the current project before opening a project." There is no need to save this program, so click on the Discard button. Then a second Open Project dialog box will appear. 32. Find the folder corresponding to the program you saved earlier, double-click on the folder, and double-click on the file with extension sln. You have now recovered the first program. Note: As an alternative to using the Open Project dialog box in Steps 31 and 32 to return to the first program, click on the Start Page tab at the top of the Main area, and click on the program in the Recent Projects pane of the Start Page. 33. If you do not see the Form Designer for the program, double-click on Form1.vb in the Solution Explorer.

■ A Button Walkthrough 1. Click on the New Project icon to start a new program. (Give a name, such as ButtonProg, to the program, and click on OK.) 2. Double-click on the Button icon in the Toolbox to place a button on the form. (The Button icon is the second icon in the Common Controls portion of the Toolbox.) 3. Move the button to the center of the form. 4. Activate the Properties window, highlight the Text property, type "Please Push Me," and press Enter. See Figure 2.12. The button is too small.

FIGURE 2.12

Setting the Text property.

5. Click on the button to select it, and then enlarge it to accommodate the phrase "Please Push Me" on one line. 6. Run the program, and click on the button. The button appears to move in and then out. In Section 2.3, we will write code that is executed when a button is pushed. 7. End the program and select the button. 8. From the Properties window, edit the Text setting by inserting an ampersand (&) before the first letter, P. Press the Enter key, and notice that the first letter P on the button is now underlined. See Figure 2.13. Pressing Alt+ P while the program is running triggers the same event as clicking the button. However, the button will not appear to move in and out. Here, P is referred to as the access key for the button. (The access key is always specified as the character following the ampersand.)

FIGURE 2.13

Designating P as an access key.

■ A Label Walkthrough 1. Click on the New Project icon to begin a new program. Feel free to select the default name, such as WindowsApplication1. 2. Double-click on the label icon to place a label on the form. (The label icon is a large letter A.) Move the label to the center of the form. 3. Activate the Properties window, highlight the Text property, type "Enter Your Phone Number:", and press Enter. (Such a label would be placed next to a text box into which the user will type a phone number.) Notice that the label widened to accommodate the text. This happened because the AutoSize property of the label is set to True by default. 4. Change the AutoSize property to False. Press Enter. Notice that the label now has eight sizing handles when selected. 5. Make the label narrower and longer until the words occupy two lines. 6. Activate the Properties window, and click on the down arrow to the right of the setting for the TextAlign property. Experiment by clicking on the various rectangles and observing their effects. The combination of sizing and alignment permits you to design a label easily.

7. Run the program. Nothing happens, even if you click on the label. Labels just sit there. The user cannot change what a label displays unless you write code to make the change. 8. End the program.

■ A List Box Walkthrough 1. Click on the New Project icon to begin a new program. Feel free to select the default name, such as WindowsApplication1. 2. Place a list box on the form. (The list box icon is the ninth icon in the Common Controls group of the Toolbox.) 3. Press F4 to activate the Properties window and notice that the list box does not have a Text property. The word ListBox1 is actually the setting for the Name property. 4. Also place a text box, a button and a label on the form. 5. Click on the Object box of the Properties window. The name of the form and the names of the four controls are displayed. If you click on one of the names, that object will become selected and its properties displayed in the Properties window. 6. Run the program. Notice that the word ListBox1 has disappeared, but the words Button1 and Label1 are still visible. The list box is completely blank. In subsequent sections, we will write code to place information into the list box.

■ The Name Property Every control has a Name property. It is used in code to refer to the control. By default, controls are given names like TextBox1 and TextBox2. You can use the Properties window to change the Name property of a control to a more meaningful name. (The Name property is always the third property in the alphabetized list of properties. A control's name must begin with a letter and can be a maximum of 215 characters. It can include numbers and underline (_) characters, but cannot include punctuation or spaces.) Also, it is good coding practice to have each name begin with a three- or four-letter prefix that identifies the type of the control. See Table 2.1. The form itself also has a Name property. Beginning with Section 2.3, we will use suggestive names and these prefixes whenever possible.

TABLE 2.1

Some controls and their three-letter prefixes.

Control	Prefix	Example
form	frm	frmLottery
button	btn	btnComputeTotal
label	lbl	lblInstructions
list box	lst	lstOutput
text box	txt	txtAddress

The Name property of the form itself also can be changed. Suppose you want to change the name of the form from Form1 to frmPayroll. The most efficient way to set the name of the form is to change the name of the file Form1.vb appearing in the Solution Explorer window to frmPayroll.vb. To make the change, right-click on Form1.vb in the

Solution Explorer window, click on Rename, type in the new name (frmPayroll.vb), and press Enter. Important: Make sure that the new filename has the extension "vb". The Name and Text properties of a button are both initially set to something like Button1. However, changing one of these properties does not affect the setting of the other property, and similarly for the Name and Text properties of forms, text boxes, and labels. The Text property of a form specifies the words appearing in the form's title bar.

■ Two Help Walkthroughs The Help menu provides four ways for you to obtain information about Visual Basic. You can browse topic titles using the Contents window, search for topics by keyword using the Index window, search the full text of topics using the Search page, or browse for topics by category using How Do I. The following walkthroughs demonstrate the Index and How Do I features. Index Walkthrough 1. Click on Index in the Help menu to invoke the Index window. 2. If the "Filtered by:" box does not say "Visual Basic" or "Visual Basic Express Edition," click on the down arrow and select one of them from the drop-down list. 3. Type "buttons" into the "Look for" box, look down the list of subheadings of "buttons," and then click on "introducing." (The page that appears has the title "Interacting with the User: Using Buttons.") 4. Read the first few paragraphs of the discussion. "How Do I" Walkthrough 1. Click on "How Do I" in the Help menu. The page that appears contains a short list of underlined links. 2. Click on the underlined link "Learn the Visual Basic Language (How Do I in Visual Basic Express)." The page that appears contains another list of underlined links under the heading "In Visual Basic Express." 3. Click on the underlined link "Introduction to the Visual Basic Programming Language." 4. Read the introductory discussion of Visual Basic in the page that appears.

■ Fonts The default font for controls is Microsoft Sans Serif. Two other useful fonts are Courier New and Wingdings. Courier New is a fixed-width font; that is, each character has the same width. With such a font, the letter i occupies the same space as the letter m. Fixed-width fonts are used with tables when information is to be aligned in columns.

The Wingdings font consists of assorted small pictures and symbols, each corresponding to a character on the keyboard. For instance if one of the characters %, (, 1, or J is typed into the Text setting of a control whose Font is Wingdings, the control will display a bell, phone, open folder, or smiling face, respectively. To view the character set for a Windows font, click on the Windows Start button in the Windows task bar and successively select All Programs, Accessories, System Tools, and Character Map. Then click on Character Map, or press the Enter key. After selecting the font, click on any item to enlarge it. You can insert the keyboard character for the item into the Clipboard by pressing the Select button and then the Copy button. To place the character into the Text property of a control having that font, just move the cursor to the Settings box for the Text property and press Ctrl+ V.

■ Auto Hide The Auto Hide feature allows you to make more room for the Main area of the screen by hiding windows (such as the Toolbox, Solution Explorer, or Properties window). Let's illustrate the feature with a walkthrough using the Toolbox window. We start by discussing the situation where the feature is disabled. 1. If the Toolbox window is currently visible and the pushpin icon in the window title is vertical, then the Auto Hide feature is disabled. (If the Toolbox window is not visible, press Alt/V/X to select Toolbox from the View menu. If the pushpin icon is horizontal, then click on the icon to make it vertical.) When the Auto Hide feature is disabled, the Toolbox window stays stationary and is always ready for use. 2. Click the mouse cursor somewhere outside of the Toolbox window and note that the Toolbox window stays fixed. 3. Click on the pushpin icon to make it horizontal. The Auto Hide feature is now enabled. 4. Move the mouse cursor somewhere outside of the Toolbox window and note that the window slides into a tab on the left side of the screen. The name and icon of the Toolbox window appear on the tab. 5. Hover the mouse cursor over the tab. The window slides into view and is ready for use. 6. Place a new control on the form, and then move the cursor away from the Toolbox window. The window automatically slides back to its tab on the edge of the screen. Note: We recommend that you keep the Toolbox, Solution Explorer, and Properties windows displayed unless you are creating a program with a very large form and need extra space.

■ Positioning and Aligning Controls Visual Basic provides several tools for positioning and aligning controls on a form. Proximity lines are short line segments that help you place controls a comfortable distance from each other and from the sides of the form. Snap lines are horizontal and vertical line segments that help you align controls. The Format menu is used to align controls, center controls horizontally and vertically in a form, and make a group of selected controls the same size.

◆

◆

A Positioning and Aligning Walkthrough 1. Begin a new program. 2. Place a button near the center of the form. 3. Drag the button toward the upper-right corner of the form until two short line segments appear. See Figure 2.14(a). The button is now a comfortable distance from the two sides of the form. 4. Place a second button below the first button and drag it upwards until a proximity line appears between the two buttons. The buttons are now a comfortable distance apart. 5. Resize and position the two buttons as shown in Figure 2.14(b). 6. Drag Button2 upwards until a blue line appears along the bottoms of the two buttons. See Figure 2.14(c). This blue line is called a snap line. The bottoms of the two buttons are now aligned. 7. Continue dragging Button2 upwards until a purple snap line appears just underneath the words Button1 and Button2. See Figure 2.14(d). The middles of the two buttons are now aligned. If we were to continue dragging Button2 upwards, a blue snap line would tell us when the tops are aligned. Step 10 shows another way to align the controls.

Proximity line

(b)

(a)

Snap line

(c)

(d) FIGURE 2.14

Positioning Controls.

8. Click on Button1 and then hold down the Ctrl key and click on Button2. After the mouse button is released, both buttons will be selected. Note: This process (called selection of multiple controls) can be repeated to select a group of any number of controls.

9. With the two buttons still selected, press F4 to open the Properties window. Then set the ForeColor property to Blue. Notice that the ForeColor property has been altered for both buttons at the same time. Actually, any property that is common to every control in selected multiple controls can be set simultaneously for the entire group. 10. With the two buttons still selected, open the Format menu in the Menu bar, hover over Align, and click on Tops. The tops of the two buttons are now aligned. Precisely, Button1 (the first button selected) will stay fixed, and Button2 will move up so that its top is aligned with the top of Button1. The most common uses of the submenus of the Format menu are as follows: Align: Align middles or corresponding sides of a group of selected controls. Make Same Size: Make the width and/or height of a group of selected controls the same. Horizontal Spacing: Equalize the horizontal spacing between a group of three or more selected controls arranged in a row. Vertical Spacing: Equalize the vertical spacing between a group of three or more selected controls arranged in a column. Center in Form: Center a selected control either horizontally or vertically in a form. When multiple controls are selected with the Ctrl key, the first control selected (called the primary control of the group) will have white sizing handles, while the other controls will have black sizing handles. All alignment and sizing statements initiated from the Format menu will keep the primary control fixed, and align (or size) the other controls with respect to the primary control. After multiple controls have been selected, they can be dragged as a group and deleted as a group. Exercises 35 and 36 show how the arrow keys can be used to move and size a control. The arrow keys also can be used to move and size multiple controls as a group. A group of controls also can be selected by clicking the mouse outside the controls, dragging it across the controls, and releasing it. The Select All command from the Edit menu (or the key combination Ctrl + A) causes all the controls on the form to be selected. Although these methods are easy to apply, they do not allow the programmer to choose the primary control.

■ Setting Tab Order Each time the Tab key is pressed while a program is running, the focus moves from one control to another. The following walkthrough explains how to determine the order in which the focus moves and how that order can be changed. 1. Start a new program. 2. Place a button, a text box, and a list box on a form. 3. Run the program, and successively press the Tab key. Notice that the controls receive the focus in the order they were placed on the form.

◆

4. End the program. 5. Click on Tab Order in the View menu. The screen appears as in Figure 2.15(a). The controls are numbered from 0 to 2 in the order they were created. Each of the numbers is referred to as a tab index. 6. Click on the list box, then the button, and finally the text box. Notice that the tab indexes change as shown in Figure 2.15(b).

(a)

(b) FIGURE 2.15

7. Click again on Tab Order in the View menu to set the new tab order. 8. Run the program again, and successively press Tab. Notice that the controls receive the focus according to the new tab order. 9. End the program. 10. Add a label to the form, rerun the program, and successively press Tab. Notice that the label does not receive the focus. Whether or not a control can receive the focus is determined by the setting of its TabStop property. By default, the setting for TabStop property is True for buttons, text boxes, and list boxes, and False for labels. In this book we always use these default settings. Note: Even though labels do not receive the focus while tabbing, they are still assigned a tab index.

■ Comments 1. While you are working on a program, the program resides in memory. Removing a program from memory is referred to as closing the program. A program is automatically closed when you begin a new program. Also, it can be closed directly with the Close Project command from the File menu. 2. Three useful properties that have not been discussed are the following: (a) BackColor: This property specifies the background color for the form or a control.

(b) Visible: Setting the Visible property to False causes an object to disappear when the program is run. The object can be made to reappear with code. (c) Enabled: Setting the Enabled property of a control to False restricts its use. It appears grayed and cannot receive the focus. Controls sometimes are disabled temporarily if they do not apply to the current state of the program. 3. Most properties can be set or altered with code as the program is running instead of being preset from the Properties window. For instance, a button can be made to disappear with a line such as Button1.Visible = False. The details are presented in Section 2.3. 4. If you inadvertently double-click on a form, a window containing text will appear. (The first line is Public Class Form1.) This is the Code Editor, which is discussed in the next section. Press Ctrl+Z to undo the addition of this new code. To return to the Form Designer, click on the tab above the Main area labeled “Form1.vb [Design].” 5. We have seen two ways to place a control onto a form. A third method is to drag the control from the Toolbox to the form. 6. The Recent Projects pane in the Start Page not only displays a list of recently updated programs, it also lets you open existing programs and create new programs. To open any existing program or to create a new program, click on the appropriate word “Project…” at the bottom of the pane. 7. The Getting Started pane in the Start Page displays a list of Help topics, Web sites, technical articles, and other sources that highlight features in the product. Practice Problems 2.2 1. What is the difference between the Text and the Name properties of a button? 2. The first two group names in the Toolbox are “All Windows Forms” and “Common Controls.” How many groups are there? EXERCISES 2.2

1. Create a form with two buttons, run the program, and click on each button. Do you notice anything different about a button after it has been clicked? 2. While a program is running, a control is said to lose focus when the focus moves from that control to another control. In what three ways can the user cause a control to lose focus? In Exercises 3 through 24, carry out the task. Use a new program for each exercise. 3. 4. 5. 6.

Place “CHECKING ACCOUNT” in the title bar of a form. Create a text box containing the words “PLAY IT, SAM” in blue letters. Create a text box with a yellow background. Create a text box named txtGreeting and containing the word “HELLO” in large italic letters.

7. Create a label containing the sentence “After all is said and done, more is said than done.” The sentence should occupy three lines, and each line should be centered horizontally in the label. 8. Create a read-only text box containing the words “Visual Basic” in bold white letters on a red background. 9. Create a text box named txtLanguage and containing the words “Visual Basic 2008” in Courier New font. 10. Create a yellow button named btnPush and containing the word “PUSH”. 11. Create a white button containing the word “PUSH” in large italic letters. 12. Create a button containing the word “PUSH” in bold letters in which the letter P is underlined. 13. Create a button containing the word “PUSH” with the letter H as the access key. 14. Create a label containing the word “ALIAS” in white on a blue background. 15. Create a label named lblAKA and containing the centered italicized word “ALIAS”. 16. Place BALANCE SHEET in the title bar of a form, where the form has a yellow background. 17. Create a label containing VISUAL on the first line and BASIC on the second line. Each word should be right justified. 18. Create a form named frmHello whose title bar reads “Hello World”. 19. Create a label containing a picture of a diskette. (Hint: Use the Wingdings character 6.) Make the diskette as large as possible. 20. Create a label containing the bold word “ALIAS” in the Courier New font. 21. Create a list box with a yellow background. 22. Create a list box that will be invisible when the program is run. 23. Create a form named frmYellow with a yellow background. 24. Create a button containing a picture of a red bell. (Hint: Use the Wingdings character %.) Make the bell as large as possible. In Exercises 25 through 30, create the interface shown in that figure. (These exercises give you practice creating controls and assigning properties. The interfaces do not necessarily correspond to actual programs.) 25.

26.

27.

28.

29.

30.

31. Create a replica of your bank check on a form. Words common to all checks, such as “PAY TO THE ORDER OF,” should be contained in labels. Items specific to your checks, such as your name at the top left, should be contained in text boxes. Make the check on the screen resemble your personal check as much as possible. 32. Create a replica of your campus ID on a form. Words that are on all student IDs, such as the name of the college, should be contained in labels. Information specific to your ID, such as your name and Social Security number, should be contained in text boxes. Simulate your picture with a text box containing a smiling face—a size 24 Wingdings J. 33. Consider the form shown in Exercise 25. Assume the Batman button was added to the form before the Robin button. What is the tab index of the Robin button? 34. Consider the form shown in Exercise 26. Assume the first control added to the form was the label. What is the tab index of the label? The following hands-on exercises develop additional techniques for manipulating and accessing controls placed on a form. 35. Place a text box on a form and select the text box. What is the effect of pressing the various arrow keys? 36. Place a text box on a form and select the text box. What is the effect of pressing the various arrow keys while holding down the Shift key? 37. Repeat Exercise 36 for selected multiple controls. 38. Repeat Exercise 35 for selected multiple controls. 39. Place a label and a list box on a form and change their font sizes to 12 at the same time. 40. Place a button in the center of a form and select it. Hold down the Ctrl key and press an arrow key. Repeat this process for each of the other arrow keys. Describe what happens.

41. Place a label and a text box on a form with the label to the left of and above the text box. Select the label. Hold down the Ctrl key and press the down-arrow key twice. With the Ctrl key still pressed, press the right-arrow key. Describe what happens. 42. Place two buttons on a form with one button to the right of and below the other button. Select the lower button, hold down the Ctrl key, and press the left-arrow key. With the Ctrl key still pressed, press the up-arrow key. Describe the effect of pressing the two arrow keys. 43. Experiment with the Align command on the Format menu to determine the difference between the center and the middle of a control. 44. Place four large buttons vertically on a form. Use the Format menu to make them the same size and to make the spacing between them uniform. 45. Place a text box on a form, select the text box, and open its Properties window. Double-click on the name (not the Settings box) of the ReadOnly property. Double-click again. What is the effect of double-clicking on a property whose possible settings are True and False? 46. Place a button on a form, select the button, and open its Properties window. Double-click on the name (not the Settings box) of the ForeColor property. Double-click repeatedly. Describe what is happening. Solutions to Practice Problems 2.2

1. The text is the words appearing on the button, whereas the name is the designation used to refer to the button. Initially, they have the same value, such as Button1. However, each can be changed independently of the other.

2. 11 groups. Figure 2.16 shows the Toolbox after each of the - signs preceding group names have been converted to + signs.

Until Chapter 9, all of the controls discussed in this book can be found in the Common Controls group. The group General, which holds text fragments rather than controls, will first be used in Section 3.3 to hold snippets of programming code.

FIGURE 2.16

Toolbox group names

2.3

Visual Basic Events

When a Visual Basic program runs, the form and its controls appear on the screen. Normally, nothing happens until the user takes an action, such as clicking a control or pressing a key. We call such an action an event. The programmer writes code that reacts to an event by performing some functionality. The three steps in creating a Visual Basic program are as follows: 1. Create the interface; that is, generate, position, and size the objects. 2. Set properties; that is, configure the appearance of the objects. 3. Write the code that executes when events occur. Section 2.2 covered Steps 1 and 2; this section is devoted to Step 3. Code consists of statements that carry out tasks. In this section, we limit ourselves to statements that change properties of a control or the form while a program is running. Properties of controls are changed in code with statements of the form controlName.property = setting

where controlName is the name of the control, property is one of the properties of the control, and setting is a valid setting for that property. Such statements are called assignment statements. They assign values to properties. Three examples of assignment statements are as follows: 1. The statement txtBox.Text = "Hello" displays the word Hello in the text box. 2. The statement btnButton.Visible = True

makes the button visible. 3. The statement txtBox.ForeColor = Color.Red

sets the color of the characters in the text box named txtBox to red. Most events are associated with controls. The event "click on btnButton" is different from the event "click on lstBox." These two events are specified btnButton.Click and lstBox.Click. The statements to be executed when an event occurs are written in a block of code called an event procedure. The first line of an event procedure (called the header) has the form Private Sub objectName_event(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles objectName.event

Since we do not make any use of the lengthy text inside the parentheses in this book, for the sake of readability we replace it with an ellipsis. However, it will automatically appear in our programs each time Visual Basic creates the header for an event procedure. The structure of an event procedure is

Private Sub objectName_event(...) Handles objectName.event statements End Sub

where the three dots (that is, the ellipsis) represent ByVal sender As System.Object, ByVal e As System.EventArgs

Words such as "Private," "ByVal," "As," "Sub," "Handles," and "End" have special meanings in Visual Basic and are referred to as keywords or reserved words. The Visual Basic editor automatically capitalizes the first letter of a keyword and displays the word in blue. The word "Sub" in the first line signals the beginning of the procedure, and the first line identifies the object and the event occurring to that object. The last line signals the termination of the event procedure. The statements to be executed appear between these two lines. (Note: The word "Private" indicates that the event procedure cannot be invoked by another form. This will not concern us until much later in the book. The expression following Handles identifies the object and the event happening to that object. The expression "objectName_event" is the default name of the procedure and can be changed if desired. In this book, we always use the default name. The word "Sub" is an abbreviation of Subroutine.) For instance, the event procedure Private Sub btnButton_Click(...) Handles btnButton.Click txtBox.ForeColor = Color.Red End Sub

changes the color of the words in the text box to red when the button is clicked.

■ An Event Procedure Walkthrough The form in Figure 2.17, which contains two text boxes and a button, will be used to demonstrate what event procedures are and how they are created. Three event procedures will be used to alter the appearance of a phrase appearing in a text box. The event procedures are named txtFirst_TextChanged, btnRed_Click, and txtFirst_Leave.

FIGURE 2.17

OBJECT

PROPERTY

SETTING

frmDemo txtFirst txtSecond btnRed

Text

Demonstration

Text

Change Color to Red

The interface for the event procedure walkthrough.

1. Create the interface in Figure 2.17 in the Form Designer. The Name properties of the form, text boxes, and button should be set as shown in the Object column. The Text property of the form should be set to Demonstration, and the Text property of the button should be set to Change Color to Red. No properties need be set for the text boxes.

2. Click the right mouse button anywhere on the Main area, and click on View Code. The Form Designer IDE is replaced by the Code Editor IDE (also known as the Code view or the Code window). See Figure 2.18. Code Editor tab

Class Name box

FIGURE 2.18

Form Designer tab

Method Name box

The Visual Basic IDE in Code Editor mode.

The tab labeled frmDemo.vb corresponds to the Code Editor. You press the tab labeled frmDemo.vb [Design], when you want to return to the Form Designer. Just below the tabs are two drop-down list boxes. The left box is called the Class Name box, and the right box is called the Method Name box. (When you hover the mouse pointer over one of these list boxes, its type appears in a tooltip.) We will place our program code between the two lines shown. Let's refer to this region as the program region. Figure 2.18 shows that the Code Editor IDE has a Toolbox, Solution Explorer, and Properties window that support Auto Hide. The Solution Explorer window for the Code Editor functions exactly like the one for the Form Designer. The Code Editor's Toolbox has just one group, General, that is used to store code fragments which can be copied into a program when needed. The Code Editor's Properties window will not be used in this textbook. 3. Click on the tab labeled "frmDemo.vb [Design]" to return to the Form Designer. (You also can invoke the Form Designer by clicking Designer in the View menu, or by right-clicking the Code Editor and clicking View Designer.) 4. Double-click on the first text box. The Code Editor reappears, but now the following two lines of code have been added to the program region and the cursor is located on the blank line between them.

Private Sub txtFirst_TextChanged(...) Handles txtFirst.TextChanged End Sub

The first line is the header for the event procedure named txtFirst_TextChanged. This procedure is triggered by the event txtFirst.TextChanged. That is, whenever there is a change in the text displayed in the text box txtFirst, the code between the two lines just shown will be executed. 5. Type the line txtFirst.ForeColor = Color.Blue

at the cursor location. When you type the first period, a list containing all the properties of text boxes appears. See Figure 2.19(a). (Each property is preceded by a little Properties window icon. The list also contains something called methods, which we will discuss later.) At this point, you can scroll up the list and doubleclick on ForeColor to automatically enter that property. See Figure 2.19(b). Or, you can keep typing. After you have typed "For," the list shortens to the single word ForeColor. At that point, you can press the Tab key to enter the word "ForeColor." This feature, known as Member Listing, is one of the features of Visual Basic that use a Microsoft context-sensitive help technology called IntelliSense. txtFirst.

txtFirst.

(a)

(b) FIGURE 2.19

IntelliSence at work.

6. Return to the Form Designer and double-click on the button. The Code window reappears, and the first and last lines of the event procedure btnRed_Click appear in the program region. Type the line that sets the ForeColor property of txtFirst to Red. The event procedure will now appear as follows: Private Sub btnRed_Click(...) Handles btnRed.Click txtFirst.ForeColor = Color.Red End Sub

7. Click on the down-arrow button to the right of the Class Name box and then click on txtFirst in the drop-down list.

8. Click on the down-arrow button to the right of the Method Name box and then click on Leave in the drop-down list box. (The event txtFirst.Leave is triggered when the focus is removed from the text box.) The first and last lines of the event procedure txtFirst_Leave will be displayed. In this procedure, type the line that sets the ForeColor property of txtFirst to Black. The Code Editor will now look as follows: Public Class frmDemo Private Sub txtFirst_Leave(...) Handles txtFirst.Leave txtFirst.ForeColor = Color.Black End Sub Private Sub txtFirst_TextChanged(...) Handles txtFirst.TextChanged txtFirst.ForeColor = Color.Blue End Sub Private Sub btnRed_Click(...) Handles btnRed.Click txtFirst.ForeColor = Color.Red End Sub End Class

9. Hover the cursor over the word "ForeColor." Visual Basic now displays information about the foreground color property. This illustrates another help feature of Visual Basic. 10. Now run the program by pressing F5. 11. Type something into the text box. In Figure 2.20, the blue word "Hello" has been typed. (Recall that a text box has the focus whenever it is ready to accept typing— that is, whenever it contains a blinking cursor.)

FIGURE 2.20

Text box containing input.

12. Click on the second text box. The contents of the first text box will become black. When the second text box was clicked, the first text box lost the focus; that is, the event Leave happened to txtFirst. Thus, the event procedure txtFirst_Leave was invoked, and the code inside the procedure was executed. 13. Click on the button. This invokes the event procedure btnRed_Click, which changes the color of the words in txtFirst to Red.

14. Click on the first text box, and type the word "Friend" after the word "Hello." As soon as typing begins, the text in the text box is changed and the TextChanged event is triggered. This event causes the color of the contents of the text box to become blue. 15. You can repeat Steps 11 through 14 as many times as you like. When you are finished, end the program by pressing Alt+F4, clicking the Stop Debugging icon on the Toolbar, or clicking the Close button (X) on the form.

■ Properties and Event Procedures of the Form You can assign properties to the form itself in code. However, a statement such as frmDemo.Text = "Demonstration" will not work. The form is referred to by the keyword Me. Therefore, the proper statement is Me.Text = "Demonstration"

To display a list of the events associated with frmDemo, select “(frmDemo Events)” from the Class Name box and then open the Method Name box.

■ The Header of an Event Procedure As mentioned earlier, in a header for an event procedure such as Private Sub btnOne_Click(...) Handles btnOne.Click

btnOne_Click is the name of the event procedure, and btnOne.Click identifies the event that triggers the procedure. The name can be changed at will. For instance, the header can be changed to Private Sub ButtonPushed(...) Handles btnOne.Click

Also, an event procedure can be triggered by more than one event. For instance, if the previous line is changed to Private Sub ButtonPushed(...) Handles btnOne.Click, btnTwo.Click

the event will be triggered if either btnOne or btnTwo is clicked. We have been using ellipses 1 Á 2 as place holders for the phrase ByVal sender As System.Object, ByVal e As System.EventArgs

In Chapter 5, we will gain a better understanding of this type of phrase. Essentially, the word “sender” carries a reference to the object that triggered the event, and the letter “e” carries some additional information that the sending object wants to communicate. We will not make use of either “sender” or “e”.

■ Context-Sensitive Help Consider the program created in the event procedure walkthrough. If you click on the word txtFirst and then press the F1 key, a discussion of the Textbox control will appear. This feature of Visual Basic is called context-sensitive help. You can click on any element of a program and press F1 to obtain information relevant to that element. For instance, if you click on Sub and press F1, you will obtain information about procedures.

■ Text Files Programs often make use of data stored in files. In Chapter 10 we work with sophisticated data files called databases. Until then we will rely on simple data files known as text files. Text files can be created, viewed, and modified with sophisticated word processors such as Word, or with elementary word processors such as the Windows accessories WordPad and Notepad. They differ from files normally created with Word in that they have no formatting. They are pure text and nothing else – hence the name text file. The Visual Basic IDE provides simple ways to create and manage text files. The following walkthrough creates a text file: 1. Click on the New Project icon to start a new program. 2. Give a name, such as TextFileProg, to the program, and click on OK. 3. If the Solution Explorer window is not visible, click on Solution Explorer in the View menu. 4. Highlight the name of the program at the top of the Solution Explorer window. 5. Press Alt/P/W to invoke Add New Item from the Project menu. Alternately, click on the Add New Item icon in the Toolbar. (An Add New Item dialog box will appear.) 6. Select the Text File icon in the dialog box, and enter the filename PAYROLL.TXT in the Name text box. (Had we omitted the extension, Visual Basic would have automatically added the extension “.txt” to the name.) 7. Click the Add button. (The Text Editor, an elementary word processor, will appear in the Main area, and the filename will appear in the Solution Explorer window.) 8. Type the following information into the Text Editor. (It gives the names of employees, their hourly wages, and the number of hours worked in the past week. This text file will be used in Section 3.3.) Mike Jones 9.35 35 John Smith 10.75 33

9. Right-click on the tab containing the filename PAYROLL.TXT and click on “Save PAYROLL.TXT.” 10. Right-click on the tab containing the filename PAYROLL.TXT and click on Close.

■ Viewing All Possible Event Procedures for a Control All the event procedures associated with the selected control can be displayed and described in the Form Designer Properties window by clicking on the Events button (pictured as a lightning bolt) on the toolbar at the top of the Properties window. Figure 2.21 shows some of the 58 events for the button from the preceding event procedure walkthrough. A description of the selected event appears in the Description pane. Don’t be alarmed by the large number of events. There is only one event that is used extensively throughout this book – the Click event for buttons. Note: After viewing events in the Properties window, click on the Properties button (to the left of the Events button) to return to displaying properties in the window.

FIGURE 2.21

Events displayed in the Properties window.

■ Opening a Program Beginning with the next chapter, each example contains a program. These programs can be downloaded from my Prentice-Hall website for this book. See the discussion on page xv for details. The process of placing a program stored on a disk into the Visual Basic environment is referred to as opening the program. Let’s open the downloaded program 7-1-3 from Chapter 7. That program allows you to enter the name of a football team, and then displays the numbers of Super Bowls the team won. 1. From Visual Basic, click on Open Project in the File menu. (An Open Project dialog box will appear.) 2. Display the contents of the Ch07 subfolder downloaded from the website. 3. Double-click on 7-1-3. 4. Double-click on 7-1-3.sln. 5. If the Solution Explorer window is not visible, click on Solution Explorer in the View menu.

6. In the Solution Explorer window, click on frmBowl.vb. (Five buttons will appear at the top of the Solution Explorer window. See Figure 2.22.) At any time

you can click on the View Code button to invoke the Code Editor, or you can click on the View Designer button to invoke the Form Designer. Show All Files

FIGURE 2.22

Refresh

View Code

View Designer

Solution Explorer window.

7. Press F5 to run the program. 8. Type in the name of a football team (such as 49ers or Giants), and press the “Display Games Team Won” button. (See Figure 2.23.) You can repeat this process as many times as desired.

FIGURE 2.23

Output for program 7-1-3.

9. To end the program, click the Close button (X). The program just executed uses a text file named SBWINNERS.TXT. To view the text file, open the folder bin, open the subfolder Debug, and click on SBWINNERS.TXT.

(If the bin folder is not visible, click on the Show All Files button. If SBWINNERS.TXT is not listed in the Debug subfolder, click the Refresh button and reopen the folders. After reading Chapter 3, you will understand why text files are placed in the Debug subfolder of the bin folder.) The first line of the file gives the winner of the first Super Bowl, the second line gives the winner of the second Super Bowl, and so on. To close the text file, right-click on the tab labeled SBWINNERS.TXT, and click on Close.

■ Comments 1. The Visual Basic editor automatically indents the statements inside procedures. In this book, we indent by two spaces. To instruct your editor to indent by two spaces, select Options from the Tools menu, and uncheck the “Show all settings” box in the Options window that appears. Expand “Text Editor Basic” or “Text Editor,” click on “Editor,” enter 2 into the “Indent size:” box, and click on OK. 2. The event control.Leave is triggered when the specified control loses the focus. Its counterpart is the event control. Enter which is triggered when the specified control gets the focus. A related statement is control.Focus()

which moves the focus to the specified control. 3. We have ended our programs by clicking the Stop Debugging icon or pressing Alt + F4. A more elegant technique is to create a button, call it btnQuit, with caption Quit and the following event procedure: Private Sub btnQuit_Click(...) Handles btnQuit.Click End End Sub

4. For statements of the form object.Text = setting

the expression for setting must be surrounded by quotes. (For instance, lblName. Text = “Name”.) For properties where the proper setting is one of the words True or False, these words should not be surrounded by quotation marks. 5. Names of existing event procedures associated with an object are not automatically changed when you rename the object. You must change them yourself. However, the event that triggers the procedure (and all other references to the control) will change automatically. For example, suppose an event procedure is Private Sub btnOne_Click(...) Handles btnOne.Click btnOne.Text = "Press Me" End Sub

and, in the Form Designer, you change the name of btnOne to btnTwo. Then, when you return to the Code Editor the procedure will be Private Sub btnOne_Click(...) Handles btnTwo.Click btnTwo.Text = "Press Me" End Sub

6. The Code Editor has many features of a word processor. For instance, the operations cut, copy, paste, undo, and redo can be carried out with the sixth through ninth icons from the Toolbar. These operations, and several others, also can be initiated from the Edit menu. 7. The Code Editor can detect certain types of errors. For instance, if you type txtFirst.Text = hello

and then move away from the line, the automatic syntax checker will underline the word “hello” with a blue squiggle to indicate that something is wrong. When the mouse cursor is hovered over the offending wording, the editor will display a message explaining what is wrong. If you try to run the program without correcting the error, the dialog box in Figure 2.24 will appear.

FIGURE 2.24

Error dialog box.

8. When you double-click on a control in the Form Designer, the header for the most used event procedure is placed in the Code Editor. The event that appears most frequently in this book is the Click event for button controls. 9. Font properties, such as the name, style, and size, are usually specified at design time. The setting of the properties can be displayed in code with statements such as 1stBox.Items.Add(txtBox.Font.Name) 1stBox.Items.Add(txtBox.Font.Bold) 1stBox.Items.Add(txtBox.Font.Size)

However, a font’s name, style, and size properties cannot be altered in code with statements of the form txtBox.Font.Name = "Courier New" txtBox.Font.Bold = True txtBox.Font.Size = 16

10. When you make changes to a program, asterisks appear as superscripts on the tabs labeled “frmName.vb [design]” and “frmName.vb.” The asterisks disappear when the program is saved or run. Note: If the program has been saved to disk, all files for the program will be automatically updated on the disk whenever the program is saved or run.

Practice Problem 2.3 1. What event procedure is displayed when you double-click on each of the following controls in the Form Designer? (a) text box (b) button (c) label (d) list box 2. Give a statement that will prevent the user from typing into txtBox. EXERCISES 2.3

In Exercises 1 through 6, describe the contents of the text box after the button is clicked. 1. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = "Hello" End Sub

2. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.ForeColor = Color.Red txtBox.Text = "Hello" End Sub

3. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.BackColor = Color.Orange txtBox.Text = "Hello" End Sub

4. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = "Goodbye" txtBox.Text = "Hello" End Sub

5. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = "Hello" txtBox.Visible = False End Sub

6. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.BackColor = Color.Yellow txtBox.Text = "Hello" End Sub

In Exercises 7 through 10, assume that the three objects on the form were created in the order txtFirst, txtSecond, and lblOne. Determine the output displayed in lblOne when the program is run and the Tab key is pressed. Note: Initially, txtFirst has the focus. 7. Private Sub txtFirst_Leave(...) Handles txtFirst.Leave lblOne.ForeColor = Color.Green lblOne.Text = "Hello" End Sub

8. Private Sub txtFirst_Leave(...) Handles txtFirst.Leave lblOne.BackColor = Color.White lblOne.Text = "Hello" End Sub

9. Private Sub txtSecond_Enter(...) Handles txtSecond.Enter lblOne.BackColor = Color.Gold lblOne.Text = "Hello" End Sub

10. Private Sub txtSecond_Enter(...) Handles txtSecond.Enter lblOne.Visible = False lblOne.Text = "Hello" End Sub

In Exercises 11 through 16, determine the errors. 11. Private Sub btnOutput_Click(...) Handles btnOutput.Click Form1.Text = "Hello" End Sub

12. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = Hello End Sub

13. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtFirst.ForeColor = Red End Sub

14. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox = "Hello" End Sub

15. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Font.Size = 20 End Sub

16. Private Sub btnOutput_Click(...) Handles btn1.Click, btn2.Click Me.Color = Color.Yellow End Sub

In Exercises 17 through 28, write a line (or lines) of code to carry out the task. 17. 18. 19. 20.

Display “E.T. phone home.” in lblTwo. Display “Play it, Sam.” in lblTwo. Display “The stuff that dreams are made of.” in red letters in txtBox. Display “Life is like a box of chocolates.” in txtBox with blue letters on a gold background. 21. Disable txtBox. 22. Change the words in the form’s title bar to “Hello World.” 23. Make lblTwo disappear.

24. 25. 26. 27. 28. 29. 30. 31.

Change the color of the letters in lblName to red. Enable the disabled button btnOutcome. Give the focus to btnCompute. Give the focus to txtBoxTwo. Change the background color of the form to White. Describe the Enter event in your own words. Describe the Leave event in your own words. The label control has an event called DoubleClick that responds to a doubleclicking of the left mouse button. Write a simple program to test this event. Determine whether you can trigger the DoubleClick event without also triggering the Click event. 32. Write a simple program to demonstrate that a button’s Click event is triggered when you press the Enter key while the button has the focus.

In Exercises 33 through 38, the interface and initial properties are specified. Write the program to carry out the stated task. 33. When one of the three buttons is pressed, the words on the button are displayed in the text box with the stated alignment. Note: Rely on IntelliSense to provide you with the proper settings for the TextAlign property. OBJECT

PROPERTY

SETTING

frmAlign txtBox btnLeft btnCenter btnRight

Text ReadOnly Text Text Text

Text Alignment True Left Justify Center Right Justify

34. When one of the buttons is pressed, the face changes to a smiling face (Wingdings character “J”) or a frowning face (Wingdings character “L”). OBJECT

PROPERTY

SETTING

frmFace lblFace

Text Font Name Font Size Text Text Text

Face Wingdings 24 K Smile Frown

btnSmile btnFrown

35. Pressing the buttons alters the background and foreground colors in the text box. OBJECT

PROPERTY

SETTING

frmColors lblBack btnRed btnBlue txtBox

Text Text Text Text Text TextAlign Text Text Text

Colorful Text Background Red Blue Beautiful Day Center Foreground White Yellow

lblFore btnWhite btnYellow

36. When one of the three text boxes receives the focus, its text becomes red. When it loses the focus, the text returns to black. The buttons set the alignment in the text boxes to Left or Right. Note: Rely on IntelliSense to provide you with the proper settings for the TextAlign property. OBJECT

PROPERTY

SETTING

frm123 txtOne txtTwo txtThree btnLeft btnRight

Text Text Text Text Text Text

One, Two, Three One Two Three Left Right

37. When the user moves the focus to one of the three small text boxes at the bottom of the form, an appropriate saying is displayed in the large text box. Use the sayings “I like life, it’s something to do.”; “The future isn’t what it used to be.”; and “Tell the truth and run.” OBJECT

PROPERTY

SETTING

frmQuote txtQuote txtLife txtFuture txtTruth

Text ReadOnly Text Text Text

Sayings True Life Future Truth

2.3

Visual Basic Events

38. The user can disable or enable the text box by clicking on the appropriate button. After the user clicks the Enable button, the text box should receive the focus. OBJECT

PROPERTY

SETTING

frmTextBox txtBox btnDisable btnEnable

Text

Text Box

Text Text

Disable Text Box Enable Text Box

In Exercises 39 through 44, write a program with a Windows-style interface to carry out the task. 39. The form contains four square buttons arranged in a rectangular array. Each button has the caption “Push Me.” When the user clicks on a button, the button disappears and the other three become or remain visible. 40. A form contains two text boxes and one large label between them with no preset caption. When the first text box receives the focus, the label reads “Enter your full name.” When the second text box receives the focus, the label reads “Enter your phone number, including area code.” 41. Use the same form and properties as in Exercise 34, with the captions for the buttons replaced with Vanish and Reappear. Clicking a button should produce the stated result. 42. Simulate a traffic light with three small square text boxes placed vertically on a form. Initially, the bottom text box is solid green and the other text boxes are dark gray. When the Tab key is pressed, the middle text box turns yellow and the bottom text box turns dark gray. The next time Tab is pressed, the top text box turns red and the middle text box turns dark gray. Subsequent pressing of the Tab key cycles through the three colors. Hint: First place the bottom text box on the form, then the middle text box, and finally the top text box. 43. The form contains a single read-only text box and two buttons. When the user clicks on one of the buttons, the sentence “You just clicked on a button.” is displayed in the text box. The program should consist of a single event procedure. 44. The form contains two text boxes into which the user types information. When the user clicks on one of the text boxes, it becomes blank and its contents are displayed in the other text box. Note: A text box can be cleared with the statement txtBox.Clear() or the statement txtBox.Text = "".

◆

Solutions to Practice Problem 2.3 1. (a) (b) (c) (d)

TextChanged Click Click SelectedIndexChanged

2. Three possibilities are txtBox.Enabled = False txtBox.ReadOnly = True txtBox.Visible = False

CHAPTER 2 SUMMARY 1. The Visual Basic Form Designer displays a form that can hold a collection of controls for which various properties can be set. Some examples of controls are text boxes, labels, buttons, and list boxes. Some useful properties are Text (sets the text displayed in a control), Name (used to give a meaningful name to a control), Font.Name (selects the name of the font used), Font.Size (sets the size of the text displayed), Font.Bold (displays boldface text), Font.Italic (displays italics text), BackColor (sets the background color), ForeColor (sets the color of the text), ReadOnly (determines whether text can be typed into a text box when the program is running), TextAlign (sets the type of alignment for the text in a control), Enabled (determines whether a control can respond to user interaction), and Visible (determines whether an object can be seen or is hidden). 2. An event procedure is executed when something happens to a specified object. Some events are object.Click (object is clicked), object.TextChanged (a change occurred in the value of the object’s Text property), object.Leave (object loses the focus), and object.Enter (object receives the focus). Note: The statement object.Focus() moves the focus to the specified object. 3. Text files, which can be created and managed in the Visual Basic IDE, provide a convenient device for holding data to be accessed by a program. 4. Programming assistance can be accessed through the Help menu, or via IntelliSense and context-sensitive help. 5. Tab order, the order in which the user moves the focus from one control to another by pressing the Tab key while the program is running, can be set from the View menu.

3 Variables, Input, and Output

3.1

Numbers

Much of the data processed by computers consists of numbers. In computerese, numbers are called numeric literals. This section discusses the operations that are performed with numbers and the ways numbers are displayed.

■ Arithmetic Operations The five standard arithmetic operations in Visual Basic are addition, subtraction, multiplication, division, and exponentiation. Addition, subtraction, and division are denoted in Visual Basic by the standard symbols +, - , and /, respectively. However, the notations for multiplication and exponentiation differ from the customary mathematical notations as follows: Mathematical Notation a # b or a * b ar

Visual Basic Notation a*b a–r

(The asterisk [*] is the upper character of the 8 key. The caret [¿] is the upper character of the 6 key.) One way to show a number on the screen is to display it in a list box. If n is a number, then the instruction lstBox.Items.Add(n)

displays the number n as the last item in the list box. Add is called a method. (Generally, a method is a process that performs a task for a particular object.) If the parentheses contain a combination of numbers and arithmetic operations, the Add method carries out the operations and displays the result. Another important method is Clear. The statement lstBox.Items.Clear()

erases all the items displayed in the list box lstBox. Example 1 The following program applies each of the five arithmetic operations. Preceding the program is the form design and a table showing the names of the objects on the form and the settings, if any, for properties of these objects. This form design is also used in the discussion and examples in the remainder of this section. The word “Run” in the phrasing [Run Á] indicates that F5 should be pressed to execute the program. Notice that in the output 3 / 2 is displayed in decimal form. Visual Basic never displays numbers as common fractions. In the evaluation of 2 * 13 + 42, the operation inside the parentheses is calculated first. Note: All programs appearing in examples and case studies are provided on the companion website for this book. See the discussion on page xv for details.

3.1

Numbers

OBJECT

PROPERTY

SETTING

frmArithmetic lstResults btnCompute

Text

3-1-1

Text

Compute

Private Sub btnCompute_Click(...) Handles btnCompute.Click lstResults.Items.Clear() lstResults.Items.Add(3 + 2) lstResults.Items.Add(3 - 2) lstResults.Items.Add(3 * 2) lstResults.Items.Add(3 / 2) lstResults.Items.Add(3 ^ 2) lstResults.Items.Add(2 * (3 + 4)) End Sub

[Run, and then click the button.]

■ Variables In applied mathematics problems, quantities are referred to by names. For instance, consider the following high school algebra problem: “If a car travels at 50 miles per hour, how far will it travel in 14 hours? Also, how many hours are required to travel 410 miles?” The solution to this problem uses the well-known formula distance = speed * time elapsed Here’s how this problem would be solved with a computer program: Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim speed As Double

Chapter 3 Variables, Input, and Output Dim timeElapsed As Double Dim distance As Double lstResults.Items.Clear() speed = 50 timeElapsed = 14 distance = speed * timeElapsed lstResults.Items.Add(distance) distance = 410 timeElapsed = distance / speed lstResults.Items.Add(timeElapsed) End Sub

[Run, and then click the button. The following is displayed in the list box.] 700 8.2

Skip the second, third, and fourth lines of the event procedure for now. We will return to them soon. The sixth line sets the speed to 50, and the seventh line sets the time elapsed to 14. The eighth line multiplies the value for the speed by the value for the time elapsed and sets the distance to this product. The next line displays the answer to the distance-traveled question. The three lines before the End Sub statement answer the time-required question in a similar manner. The names speed, timeElapsed, and distance, which hold values, are referred to as variables. Consider the variable timeElapsed. In the seventh line, its value was set to 14. In the eleventh line, its value was changed as the result of a computation. On the other hand, the variable speed had the same value, 50, throughout the program. In general, a variable is a name that is used to refer to an item of data. The value assigned to the variable may change during the execution of the program. In Visual Basic, variable names can be up to 16,383 characters long, must begin with a letter or an underscore, and can consist only of letters, digits, and underscores. (The shortest variable names consist of a single letter.) Visual Basic does not distinguish between uppercase and lowercase letters used in variable names. Some examples of variable names are total, numberOfCars, taxRate_2008, and n. As a convention, we write variable names in lowercase letters except for the first letters of additional words (as in gradeOnFirstExam). This convention is called camel casing. If var is a variable and n is a literal, then the statement var = n

assigns the number n to the variable var. (Such a statement is another example of an assignment statement.) A variable is declared to be of a certain type depending on the sort of data that can be assigned to it. The most versatile type for holding numbers is called Double. A variable of type Double can hold whole, fractional, or mixed numbers between about -1.8 # 10308 and 1.8 # 10308. Dim statements (also called declaration statements) declare the names and types of the variables to be used in the program. The second, third, and fourth lines of this event procedure declare three variables of type Double and give them the names speed, timeElapsed, and distance.

In general, a statement of the form Dim varName As Double

declares a variable named varName to be of type Double. Actually, the Dim statement causes the computer to set aside a location in memory with the name varName. Since varName is a numeric variable, the Dim statement initially places the number zero in that memory location. (We say that zero is the initial value or default value of the variable.) Each subsequent assignment statement having varName to the left of the equal sign will change the value of the number. The initial value can be set to a value other than zero. To specify a nonzero initial value, follow the declaration statement with an equal sign followed by the initial value. The statement Dim varName As Double = 50

declares the specified variable as a variable of type Double and gives it the initial value 50. The statement lstBox.Items.Add(varName)

looks into this memory location for the current value of the variable and displays that value in the list box. A combination of literals, variables, and arithmetic operations that can be evaluated to yield a number is called a numeric expression. Expressions are evaluated by replacing each variable by its value and carrying out the arithmetic. Some examples of expressions are 2 * distance + 7, n + 1, and 1a + b2>3. Example 2 The following program displays the default value of a variable and the value of an expression: Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim a As Double Dim b As Double = 3 lstResults.Items.Clear() lstResults.Items.Add(a) lstResults.Items.Add(b) a = 5 lstResults.Items.Add(a * (2 + b)) End Sub

[Run, and then click the button. The following is displayed in the list box.] 0 3 25

If var is a variable, then the assignment statement var = expression

first evaluates the expression on the right and then assigns its value to the variable on the left. For instance, the event procedure in Example 2 can be written as Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim a As Double Dim b As Double Dim c As Double lstResults.Items.Clear() a = 5 b = 3 c = a * (2 + b) lstResults.Items.Add(c) End Sub

The expression a * 12 + b2 is evaluated to 25, and then this value is assigned to the variable c.

■ Incrementing the Value of a Variable Because the expression on the right side of an assignment statement is evaluated before an assignment is made, a statement such as var = var + 1

is meaningful. It first evaluates the expression on the right (that is, it adds 1 to the original value of the variable var) and then assigns this sum to the variable var. The effect is to increase the value of the variable var by 1. In terms of memory locations, the statement retrieves the value of var from var's memory location, uses it to compute var + 1, and then places the sum back into var's memory location. This type of calculation is so common that Visual Basic provides a special operator to carry it out. The statement var = var + 1 can be replaced with the statement var += 1

In general, if n has a numeric value, then the statement var += n

adds n to the value of var.

■ Built-In Functions: Math.Sqrt, Int, Math.Round There are several common operations that we often perform on numbers other than the standard arithmetic operations. For instance, we may take the square root of a number or round a number. These operations are performed by built-in functions. Functions associate with one or more values called the input, and a single value called the output. The function is said to return the output value. The three functions considered here have numeric input and output. The function Math.Sqrt calculates the square root of a number. The function Int finds the greatest integer less than or equal to a number. Therefore, Int discards the decimal part of positive numbers. The value of Math.Round(n, r) is the number n rounded to r decimal places. The parameter r can be omitted. If so, n is rounded to a whole number. Some examples follow:

Math.Sqrt192 is 3. Math.Sqrt102 is 0. Math.Sqrt122 is 1.414214.

Int12.72 is 2. Int132 is 3. Int1 -2.72 is - 3.

Math.Round12.72 is 3. Math.Round12.317, 22 is 2.32. Math.Round12.317, 12 is 2.3.

The terms inside the parentheses can be numbers (as shown), numeric variables, or numeric expressions. Expressions are first evaluated to produce the input. Example 3 The following program evaluates each of the functions for a specific input given by the value of the variable n: Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim n As Double Dim root As Double n = 6.76 root = Math.Sqrt(n) lstResults.Items.Clear() lstResults.Items.Add(root) lstResults.Items.Add(Int(n)) lstResults.Items.Add(Math.Round(n, 1)) End Sub

[Run, and then click the Compute button. The following is displayed in the list box.] 2.6 6 6.8

Example 4 an expression:

The following programevaluates each of the preceding functions at

```
Private Sub btnCompute_Click(...) Handles btnCompute.Click Dima As Double Dimb As Double a = 2 b = 3 lstResults.Items.Clear() lstResults.Items.Add(Math.Sqrt(5 * b + 1)) lstResults.Items.Add(Int(a ^ b + 0.8)) lstResults.Items.Add(Math.Round(a / b, 3)) End Sub
```

[Run, and then click the button. The following is displayed in the list box.] 4 8 0.667

◆

67

68

◆

Chapter 3 Variables, Input, and Output

■ The Integer Data Type In this text, we sometimes need to use variables of type Integer. An Integer variable is declared with a statement of the formDim varName As Integer

and can be assigned only whole numbers fromabout - 2 billion to 2 billion. Integer variables are used primarily for counting.

■ Multiple Declarations Several variables of the same type can be declared with a single Dimstatement. For instance, the two Dimstatements in Example 2 can be replaced by the single statement Dima, b As Double

Two other types of multiple-declaration statement are Dima As Double, b As Integer Dimc As Double = 2, b As Integer = 5

■ Parentheses Parentheses cannot be used to indicate multiplication, as is commonly done in algebra. For instance, the expression x1y + z2 is not valid. It must be written as x * 1y + z2. Parentheses should be used when necessary to clarify the meaning of an expression. When there are no parentheses, the arithmetic operations are performed in the following order: (1) exponentiations; (2) multiplications and divisions; (3) additions and subtractions. In the event of ties, the leftmost operation is carried out first. Table 3.1 summarizes these rules. Note: If you use parentheses liberally, you will not have to rely on the precedence table for arithmetic operations. For instance, write (2 * 3) + 4 instead of 2 * 3 + 4. Write (2 ^ 3) + 4 instead of 2 ^ 3 + 4.

TABLE 3.1

Level of precedence for arithmetic operations. () ¿

* /

+-

Inner to outer, left to right Left to right in expression Left to right in expression Left to right in expression

■ Three Types of Errors Grammatical errors, such as misspellings, omissions, or incorrect punctuations, are called syntax errors. Most syntax errors are spotted by the Code Editor when they are entered. The editor underlines the syntax error with a blue squiggly line and displays a description of the error when the mouse cursor is hovered over the squiggly line. Some incorrect statements and their errors are as follows: Statement lstBox.Itms.Add(3) lstBox.Items.Add12+2 Dimm; n As Integer

Reason for Error The word Items is misspelled. The number following the plus sign is missing. The semicolon should be a comma.

3.1

Numbers

Errors that occur while a programis running are called run-time errors or exceptions. They usually result fromthe inability of the computer to carry out the intended task. For instance, if the file DATA.TXT is not in the root folder of the C drive, then a statement that refers to the file by the filespec “C:\DATA.TXT” will cause the programto stop executing and produce a message box with the title FileNotFoundException was unhandled.

Also, a yellow arrow will appear at the left side of the line of code that caused the error. At that point, you should end the program. A third type of error is the so-called logical error. Such an error occurs when a programdoes not performthe way it was intended. For instance, the line average = firstNum + secondNum / 2

is syntactically correct. However, the missing parentheses in the line of code are responsible for an incorrect value being generated. Appendix D discusses debugging tools that can be used to detect and correct logical errors.

■ The Error List Window Syntax errors are not only indicated in the Code Editor, but also are listed in the Error List window. (Note: If the Error List window is not visible in the IDE, click on Error List in the View menu.) Example 5

The following programcontains three errors.

```
Private Sub btnCompute_Click(...) Handles btnCompute.Click Dimm; n As Double lstResults.Items.Add(5 lstResults.Items.Add(a) End Sub
```

[Press Alt/V/I to display the Error List window.]

The Private Sub statement is in line 3. (Line 1 contains the Public Class statement and line 2 is a blank line.) Click on one of the three errors in the Error List window and then press F1 to see a discussion of that error.

◆

■ Comments 1. Declaring variables at the beginning of each event procedure is regarded as good programming practice because it makes programs easier to read and helps prevent certain types of errors. 2. Keywords (reserved words) cannot be used as names of variables. For instance, the statements Dimprivate as Double and Dimsub As Double are not valid. To obtain a complete list of Visual Basic keywords, look up reserved keywords in Help's Index. 3. Names given to variables are sometimes referred to as identifiers. 4. In math courses, literals are referred to as constants. However, the word "constant" has a special meaning in programming languages. 5. Numeric literals used in expressions or assigned to variables must not contain commas, dollar signs, or percent signs. Also, mixed numbers, such as 8 1/2, are not allowed. 6. Although requesting the square root of a negative number does not terminate the execution of the program, it can produce unexpected results. For instance, the statement lstBox.Items.Add(Math.Sqrt(-1))

displays NaN. Note: NaN is an abbreviation for "Not a Number." 7. If the value of numVar is 0 and numVar has type Double, then the statements numVarInv = 1 / numVar lstBox.Items.Add(numVarInv) lstBox.Items.Add(1 / numVarInv)

cause the following items to be displayed in the list box: Infinity 0

8. When n is halfway between two successive whole numbers (such as 1.5, 2.5, 3.5, and 4.5), then it rounds to the nearest even number. For instance, Math.Round (2.5) is 2 and Math.Round(3.5) is 4. 9. In addition to the five arithmetic operators discussed at the beginning of this section, the Mod operator and the integer division operator (\) are two other useful operators. Let mand n be positive whole numbers. When you use long division to divide mby n, you obtain an integer result and an integer remainder. The result is m\ n and the remainder is mMod n. For instance, 14 \ 3 is 4 and 14 Mod 3 is 2 19 \ 5 is 3 and 19 Mod 5 is 4 10 \ 2 is 5 and 10 Mod 2 is 0. 10. In scientific notation, numbers are written in the formb # 10r, where b is a number of magnitude from 1 up to (but not including) 10, and r is an integer. Visual Basic displays very large numbers in scientific notation where b # 10r is written as bEr. (The letter E is an abbreviation for exponent.) For instance, when the statement lstBox.Items.Add(123 * 10 ^ 15) is executed, 1.23E+17 is displayed in the list box.

11. When you first enter a statement such as Dimn As Double, a squiggle will appear under the variable name and the Error List window will record a warning. The squiggle merely indicates that the variable has not yet been assigned a value. If the squiggle is still present after the entire event procedure has been entered, this will tell you that the variable was never used and that the declaration statement should be removed. Practice Problems 3.1 1. Evaluate 2 + 3 * 4. 2. Explain the difference between the assignment statement var1 = var2

and the assignment statement var2 = var1

3. Complete the table by filling in the value of each variable after each line is executed. a Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dima, b, c As Double 0 a = 3 3 b = 4 3 c = a + b a = c * a lstResults.Items.Add(a - b) b = b * b End Sub

b

c

0 0 4

0 0 0

4. Write a statement that increases the value of the numeric variable var by 5%. EXERCISES 3.1

In Exercises 1 through 6, evaluate the numeric expression without the computer, and then use Visual Basic to check your answer. 1. 3 * 4 3. 1>12¿32 5. 15 - 32 * 4

2. 7¿2 4. 3 + 14 * 52 6. 3 * 11 - 22¿52

In Exercises 7 through 10, evaluate the expression. 7. 7\3 9. 7 Mod 3

8. 14 Mod 4 10. 14\4

In Exercises 11 through 16, determine whether the name is a valid variable name. 11. sales.2008 13. fOrM_1040 15. expenses?

12. room&Board 14. 1040B 16. INCOME 2008

In Exercises 17 through 22, evaluate the numeric expression where a = 2, b = 3, and c = 4. 17. 1a * b2 + c 19. 11 + b2 * c 21. b¿1c - a2

18. a * 1b + c2 20. a¿c 22. 1c - a2¿b

In Exercises 23 through 28, write an event procedure to calculate and display the value of the expression. 23. 7 # 8 + 5 25. 5.5% of 20 27. 1713 + 1622

24. 11 + 2 # 923 26. 15 - 312 + 342 28. 4 1/2 - 3 5/8

In Exercises 29 and 30, complete the table by filling in the value of each variable after each line is executed. 29. x

y

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim x, y As Double x = 2 y = 3 * x x = y + 5 lstResults.Items.Clear() lstResults.Items.Add(x + 4) y = y + 1 End Sub

30. bal

inter

withDr

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim bal, inter, withDr As Double bal = 100 inter = 0.05 withDr = 25 bal += inter * bal bal = bal - withDr End Sub

In Exercises 31 through 38, determine the output displayed in the list box by the lines of code. 31. Dim amount As Double amount = 10 lstOutput.Items.Add(amount - 4)

32. Dim a, b As Integer a = 4

3.1

b = 5 * a lstOutput.Items.Add(a + b)

33. Dim n As Integer = 7 n += 1 lstOutput.Items.Add(1) lstOutput.Items.Add(n) lstOutput.Items.Add(n + 1)

34. Dim num As Integer = 5 num = 2 * num lstOutput.Items.Add(num)

35. Dim a, b As Integer lstOutput.Items.Add(a + 1) a = 4 b = a * a lstOutput.Items.Add(a * b)

36. Dim tax As Double tax = 200 tax = 25 + tax lstOutput.Items.Add(tax)

37. Dim x As Double = 3 x += 2 lstOutput.Items.Add(x * x) lstOutput.Items.Add(x + 3 * x)

38. Dim n As Double = 2, m As Double = 5 lstOutput.Items.Add(3 * n) n += n lstOutput.Items.Add(n + m) lstOutput.Items.Add(n - m)

In Exercises 39 through 44, identify the errors. 39. Dim a, b, c As Double a = 2 b = 3 a + b = c lstOutput.Items.Add(c)

40. Dim a, b, c, d As Double a = 2 b = 3 c = d = 4 lstOutput.Items.Add(5((a + b) / (c + d)

41. Dim balance, deposit As Double balance = 1,234 deposit = $100 lstOutput.Items.Add(balance + deposit)

42. Dim interest, balance As Double 0.05 = interest balance = 800 lstOutput.Items.Add(interest * balance)

43. Dim 9W As Double 9W = 2 * 9W lstOutput.Items.Add(9W)

44. Dim n As Double = 1.2345 lstOutput.Items.Add(Round(n, 2))

In Exercises 45 and 46, rewrite the code using one line. 45. Dim quantity As Integer quantity = 12

46. Dim m As Integer Dim n As Double m = 2 n = 3

In Exercises 47 through 52, find the value of the given function. 47. Int(10.75) 50. Math.Sqrt(64)

48. Int19 - 22 51. Math.Round(3.1279, 3)

49. Math.Sqrt13 * 122 52. Math.Round1- 2.62

In Exercises 53 through 58, find the value of the given function where a and b are numeric variables of type Double, a = 5 and b = 3. 53. Int1-a > 22 56. Math.Sqrt14 + a2

54. Math.Round(a / b) 57. Math.Round1a + .52

55. Math.Sqrt1a - 52 58. Int1b * 0 .52

In Exercises 59 through 66, write an event procedure starting with a Private Sub btnCompute_Click1 Á 2 Handles btnCompute.Click statement, ending with an End Sub statement, and having one line for each step. Lines that display data should use the given variable names. 59. The following steps calculate a company's profit: (a) Declare all variables. (b) Assign the value 98456 to the variable revenue. (c) Assign the value 45000 to the variable costs. (d) Assign the difference between the variables revenue and costs to the variable profit. (e) Display the value of the variable profit in a list box. 60. The following steps calculate the amount of a stock purchase: (a) Declare all variables. (b) Assign the value 25.625 to the variable costPerShare. (c) Assign the value 400 to the variable numberOfShares. (d) Assign the product of costPerShare and numberOfShares to the variable amount. (e) Display the value of the variable amount in a list box.

61. The following steps calculate the price of an item after a 30% reduction: (a) Declare all variables. (b) Assign the value 19.95 to the variable price. (c) Assign the value 30 to the variable discountPercent. (d) Assign the value of (discountPercent divided by 100) times price to the variable markDown. (e) Decrease price by markdown. (f) Display the value of price in a list box. 62. The following steps calculate a company's break-even point, the number of units of goods the company must manufacture and sell in order to break even: (a) Declare all variables. (b) Assign the value 5000 to the variable fixedCosts. (c) Assign the value 8 to the variable pricePerUnit. (d) Assign the value 6 to the variable costPerUnit. (e) Assign the value fixedCosts divided by (the difference of pricePerUnit and costPerUnit) to the variable breakEvenPoint. (f) Display the value of the variable breakEvenPoint in a list box. 63. The following steps calculate the balance after three years when $100 is deposited in a savings account at 5% interest compounded annually: (a) Declare all variables. (b) Assign the value 100 to the variable balance. (c) Increase the variable balance by 5% of its value. (d) Increase the variable balance by 5% of its value. (e) Increase the variable balance by 5% of its value. (f) Display the value of the variable balance in a list box. 64. The following steps calculate the balance at the end of three years when $100 is deposited at the beginning of each year in a savings account at 5% interest compounded annually: (a) Declare all variables. (b) Assign the value 100 to the variable balance. (c) Increase the variable balance by 5% of its value, and add 100. (d) Increase the variable balance by 5% of its value, and add 100. (e) Increase the variable balance by 5% of its value. (f) Display the value of the variable balance in a list box. 65. The following steps calculate the balance after 10 years when $100 is deposited in a savings account at 5% interest compounded annually: (a) Declare all variables. (b) Assign the value 100 to the variable balance. (c) Multiply the variable balance by 1.05 raised to the 10th power. (d) Display the value of the variable balance in a list box. 66. The following steps calculate the percentage profit from the sale of a stock: (a) Declare all variables. (b) Assign the value 10 to the variable purchasePrice.

(c) Assign the value 15 to the variable sellingPrice. (d) Assign, to the variable percentProfit, 100 times the value of the difference between sellingPrice and purchasePrice divided by purchasePrice. (e) Display the value of the variable percentProfit in a list box. In Exercises 67 through 72, write an event procedure to solve the problem and display the answer in a list box. The program should use variables for each of the quantities. 67. Suppose each acre of farmland produces 18 tons of corn. How many tons of corn can be produced on a 30-acre farm? 68. Suppose a ball is thrown straight up in the air with an initial velocity of 50 feet per second and an initial height of 5 feet. How high will the ball be after 3 seconds? Note: The height after t seconds is given by the expression -16t2 + v0t + h0, where v0 is the initial velocity and h0 is the initial height. 69. If a car left Washington, D.C., at 2 o'clock and arrived in New York at 7 o'clock, what was its average speed? Note: New York is 233 miles from Washington. 70. A motorist wants to determine her gas mileage. At 23,352 miles (on the odometer), the tank is filled. At 23,695 miles the tank is filled again with 14 gallons. How many miles per gallon did the car average between the two fillings? 71. A U.S. geological survey showed that Americans use an average of 1600 gallons of water per person per day, including industrial use. How many gallons of water are used each year in the United States? Note: The current population of the United States is about 304 million people. 72. According to FHA specifications, each room in a house should have a window area equal to at least 10 percent of the floor area of the room. What is the minimum window area for a 14-ft by 16-ft room? Solutions to Practice Problems 3.1 1. 24. Multiplications are performed before additions. If the intent is for the addition to be performed first, the expression should be written 12 + 32 * 4. 2. The first assignment statement assigns the value of the variable var2 to the variable var1, whereas the second assignment statement assigns var1's value to var2. 3.

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim a, b, c As Double a = 3 b = 4 c = a + b a = c * a lstResults.Items.Add(a - b) b = b * b End Sub

a

b

c

0 3 3 3 21 21 21

0 0 4 4 4 4 16

0 0 0 7 7 7 7

Each time an assignment statement is executed, only one variable (the variable to the left of the equal sign) has its value changed. 4. Each of the three following statements increases the value of var by 5%. var = var + 0.05 * var var = 1.05 * var var += 0.05 * var

3.2

Strings

The most common types of data processed by Visual Basic are numbers and strings. Sentences, phrases, words, letters of the alphabet, names, telephone numbers, addresses, and Social Security numbers are all examples of strings. Formally, a string literal is a sequence of characters that is treated as a single item. String literals can be assigned to variables, displayed in text boxes and list boxes, and combined by an operation called concatenation (denoted by &).

■ Variables and Strings A string variable is a name used to refer to a string. The allowable names of string variables are the same as those of numeric variables. The value of a string variable is assigned or altered with assignment statements and displayed in a list box like the value of a numeric variable. String variables are declared with statements of the form Dim varName As String

Example 1 The following code shows how assignment statements and the Add method are used with strings. The string variable president is assigned a value by the third line and this value is displayed by the sixth line. The quotation marks surrounding each string literal are not part of the literal and are not displayed by the Add method. (The form for this example contains a button and a list box.) Private Sub btnDisplay_Click(...) Handles btnDisplay.Click Dim president As String president = "George Washington" lstOutput.Items.Clear() lstOutput.Items.Add("president") lstOutput.Items.Add(president) End Sub

[Run, and then click the button. The following is displayed in the list box.] president George Washington

If x, y, Á , z are characters and strVar is a string variable, then the statement strVar = "xy...z"

assigns the string literal xy Á z to the variable and the statement

◆

◆

Chapter 3 Variables, Input, and Output lstBox.Items.Add("xy...z")

or lstBox.Items.Add(strVar)

displays the string xy Á z in a list box. If strVar2 is another string variable, then the statement strVar2 = strVar

assigns the value of the variable strVar to the variable strVar2. (The value of strVar will remain the same.) String literals used in assignment or lstBox.Items.Add statements must be surrounded by quotation marks, but string variables are never surrounded by quotation marks.

■ Option Explicit and Option Strict Option Explicit and Option Strict are two options that affect programming. Throughout this book, we assume that both options are in effect. Having these two options working is considered good programming practice. Option Explicit requires that all variables be declared with Dim statements. The absence of this option can lead to errors resulting from the misspelling of names of variables. Option Strict requires explicit conversions in most cases where a value or variable of one type is assigned to a variable of another type. The absence of this option can lead to data loss. Visual Basic provides a way to enforce Option Explicit and Option Strict for all programs you create. Press Alt/ Tools/Options to open the Options dialog box. In the left pane, click on the symbol (+ or 覽) to the left of Projects and Solutions to expand that entry. Then click on the subentry VB Defaults. Four default project settings will appear on the right. (See Figure 3.1.) If the settings for Option Explicit and Object Strict are not already set to On, change them to On. Note: Option Infer is discussed in Chapter 6.

FIGURE 3.1

Option default project settings.

3.2

Strings

■ Using Text Boxes for Input and Output The content of a text box is always a string. Therefore, statements such as strVar = txtBox.Text

and txtBox.Text = strVar

can be used to assign the contents of the text box to the string variable strVar and vice versa. Numbers typed into text boxes are stored as strings. With Option Strict set to On, such strings must be converted to numeric values before they can be assigned to numeric variables or used in numeric expressions. The functions CDbl and CInt convert strings representing numbers into numbers of type Double and Integer, respectively. Going in the other direction, the function CStr converts a number into a string representation of the number. Therefore, statements such as dblVar = CDbl(txtBox.Text)

and txtBox.Text = CStr(dblVar)

can be used to assign the contents of a text box to the Double variable dblVar and vice versa. CDbl, CInt, and CStr, which stand for “convert to Double,” “convert to Integer,” and “convert to String,” are referred to as data-conversion or type-casting functions. Example 2 by the user:

The following program computes the sum of two numbers supplied

OBJECT

PROPERTY

SETTING

frmAdd lblFirstNum

Text AutoSize Text

Addition False First Number:

AutoSize Text

False Second Number:

Text Text ReadOnly

Compute Sum Sum: True

txtFirstNum lblSecondNum txtSecondNum btnCompute lblSum txtSum

Private Sub btnCompute_Click() Handles btnCompute.Click Dim num1, num2, sum As Double num1 = CDbl(txtFirstNum.Text) num2 = CDbl(txtSecondNum.Text) sum = num1 + num2 txtSum.Text = CStr(sum) End Sub

◆

79

80

◆

Chapter 3 Variables, Input, and Output

[Run, type 45 into the first text box, type 55 into the second text box, and click on the button.]

■ Auto Correction The Auto Correction feature of IntelliSense suggests corrections when errors occur, and allows you to select a correction to be applied to the code. When a prohibited statement is entered, a blue squiggly error line appears under the offending part of the statement. If the squiggly line has a short red line segment at its right end, the Auto Correction feature is available for the error. When you hover the cursor over the squiggly line, a small box containing an exclamation mark inside a red circle appears. Clicking on the small box produces an Auto Correction helper box that describes the error and makes a suggestion for fixing it. Figure 3.2 shows a typical Auto Correction helper box for a data-type-conversion error. If you click on the line beginning "Replace," the change will be made for you.

FIGURE 3.2

An Auto Correction helper box.

Источник: https://dokumen.pub/an-introduction-to-programming-using-visual-basic-2008-with-visual-studio-expression-edition-dvd-7th-ed-9780136060727-2008023532-0136060722.html
10

Источник: http://www.geometry.net/detail/basic_v/visual_basic_programming_page_no_10.html
6 10

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Visual Basic Programming: **more books (100)**

1. **Visual Basic Programming for the Absolute Beginner w/CD (For the Absolute Beginner (Series).)** by **Michael Vine**, 2002-07-01
2. **Programming In Visual Basic.net** by **Julia Case Bradley**, **A. C. Millspaugh**, 2005-05
3. **MicrosoftVisual Basic: Game Programming for Teens** by **Jonathan S. Harbour**, 2007-09-25
4. **Programming in Visual Basic 2010: The Very Beginner's Guide** by **Jim McKeown**, 2010-03-29
5. **Clearly Visual Basic: Programming with Microsoft Visual Basic 2008** by **Diane Zak**, 2008-08-15
6. **Programming Visual Basic 2008: Build .NET 3.5 Applications with Microsoft's RAD Tool for Business** by **Tim Patrick**, 2008-05-27
7. **Introduction to Programming Using Visual Basic 2010 (8th Edition)** by **David I. Schneider**, 2010-08-30
8. **Practical Database Programming with Visual Basic.NET** by **Ying Bai**, 2008-10-20
9. **Programming in Visual Basic 2010** by **Julia Case Bradley**, **Anita Millspaugh**, 2010-07-28
10. **Programming in Visual Basic 2008** by **Julia Case Bradley**, **Anita Millspaugh**, 2008-05-30
11. **Introduction to Programming Using Visual Basic 2008, An (w/VS2008 DVD) (7th Edition)** by **David I. Schneider**, 2008-07-13
12. **Visual Basic Game Programming for Teens** by **Jonathan S. Harbour**, 2004-12-21
13. **Programming Microsoft Visual Basic 6.0** by **Francesco Balena 196**, 1999-05-26
14. **Visual Basic Game Programming with DirectX (The Premier Press Game Development Series)** by **Jonathan S. Harbour**, 2002-01-02

181. Basic Books And Tutorials
The CDROM features custom applications, code, and scores of valuable visual Basicprogramming aids, ready-to-run examples of advanced VB applets, and a dozen
http://www.mathtools.net/Basic/Books_and_Tutorials/

183. EliteVB
visual basic News. MCPCentral announces NetTools TCP/IP Tool Chest; Connections.NET Tour; Dev4Dev Beta Testers Needed! devMail.Net 2.1 released.
http://www.extreme-vb.net/

184. ProgrammingMSAccess.Com
Samples, book excerpts, presentations, FAQs, tutorials on Access 2000 and the other Office 2000 component Category Computers Software Databases MS Access...... On top of that, the first new member to join the DDG in March wins a copy ofProgramming Microsoft visual basic .NET for Microsoft Access Databases.
http://www.programmingmsaccess.com/

185. DevX
Oracle Has SQL Server Beat in Stored Procedure programming Oracle9i and SQL Server2000 add enhancements to the standard SQL language that enable you to use
http://www.devx.com/

187. Vb World
Developer.com Earthweb The Web CodeGuru Gamelan Jars Wireless Discussions Architecture Design Database DHTML Java Languages Tools Microsoft .NET Open Source Project Management Techniques Voice Web Services Wireless/Mobile XML Be a
http://www.vbworld.com/

188. TechTarget, The Most Targeted IT Media
Portal site with links, advice and a search engine designed specifically to search for application Category Computers Software Portals......Visit Our Targeted IT Media
http://www.techtarget.com/

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Page 10 181-188 of 188 Back 1,340g

- 19 Feb 2009
- Pearson Education (US)
- Pearson
- Upper Saddle River, NJ, United States
- English
- 7th edition
- 0136060722
- 9780136060727

Back cover copy

This revision of Schneider's best-selling guide is designed for readers with no prior programming experience. It focuses on developing good problem-solving skills, building a strong foundation that will give readers a sustainable understanding of programming. KEY TOPICS: Based on Visual Basic 2008, the book starts with a brief review of the preliminaries of Windows, and then focuses on problem-solving. A broad range of real-world examples, section-ending exercises, case studies, and programming projects gives readers significant hands-on experience. A useful reference for both beginning programmers and those moving from another language and that want to learn more about programming with the latest version of Microsoft's Visual Basic.
show more

Table of contents

show more

About David I. Schneider

David I. Schneider spent over 32 years teaching at the University of Maryland, and has published an impressive 28 books over the past 18 years. He is the author of several best-selling texts, including his Visual Basic Series with Prentice Hall. Why are his books best-sellers? Students love his extensive use of examples and applications-a concept that stems from his teaching experience and results in books that are "tried and true." His best-selling Visual Basic texts have been successfully used for many years and have received consistent praise from both students and instructors.
show more

Rating details

12 ratings

3.17 out of 5 stars

5 25% (3)

4 17% (2)

3 33% (4)

2 0% (0)

1 25% (3)

Book ratings by Goodreads

Goodreads is the world's largest site for readers with over 50 million reviews. We're featuring millions of their reader ratings on our book pages to help you find your new favourite book. Close X

{"bd_js_shop":"Shop","bd_js_too_long_for_shipping_label":"Sorry, that's too long for our shipping labels","bd_js_why_are_you_leaving":"Why are you leaving?","bd_omnibus_ph1_price_statement_suffix":"vs RRP","bd_js_too_long":"Sorry, that's too long","bd_js_could_not_find_address_try_again":"Sorry, we couldn't find the address. Please enter manually:","bd_js_keep_typing_to_refine_search_results":"Keep typing to refine the search results","bd_pre_order":"Pre-order","bd_js_top_categories":"Top Categories","bd_price_save":"Save {0}","bd_notify_me":"Notify me","bd_js_name_only_letters":"Sorry, full name can only contain letters","bd_js_show_more":"show more","bd_js_enter_valid_email_address":"Please enter a valid email address","bd_js_delivery_time":"Delivery time","bd_js_enter_address_manually":"Enter address manually","bd_js_more_categories":"More Categories","bd_30_day_low_price_carousel":"30-day low price","bd_js_continue_shopping":"Continue Shopping","bd_js_account_and_help":"Account & Help","bd_js_price":"Price","bd_js_basket_checkout":"Basket / Checkout","bd_add_to_basket":"Add to basket","bd_js_enter_first_last_name":"Please enter a first and last name","bd_js_please_enter_your":"Please enter your"}

Источник: https://www.bookdepository.com/Introduction-Programming-Using-Visual-Basic-2008-w-VS2008-DVD-David-I-Schneider/9780136060727?ref=grid-view&qid=1572988478363&sr=1-358 1 3 9 7 9 1

Free delivery worldwide on all books from Book Depository

5 2

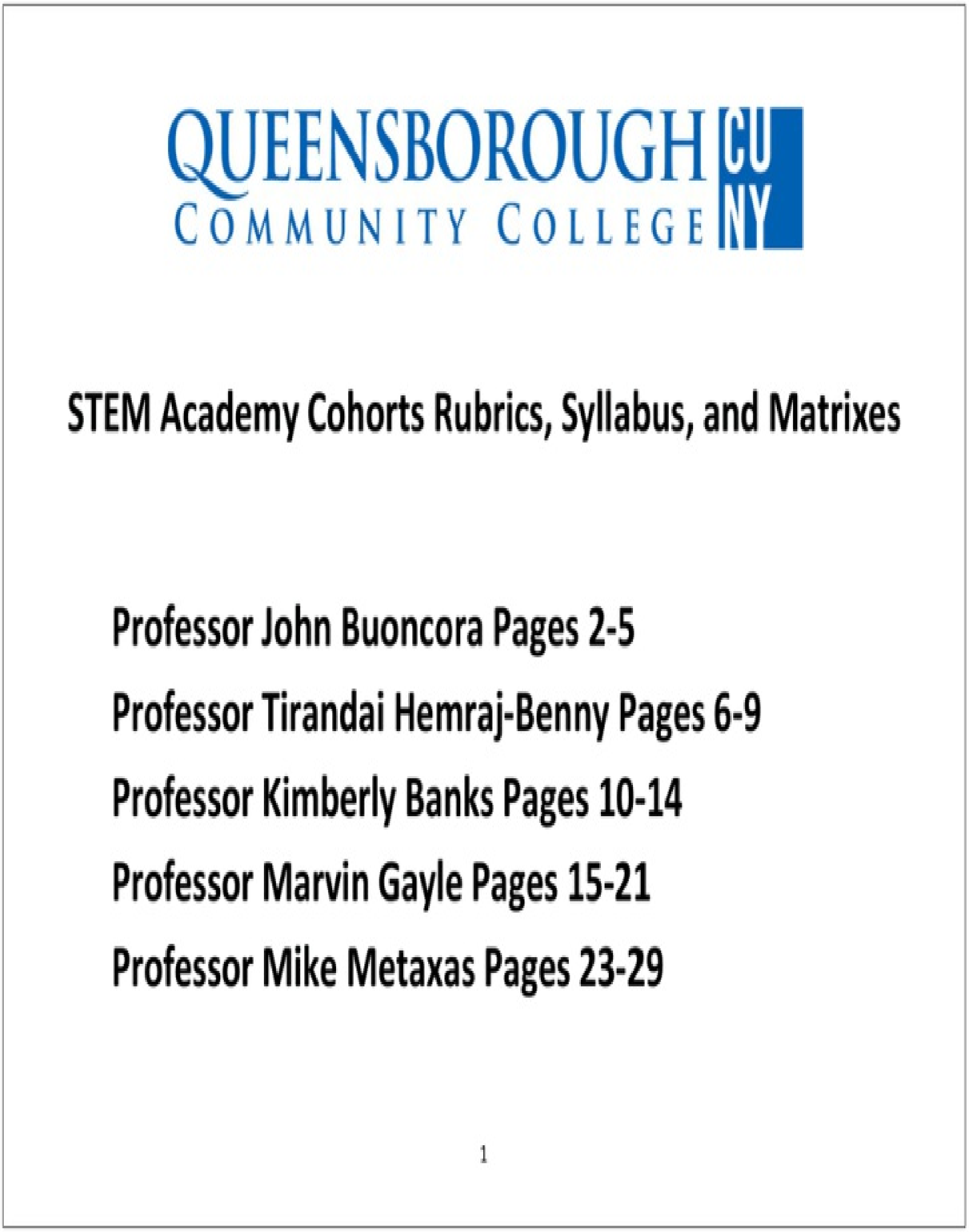
PROGRAMMING IN VISUAL BASIC 2008

Pemrograman merupakan salah satu mata kuliah dalam kurikulum Program Studi Manajemen Informatika AMIK Indonesia Banda Aceh. Mata kuliah ini bertujuan agar mahasiswa mampu memahami konsep dasar pemrograman windows dan mampu menerapkan konsep pemrograman Visual BASIC dalam mengembangkan perangkat lunak berbasis windows. Sesuai dengan tujuan tersebut, buku ajar ini tersusun atas beberapa bagian yakni dimulai dari pengenalan .NET dan Framerwok, kemudian pada bagian berikutnya dibahas mengenai Visual Studio 2015, dasar pemrograman Visual BASIC, basis data dalam pemrograman Visual BASIC, Microsoft Office Acces 2016, SQL Server 2015, Crytal Report dan Compiler. Buku ajar ini ditujukan kepada khalayak yang ingin memahami mengenai konsep pemrograman BASIC dan pengembangan perangkat lunak berbasis Windoes, terutama bagi mahasiswa Program Studi Manajemen Informatika AMIK Indonesia Banda Aceh. Buku ajar ini diharapkan dapat menjadi acuan untuk membantu mahasiswa dalam perkuliahan. Buku ajar ini dapat diselesaikan dengan baik berkat dukungan dari berbagai pihak. Penulis dalam kesempatan ini ingin menyampaikan ucapan terima kasih kepada semua pihak dan terutama kepada AMIK Indonesia Banda Aceh. Semoga buku ajar ini dapat bermanfaat bagi semua pihak yang membutuhkan. Penulis menyadari, dalam penulisan buku ajar ini masih banyak terdapat kekurangan. Penulis sangat mengharapkan saran yang bersifat membangun demi kesempuranaan buku ajar ini.

Источник: https://www.academia.edu/11436009/PROGRAMMING_IN_VISUAL_BASIC_2008
7 768 pages

- 203 x 254 x 30.48mm 3 10

Источник: http://www.geometry.net/detail/basic_v/visual_basic_programming_page_no_10.html
1 1



QUEENSBOROUGH COMMUNITY COLLEGE CUNY

STEM Academy Cohorts Rubrics, Syllabus, and Matrixes

Professor John Buoncora Pages 2-5

Professor Tirandai Hemraj-Benny Pages 6-9

Professor Kimberly Banks Pages 10-14

Professor Marvin Gayle Pages 15-21

Professor Mike Metaxas Pages 23-29

STEM A

Academ my Coh horts R Rubrics,, Syllab bus, an nd Matrrixes Professor John B Buonco ora Pagges 2-5 dai Hem mraj-Be enny Paages 6--9 Professor Tirand erly Ban nks Pagges 10--14 Professor Kimbe n Gayle e Pagess 15-211 Professor Marvin Metaxaas Page es 23-229 Professor Mike M 1 Assessment for STEM Academy – Scoring Rubrics Instructor: John Buoncora Course: ET 110 (Electrical Circuit Analysis I) General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Use the varied forms of mathematical communication: language, symbolic notation, graphs, charts. Assessment Plan and Scoring Rubrics: Assessment Plan &amp; Scoring Rubrics 0 (Lowest score) 1 2 3 4 (Highest Score) a: Only able to find the equivalent (total) resistance of the entire circuit of the problem. b: a) plus able to find the source current or voltage, and able to find “a few” of the resistor currents or voltages in the problem. c: b) plus able to find all of the currents, voltages and powers for the problem. d: c) and able to check the results using Kirchhoff’s Laws and conservation of energy in the problem. ET 110 Exam 2 Problem 4: The student is unable to find the total resistance of Competency/ Exercise the circuit. Description: Analyze a series-parallel resistive circuit with a single DC voltage source using equivalent resistances, KVL, KCL, and Ohm’s Law. 2 General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Application of mathematics to appropriate fields of study. Assessment Plan and Scoring Rubrics: Assessment Plan &amp; 0 (Lowest score) 1 2 3 4 (Highest Score) Scoring Rubrics ET 110 Exam 3 Problem The student is unable 1: to write any of the Competency/ mesh (loop) equations. Exercise Description: Use Mesh (Loop) analysis to analyze a circuit containing multiple voltage sources and/or current sources. a: Could only write the mesh (loop) equations in a partially correct manner for the problem. b: a) plus able to write all of the mesh equations correctly for the problem. c: b) plus able to partially solve the mesh equations and solve for at least one quantity correctly in the problem. d: c) plus able to completely solve the mesh equations and correctly solve for all quantities in the problem. Able to check all of the problem results on the original circuit diagram using KVL, KCL, and Ohms' Law. 3 General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make informed decisions. Student Learning Outcome/Competency: Differentiate between facts, assumptions, and conclusions in the formulation of a proposed solution or answer. Compare the way questions, issues or problems are formulated within various fields of study. Assessment Plan and Scoring Rubrics: Assessment Plan &amp; Scoring Rubrics ET 110 Lab 8 (Series- Parallel Circuits &amp; In- Circuit Resistance Measurement): Competency/ Exercise description: Construct Series-Parallel resistive circuits on a breadboard and apply a DC voltage source. Measure equivalent circuit resistances with power off and measure voltages and currents with power on. Calculate the theoretical values of all circuit quantities and simulate the circuit operation. 0 (Lowest score) 1 The student is unable a: Student was able to to construct any of construct only the the circuits. "simple" series-parallel circuit and the wiring may have contained errors. Assessment for STEM Academy – Summary Rubric Score Sheet 4 2 3 4 (Highest Score) b: a) without any errors and student was able to measure all circuit voltages with power applied. The student was also able to measure equivalent circuit resistances with power off. c: b) and the student was able to measure all circuit currents without blowing a fuse in the DMM. Student was also able to construct the "more complicated" series- parallel circuit. The student was able to determine if a circuit was operating within specifications as compared to the calculated values. d) and the student was able to measure all voltages and currents in "more complicated" series- parallel circuits without any issues. The student was able to calculate all circuit currents and voltages in both series-parallel circuits and simulate all circuits successfully. The student was able to compare measured and calculated values using %differences and explain the differences. Instructor: John Buoncora Course: ET 110 (Electrical Circuit Analysis I) Competency General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Use the varied forms of mathematical communication: language, symbolic notation, graphs, charts. Competency/Exercise Description: Analyze a series-parallel resistive circuit with a single DC voltage source using equivalent resistances, KVL, KCL, and Ohm's Law. General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Student Learning Outcome/Competency: Application of mathematics to appropriate fields of study. Competency/Exercise Description: Use Mesh (Loop) analysis to analyze a circuit containing multiple voltage sources and/or current sources. General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make informed decisions. Student Learning Outcome/Competency: Differentiate between facts, assumptions, and conclusions in the formulation of a proposed solution or answer. Compare the way questions, issues or problems are formulated within various fields of study. Competency/Exercise description: Construct Series-Parallel resistive circuits on a breadboard and apply a DC voltage source. Measure equivalent circuit resistances with power off and measure voltages and currents with power on. Calculate the theoretical values of all circuit quantities and simulate the circuit operation. 0 (Lowest Score) 0 The number of students in each category is indicated in the table above. Notes: The first and second rows correspond to an ET 110 Lecture sections, with 34 students assessed. The third row corresponds to an ET 110 Lab sections, with 17 students assessed. 5 0 0 1 2 3 2 4 3 1 3 2 9 8 5 4 (Highest Score) 19 20 9 Assessment for STEM Academy Instructor: Tirandai Hemraj-Benny Course: CH-121 – Fundamentals of Chemistry Laboratory General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Used the varied forms of mathematical communication: language, symbolic notation, graphs, charts, to formulate quantitative ideas and patterns. Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Lab Report #4: The student is unable to 1 2 The student correctly assigns each axis but The student is able to plot the data but cannot does not understand that draw complete conclusions. Axes are each axis should start correctly assigned and Student will perform assign each axis, does from zero and even divided but not labeled. an experiment to not label axes and does divisions should follow. investigate the not understand that each Thus, correct conclusions cannot be drawn. relationship between axis should start from zero and even divisions distance from a should follow. radioactive source and Properties of Natural Radioactivity Description: degree of radioactive signal. Student will represent their data on a graph and draw conclusions. plot the data and draw conclusions. Student does not correctly 6 3 4 The student is able to plot the data but cannot draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. The student is able to plot the data and draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. Quiz #2: Description Student will be given radioactive data to plot on a graph and draw conclusions after graded lab reports have been returned. The student is unable to plot the data and draw conclusions. Student does not correctly The student correctly assigns each axis but does not understand that each axis should start from zero and even assign each axis, does divisions should follow. not label axes and does Thus, correct conclusions not understand that each cannot be drawn. axis should start from zero and even divisions should follow. The student is able to plot the data but cannot draw complete conclusions. Axes are correctly assigned and divided but not labeled. The student is able to plot the data but cannot draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. Assessment Data: # of students The student is able to plot the data and draw complete conclusions. All axes are labeled and student understands that each axis should start from zero and even divisions should follow. Lab Report 0 3 1 2 2 2 3 1 4 0 2 3 3 0 4 4 Lab Quiz # of students 0 1 1 0 7 Instructor: Tirandai Hemraj-Benny Course: CH-120 – Fundamentals of Chemistry Lecture General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Identify problems that need a mathematical solution, and use computational methods in the mathematics applicable in everyday life. Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Exam #4: The student is not aware Question #4 that mathematical calculation is necessary Description: Student will be asked and cannot determine the oxidation numbers of to assign the any element. oxidation number of each element in compounds 1 2 3 4 The student is aware that mathematical calculation is necessary but cannot determine the oxidation numbers of any of the elements correctly. The student is aware that mathematical calculation is necessary and determines the oxidation numbers of some of the elements correctly. The student is aware that mathematical calculation is necessary and determines the oxidation numbers of most of the elements correctly. The student is aware that mathematical calculation is necessary and determines the oxidation numbers of all elements correctly. Assessment Data: # of students 0 0 1 2 2 3 8 3 6 4 0 Instructor: Tirandai Hemraj-Benny Course: CH-120 – Fundamentals of Chemistry Lecture General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make an Informed decisions. Specific Objective (Outcome): Assessment Plan and Rubrics: Assessment Plan 0 &amp; Rubrics EXAM #4: The student does not identify the specific Question #3 topic or problem, Description: where they are unable Students will be to predict the products asked to provide formed and represent the molecular, ionic and net ionic the reaction in the equations when a three types of equations. carbonate and a nitrate compound are mixed together. Assessment Data: # of students 0 2 Distinguish the problem or question from a proposed solution. 1 2 4 The student identifies the specific topic or problem, where they are able to predict the products formed but cannot represent the reaction in the three types of equations. The student identifies the specific topic or problem, where they are able to predict the products formed and can represent the reaction in the three types of equations to some extent. The student identifies the specific topic or problem, where they are able to predict the products formed and can represent the reaction in the three types of equations with a few mistakes. The student identifies the specific topic or problem, where they are able to predict the products formed and represent the reaction in the three types of equations correctly. 1 1 2 3 9 3 3 5 4 0 Instructor: Course: Kimberly Banks EN-101 English Composition CUNY Proficiency Examination Task 1 Scoring Guide A. Develops an essay that presents a focused response to the writing assignment, making appropriate and coherent connections among all parts of the assignment. B. Demonstrates understanding of the readings through summary and explanation of relevant material. C. Incorporates, as support for own thoughts, references to the readings, identifying the sources formally or informally. 6 Addresses the writing assignment fully, analytically, and perhaps critically or imaginatively, with superior focus and coherence. 6 Demonstrates superior and perhaps critical understanding of readings through accurate summary, full explanation, and insightful analysis of relevant sections. 5 Addresses the writing assignment fully and analytically, with strong focus and coherence. 4 Addresses all parts of the writing assignment with adequate focus and coherence throughout. 5 Demonstrates strong understanding of readings through accurate summary, with appropriate explanation and analysis of relevant sections. 6 Makes insightful connections and distinctions between readings and own ideas; integrates references smoothly into own essay and identifies them consistently and correctly. 5 Makes analytical connections and perhaps distinctions between readings and own ideas; integrates references into own essay and identifies them consistently and correctly. 4 Demonstrates overall understanding of readings through appropriate summary and explanation, with some analysis. 4 Makes and explains appropriate connections between readings and own ideas; identifies references consistently and correctly. 6 Communicates with precision and enhanced expression through highly effective use of vocabulary and sentence variety; infrequent, if any, lapses in use of conventions. 5 Communicates effectively throughout the essay, with few lapses in use of conventions. 10 D. Communicates clearly and effectively, using appropriate conventions of language (e.g., grammar, spelling, punctuation). 4 Communicates clearly throughout the essay; sentences may contain some lapses in use of conventions, but these rarely impede comprehension. 3 Addresses all or most parts of the writing assignment adequately, but focus may lapse briefly or connections may be missing. 3 Demonstrates generally accurate understanding of readings although summary or explanation may be incomplete or not fully relevant. 2 Addresses some parts of the writing assignment or addresses all parts superficially; focus or coherence may break down at several points. 2 Demonstrates partial understanding of the readings through summary or explanation, but understanding is flawed or explanation is incomplete. 1 Shows little or no ability to address the writing assignment; may not link thoughts between paragraphs. 3 Makes some connections between readings and own ideas but they may not all be appropriate or adequately explained; identifies most references consistently and correctly. 3 Generally communicates clearly throughout the essay although lapses in use of conventions may at times impede comprehension or prove distracting. 2 Makes few or unwarranted connections between readings and own ideas; may identify references inconsistently or incorrectly. 2 Communicates clearly at times, showing some ability to use conventions, but whole sections are unclear or errors frequently impede comprehension. 1 Demonstrates little or no understanding of text. 1 Makes no reference to background reading or makes no distinctions between background reading and own ideas. 11 1 Communicates little because few sentences demonstrate appropriate use of conventions. CPE Quick Summary Initial Proficiencies Scores A 6 3 5 1 4 4 3 8 2 6 1 0 B 2 2 5 5 8 0 C 0 0 1 2 6 13 D 0 0 2 9 4 5 B 4 2 4 10 1 0 C 2 7 6 5 2 0 D 1 4 6 8 3 0 Final Proficiencies Scores A 6 3 5 5 4 5 3 6 2 2 1 0 12 Prof. Kimberly J. Banks EN 101 Composition I, section D13 11/23/09 In-Class Essay #2 Using the selection from Henry I. Miller and Gregory Conko's The Frankenfoood Myth: How Protest and Politics Threaten the Biotech Revolution and Carol Tucker Foreman's "Killing the 'Frankenfood' Monster: How People Can Love, Not Fear, Biotech Food," discuss the advantages and disadvantages associated with biotechnology. In your essay, summarize Miller and Conko's key points distinguishing between the myths and realities of biotechnology. Draw a strong relationship between Miller and Conko's thinking and what you have just read about the need to reassure the public in Tucker Foreman's article. In light of the reading selections, discuss your own knowledge of or attitudes towards biotechnology. What experiences have you had with biotechnology and have they made your life better or worse? Also discuss the degree to which your perspective reflects the ideas of either or both writers. 13 In-Class Essay #2 CONTENT (75%) To what extent does the student address the assignment? How focused and coherent is the answer? Points Earned To what extent does the student demonstrate understanding of the reading? At what level does the student summarize, explain, and analyze evidence? How regularly does the student distinguish between his/her ideas and those of the authors in the readings? How well are the references integrated into the essay? To what extent are the references identified? How clear and effective is the essay? How well are paragraphs and sentences structured? MECHANICS (25%) What is the size and appropriateness of the student's vocabulary? To what extent are words spelled appropriately? How effective is the student's use of grammar? How appropriate and adequate is the punctuation to effectively help convey meaning? Total 14 Points Possible 25 25 25 10 5 5 5 100 Assessment for STEM Academy Instructor: Marvin Gayle Course: ET-502 – Introduction to Programming General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Application of mathematics to appropriate fields of study. Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Lab Experiment:: The student is unable to start the Lab_#7 software tool and Description: setup the Visual Student write a Visual basic program that takes Basic Software environment IDE. two inputs Resistance and voltage and then develops a software program to calculate and output the current to and appropriate number of digits. 1 2 3 4 The student is able to start the software tool and setup the Visual Basic Software environment IDE but is unable to read the user inputs without an error. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to read the user inputs but unable to display the appropriate answer. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to read the user inputs, and able to display the appropriate answer, but user interface is not balanced and appears unbalanced The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to read the user inputs, and able to display the appropriate answer, but user interface is not balanced and appear well balanced. 15 Instructor: Marvin Gayle Course: ET-502 – Introduction to Programming General Education Objective 2: Use analytical reasoning to identify issues or problems and evaluate evidence to make informed decisions. Specific Objective (Outcome): Differentiate between facts, assumptions, and conclusions in the formulation of a proposed solution or answer. Compare the way questions, issues or problems are formulated within various fields of study Assessment Plan and Rubrics: Assessment Plan &amp; 0 Rubrics Lab Experiment:: The student is unable to Lab_#9 start the software tool and setup the Visual Description: Student writes a Visual basic program that will be used to display ONLY the numbers that are perfect Squares, within a given range of numbers. Basic Software environment IDE. 1 2 3 4 The student is able to start the software tool and setup the Visual Basic Software environment IDE but is unable to develop a correct algorithm to get to an end solution The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to develop a correct algorithm but unable to successfully implement it. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to develop a correct algorithm, is able to successfully implement it, but is unable to decide in the implementation which results to display at the end of the program. The student is able to start the software tool and setup the Visual Basic Software environment IDE, is able to develop a correct algorithm, is able to successfully implement it, and is able to properly display the results to display at the end of the program. 16 P Professor Marvvin Gayle ET – 502 Introduction to Proggramming O Office Location: Technology Bu uilding Room T-2 20 TTelephone: 718 8-631-6207 EEmail: mgayle@ @qcc.cuny.edu C Course Descripttion: IIntroduction to the Visual BASIC P Programming lan nguage with application problems in electrical an d Computer Engineering Technollogy; with h hands-on experie ence in the Department's Computer Center. Hours Prerequisites Corequisites General Educatio on Objectives Course Objectives 1 Cred dit, 3 Laboratoryy Hours ET–501 Computer Applications Demonstrate masteryy of discipline-sp pecific knowledge, skills and toools required for entry into or advanceme ent in the job m market in their fi eld (caree er programs) Use an nalytical reasoning skills and ap pply logic to solvve problems This course introducees the student to fundamental prrogramming concep pts and structurees such as decision making and d loops. They w will need th hese skills for any a problem solv ving (algorithm)) and for future coursees like ET503 an nd ET560. Textbooks Introduction to Programming Using Visual Basic 2008" (w/VS2008 DVD D), 7th Edition Author David I. Schn neider Publish her Prenticce Hall ISBN ISBN-10: 013606 60722, ISBN-13: 978013 36060727 17 Department of Electrical and Computer Engineering Technology ET-502 – Introduction to Programming Fall 2009 ET_502 COURSE OUTLINE Prerequisite: ET-501 Week No Topic Reading Introduction to Blackboard. Overview of computers and computer languages: applications programming, high level/low level, compiled/interpreted. Problem solving and flowcharting. Class notes pp. 1 – 18 pp. 19 – 24 Appendix B 1 How to VB2008 Express Edition, saving and naming forms and projects. 2 Introduction to the Visual Basic Environment: Objects, Properties. Internal Documentation (The Rem statement). pp. 19 – 43 pp. 85 – 86 Visual Basic Events: Using code to change Properties of Objects. pp. 44 - 54 3 4 pp. 61 - 70 Arithmetic Operations: Level of Precedence, Numeric Variables, Print method, Numeric Functions. Declaring Variable Types. 5 Strings: String Variables, Concatenation, Scope of Variables, Using Text Boxes for input (Val and Str functions), Using the Format function with the Print method. pp. 77 - 88 pp 95 – 104 6 Mid-Term Examination Relational Operators and String Relationships And as time permits: [The ANSI (or ASCII) Character Set. String and String Related Functions (Left, Mid, Right, UCase, Trim, Len, and InStr)] 7 Using a Message Box for Output General Procedures, Sub Procedures, Variables and Expressions as 18 p. 121 – 126 pp. 104 – 106 Arguments, Passing Variables Back from Sub Procedures. 8 Review of Relational Opreators and String Relationships pp. 121 – 127 Controlling Program Flow: If Blocks, Logical Operators pp. 129 - 136 9 Decision Structures (Select Case Blocks) pp. 146 - 154 10 Looping Structures: Do Loops and For…Next Loops pp. 237 – 242 pp 267 - 275 11 Class Note PBASIC – FOR…NEXT loop and LEDs 12 Introduction to the BASIC Stamp: Controlling Outputs Class Notes PBASIC – FOR…NEXT loop and LEDs 13 Introduction to the BASIC Stamp: Reading Switch Inputs 14 Final Examination Class Notes Note: It is important to read the "Comments" appearing at the conclusion of each section. 19 Course Outcomes Outcomes Outcome 1 Description an appropriate mastery of the knowledge, techniques, skills and modern tools of their disciplines Outcome 2 an ability to identify, analyze and solve technical problems Outcome 3 an ability to communicate effectively Laboratory Experiments Lab Week_1 Week_2 Week_3 Week_4 Week_5 Week_6 Week_7 Week_8 Week_9 Week_10 Week_11 Week_12 &amp; Week_13 Week_14 Topics Overview of computers and computer languages: applications/programming, high level/low level, compiled/interpreted. Problem solving, flow charts. Loading VB6 Working Edition, saving and naming forms and projects Introduction to the Visual Basic Environment: Objects, Properties Visual Basic Events: Using code to change Properties of Objects. REM statements. Arithmetic Operations: Level of Precedence, Numeric Variables Print into picture Box Declaring Variable Types, Val-and Str-functions pp Mid-Term Examination, Relational Operators Input from and InputBox, General Procedures, Passing Variables, Scope of Variables Controlling Program Flow: If Blocks, Logical Operators Controlling Program Flow, For…Next Loops Controlling Program Flow, Do Loops, error detection Basic Stamp by Parallax, programming a microcontroller with BASIC Final Exam 20 Grades Item 1 2 3 4 Description Weight There will be NO make up exam. A missed test or exam will count as a zero 25 % 25 % 25 % 25 % TOTAL 100 % Ten weekly online exams, each worth 2.5 % Ten weekly Lab assignments each worth 2,5 % One Mid-Term Exam One Final Exam Grade Scale A A- 100-96 95-90 B+ B B- C+ C C- D+ D D- F 89-87 86-84 83-80 79-77 76-74 73-70 69-67 66-64 63-60 59-0 21 Assessment for STEM Academy Instructor: Mike Metaxas Course: ET 710 (Building and Maintaining Web Sites) General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Use HTML and Notepad to create a functional Web page Assessment Plan and Rubrics: Assessment Plan &amp; 0 1 2 3 4 Rubrics Description: The student does not The student is able to use The students The student The student Notepad but does not Use Notepad to create have basic computer understands some of understands most of understands the skills and cannot use understand the HTML basics of HTML as a Web Page. the HTML but does not the HTML and the Notepad. concepts presented. Demonstrate a basic grasp the use of a use of a browser but presented including page structure and knowledge of HTML browser to view the is missing the tags and is able to including page web page just created concept of page create a functional structure and structure. Web page rudimentary HTML tags. View the Web page in a browser 22 General Education Objective 3: Reason quantitatively and mathematically as required in their fields of interest and everyday life. Specific Objective (Outcome): Apply Microsoft Expression Web Software to Web page design. Assessment Plan and Rubrics: Assessment Plan &amp; 0 1 2 3 4 Rubrics Description: The student does not The student is able use The student is able to The student can use The student can use Use Microsoft's have basic computer the Expression Web create a Web page Expression web and Expression web and Expression Web skills and cannot use software but does not work with it to work with it to create a using Expression Web. Software to create a the Expression Web understand a WYSIWYG create a basic web web page but does not He can use most of Web page. Understand software. (what you see is what you page but does not understand publishing the basic features to what the software does get) editor. add unique own style personalize the web and using the server. and apply it to the web page. Can pages. appropriately. preview and easily Understand how to preview a page from within Expression Web and how to create a web site on a flash drive. Connect to and publish files to the web server make changes. Publishes to the server 23 Electrical and Computer Engineering Technology Department Queensborough Community College The City University of New York ET710 - Building and Maintaining Web Sites Hours: 3 Class hours, 3 Laboratory hours, 4 credits In order to pass the course a passing grade must be obtained in both the LECTURE and LABORATORY components of the course. Course Purpose/Objectives: Students will learn most of the important topics of Microsoft's Expression Web software. Building and Maintaining Web Sites teaches the student to make Web pages, Publish Web pages to a Web Server, and troubleshoot any and all problems associated with their Web site design. Prerequisites: It is expected that the student will have basic knowledge in the use of a personal computer and a basic understanding of the Vista operating system. In particular the student should know how to use Window Explorer to save, copy, delete and move files. No prior knowledge or experience in the use of a database is required. Text/Materials: Text: ET-710 Building and Maintaining Web Sites Laboratory Guide Second Edition by Professor Robert M. Kueper Software: Microsoft Expression Web, Internet

Explorer 7.0 128 Bit encryption, Windows Vista. Teaching Methods: Lectures: Important material from the laboratory guide and PowerPoint presentations will be covered in class. Students should plan to take careful notes as not all material can be found in the readings. Discussion is encouraged as is student-procured, outside material relevant to topics being covered. 24 Demonstrations: Much of the class time will be used to demonstrate the practical use of the software for this course. Students are encouraged to carefully observe the demonstrations and also to go through the tutorial material in the laboratory guide. Assignments: Weekly assignments (which enforce the project based learning objectives) include web pages, summaries and answers to critical thinking questions. All written material will be graded according to Writing Intensive principals and feedback is given to the students using the "comments" feature in Blackboard. Summaries: Summaries should be a minimum of 1 page in length and contain an introductory paragraph, a paragraph containing the details of what was necessary to complete the assignment, and a conclusion. For example: John Smith April 20, 2009 Assignment Number 3 Introduction: Here you should write about the goals of this assignment. This section should be one or two sentences long. Details: Here you should write about how and what you did in order to complete the assignment. This section should be at least one paragraph and should contain the bulk of your writing. Conclusions: Here you should write your thoughts about the assignment and what conclusions you made. Page 1 Critical Thinking Questions: Question 1 Your answer 25 Question 2 Your answer Question 3 Your answer Question 4 Your answer Question 5 Your answer Page 2 Quizzes: Weekly quizzes will be given to help ensure students stay up to date with assigned material. Internet: All material will be distributed on the Internet (Blackboard). Class Syllabus, Quizzes and Final will be posted on Blackboard. Grading: Letter grades will be determined using a standard percentage point evaluation as outlined below. Grades MAY be curved after the total semester points have been tabulated Grade A AB+ B BC+ Percent 96%-100% 90%-95% 87%-89% 84%-86% 80%-83% 77%-79% 26 C CD+ D DF 74%-76% 70%-73% 67%-69% 64%-66% 60%-63% Below 60% The final grade will be computed using the following weights: Lab Assignments Weekly Quizzes Term Project Final Total 1100 Points 1200 Points 200 Points 300 Points 2800 Points 40% 43% 7% 10% 100% Course Policies: Missed Classes: The student is responsible for obtaining material, which may have been distributed on class days when he/she was absent. This can be done through contacting a classmate or logging into blackboard and download the assignments as needed. Missed or late quizzes cannot be made up under any circumstances but with good cause and adequate notice, an early quiz may be given. Any unexcused missed exam will result in a score of 0 for that exam. Assignments: All assignments are due at the beginning of class on the week following the lab assignment. Late submissions of lab assignments may be assessed a penalty of 10% per day. Lab assignments that are two weeks past due will be may be given a zero 0 for that assignment. Academic Dishonesty: Plagiarism and cheating are serious offenses and may be punished by failure on exam, paper, project, course and or expulsion from the University. For more information, refer to the &quot;Academic Dishonesty&quot; policy in the University Undergraduate Catalog. Need for Assistance: If you have any condition, such as a physical or learning disability, which will make it difficult for you to carry out the work as outlined or which will require academic accommodations, please notify the instructor as soon as possible. 27 Posting of Grades: Final grades will not be posted. If you wish to have your final grade sent to you, please bring a self-addressed, stamped envelope to the final exam. Instructor: Mike Metaxas Office Location: Technology Building Room T-20 Telephone: 718-631-6207 Email: mmetaxas@qcc.cuny.edu 28 Syllabus Subject Material Projects Project#1: Using Notepad to make a Web Page Pages 3-10, Sample quiz Project#2: Introduction to Expression Web Pages 11-15, Quiz #1 Project#3: Create and Modifying Web Pages Pages 17-21, Quiz #2 Project#4: Creating and Verifying Hyperlinks Pages 23-29, Quiz #3 Project#5: Creating and Modifying Expression Web Templates Pages 31-36, Quiz #4 Project#6: Defining Graphic on a Web Page Pages 37-44, Quiz #5 Project#7: Creating and Utilizing Tables Pages 45-50, Quiz #6 Project#8: Creating and Manipulating a Web Form Pages 51-58, Quiz #7 Project#9: Implementing Frames Pages 59-65, Quiz #8 Project#10: Introduction to Adobe Fireworks Pages 67-72, Quiz #9 Project#11: Advanced Web Page Development features Pages 73-78, Quiz #10 Project#12: Web Administration &amp; Troubleshooting Pages 79-87 Quiz #11 Term Project Pages 88-89 Review for final Quiz #12 29 30
Источник: https://studylib.net/doc/11104365/stem%C2%A0a-academ-my%C2%A0coh-horts%C2%A0r
10

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Visual Basic Programming: **more books (100)**

1. **Visual Basic Programming for the Absolute Beginner w/CD (For the Absolute Beginner (Series).)** by **Michael Vine**, 2002-07-01
2. **Programming In Visual Basic.net** by **Julia Case Bradley, A. C. Millspaugh**, 2005-05
3. **MicrosoftVisual Basic: Game Programming for Teens** by **Jonathan S. Harbour**, 2007-09-25
4. **Programming in Visual Basic 2010: The Very Beginner's Guide** by **Jim McKeown**, 2010-03-29
5. **Clearly Visual Basic: Programming with Microsoft Visual Basic 2008** by **Diane Zak**, 2008-08-15
6. **Programming Visual Basic 2008: Build .NET 3.5 Applications with Microsoft's RAD Tool for Business** by **Tim Patrick**, 2008-05-27
7. **Introduction to Programming Using Visual Basic 2010 (8th Edition)** by **David I. Schneider**, 2010-08-30
8. **Practical Database Programming with Visual Basic.NET** by **Ying Bai**, 2008-10-20
9. **Programming in Visual Basic 2010** by **Julia Case Bradley, Anita Millspaugh**, 2010-07-28
10. **Programming in Visual Basic 2008** by **Julia Case Bradley, Anita Millspaugh**, 2008-05-30
11. **Introduction to Programming Using Visual Basic 2008, An (w/VS2008 DVD) (7th Edition)** by **David I. Schneider**, 2008-07-13
12. **Visual Basic Game Programming for Teens** by **Jonathan S. Harbour**, 2004-12-21
13. **Programming Microsoft Visual Basic 6.0** by **Francesco Balena 196**, 1999-05-26
14. **Visual Basic Game Programming with DirectX (The Premier Press Game Development Series)** by **Jonathan S. Harbour**, 2002-01-02

181. Basic Books And Tutorials

The CDROM features custom applications, code, and scores of valuable visual Basicprogramming aids, ready-to-run examples of advanced VB applets, and a dozen
http://www.mathtools.net/Basic/Books_and_Tutorials/

183. EliteVB

visual basic News. MCPCentral announces NetTools TCP/IP Tool Chest; Connections.NET Tour; Dev4Dev Beta Testers Needed! devMail.Net 2.1 released.
http://www.extreme-vb.net/

184. ProgrammingMSAccess.Com
Samples, book excerpts, presentations, FAQs, tutorials on Access 2000 and the other Office 2000 component Category Computers Software Databases MS Access...... On top of that, the first new member to join the DDG in March wins a copy ofProgramming Microsoft visual basic .NET for Microsoft Access Databases.
http://www.programmingmsaccess.com/

185. DevX
Oracle Has SQL Server Beat in Stored Procedure programming Oracle9i and SQL Server2000 add enhancements to the standard SQL language that enable you to use
http://www.devx.com/

187. Vb World
Developer.com Earthweb The Web CodeGuru Gamelan Jars Wireless Discussions Architecture Design Database DHTML Java Languages Tools Microsoft .NET Open Source Project Management Techniques Voice Web Services Wireless/Mobile XML Be a
http://www.vbworld.com/

188. TechTarget, The Most Targeted IT Media
Portal site with links, advice and a search engine designed specifically to search for application Category Computers Software Portals......Visit Our Targeted IT Media
http://www.techtarget.com/

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Page 10 181-188 of 188 Back 9 3

e99.com Bookstore

Page 10 181-188 of 188 Back 1,340g

- 19 Feb 2009
- Pearson Education (US)
- Pearson
- Upper Saddle River, NJ, United States
- English
- 7th edition
- 0136060722
- 9780136060727

Back cover copy

This revision of Schneider's best-selling guide is designed for readers with no prior programming experience. It focuses on developing good problem-solving skills, building a strong foundation that will give readers a sustainable understanding of programming. KEY TOPICS: Based on Visual Basic 2008, the book starts with a brief review of the preliminaries of Windows, and then focuses on problem-solving. A broad range of real-world examples, section-ending exercises, case studies, and programming projects gives readers significant hands-on experience. A useful reference for both beginning programmers and those moving from another language and that want to learn more about programming with the latest version of Microsoft's Visual Basic.
show more

Table of contents

show more

About David I. Schneider

David I. Schneider spent over 32 years teaching at the University of Maryland, and has published an impressive 28 books over the past 18 years. He is the author of several best-selling texts, including his Visual Basic Series with Prentice Hall. Why are his books best-sellers? Students love his extensive use of examples and applications-a concept that stems from his teaching experience and results in books that are "tried and true." His best-selling Visual Basic texts have been successfully used for many years and have received consistent praise from both students and instructors.
show more

Rating details

12 ratings

3.17 out of 5 stars

5 25% (3)

4 17% (2)

3 33% (4)

2 0% (0)

1 25% (3)

Book ratings by Goodreads

Goodreads is the world's largest site for readers with over 50 million reviews. We're featuring millions of their reader ratings on our book pages to help you find your new favourite book.
Close X

{"bd_js_shop":"Shop","bd_js_too_long_for_shipping_label":"Sorry, that's too long for our shipping labels","bd_js_why_are_you_leaving":"Why are you leaving?","bd_omnibus_ph1_price_statement_suffix":"vs RRP","bd_js_too_long":"Sorry, that's too long","bd_js_could_not_find_address_try_again":"Sorry, we couldn't find the address. Please enter manually:","bd_js_keep_typing_to_refine_search_results":"Keep typing to refine the search results","bd_pre_order":"Pre-order","bd_js_top_categories":"Top Categories","bd_price_save":"Save {0}","bd_notify_me":"Notify me","bd_js_name_only_letters":"Sorry, full name can only contain letters","bd_js_show_more":"show more","bd_js_enter_valid_email_address":"Please enter a valid email address","bd_js_delivery_time":"Delivery time","bd_js_enter_address_manually":"Enter address manually","bd_js_more_categories":"More Categories","bd_30_day_low_price_carousel":"30-day lowprice","bd_js_continue_shopping":"Continue Shopping","bd_js_account_and_help":"Account & Help","bd_js_price":"Price","bd_js_basket_checkout":"Basket / Checkout","bd_add_to_basket":"Add to basket","bd_js_enter_first_last_name":"Please enter a first and last name","bd_js_please_enter_your":"Please enter your"}

Источник: https://www.bookdepository.com/Introduction-Programming-Using-Visual-Basic-2008-w-VS2008-DVD-David-I-Schneider/9780136060727?ref=grid-view&qid=1572988478363&sr=1-358 6

ISBN 13: 9781259097096 Instructor Solutions Manual Sundaram

skyusbook

unread,
Aug 12, 2018, 5:48:14 AM8/12/18
□

Reply to author

Sign in to reply to author

Forward

Sign in to forward

Delete

You do not have permission to delete messages in this group

Link

Report message as abuse

Sign in to report message as abuse

Show original message

Either email addresses are anonymous for this group or you need the view member email addresses permission to view the original message

to

[Sanjiv Das Rangarajan Sundaram] Derivatives: Principles and Practice 1st Edition McGraw Hill, 2013 Instructor Solutions Manual ISBN 13: 9781259097096

The solutions manuals to huge list of educational books are ready for use. They all have full and precise solutions. Both instructors and students can use these solutions manuals. To get the solutions manual you need, feel free to send an email with the title.

NOT free

Contact email: skyusbook"@"gmail.com (@ with no quotations marks)

To search by author's name use Ctrl + F

Finite Math and Its Application (9th Ed., Larry J Goldstein, Schneider & Martha J. Siegel)
[L. D. Field, H. L. Li, A. M. Magill] Organic Structures from 2D NMR Spectra, Kindle Edition Set 2016 ISBN: 978-1-119-07370-3
Modeling in Transport Phenomena: A Conceptual Approach (2nd Ed.,Ismail Tosun)
Batch Distillation: Simulation, Optional Design and Control (Diwekar)
Mechanics of Flight (Warren F. Phillips)
Francis Carey, Neil Allison - Solutions Manual to accompany Organic Chemistry, Seventh Edition (2007, McGraw-Hill Science_Engineering_Math)
Calculus with Applications for the Life Sciences (Raymond N. Greenwell, Nathan P. Ritchey & Margaret L. Lial)
Principles of Combustion (2nd Ed., Kenneth Kuan-yun Kuo)
[Fulay, Donald R. Askeland, Pradeep P.] The Science , Engineering of Materials 5th Edition Thomson; 5th ISBN-13: 978-0534553968
[Mark J. Schulz, Ajit D. Kelkar, Mannur J. Sundaresan] Nanoengineering of Structural Functional and Smart Materials 1st Edition CRC Press; 1st ISBN-13: 978-0849316531
[Boas, Mary L.] Mathematical Methods in the Physical Sciences 3rd Edition Wiley; 3 ISBN-13: 978-0471198260
Computer Numerical Control: Operation and Programming (3rd Ed., Stenerson & Curran)
Introductory Algebra (8th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
Essentials of Stochastic Processes (Rick Durrett)
Customer Relationship Management: The Bottom Line to Optimizing Your ROI (Anton & Petouhoff)
[Elements of Chemical Reaction Engineering by Fogler hubbard, hamman , johnson , 3rd edition
[Baker,R. Jacob] CMOS: Mixed-Signal Circuit Design, Second Edition 2nd Edition Wiley-IEEE Press; 2 ISBN-13: 978-0470290262
Elements of Statistical Mechanics: With an Introduction to Quantum Field Theory and Numerical Simulation (Ivo Sachs & Siddhartha Sen)
Prealgebra (5th Ed., Marvin L. Bittinger, David J. Ellenbogen & Barbara L. Johnson)
[Gary Chartrand, Albert D. Polimeni, Ping Zhang] Mathematical Proofs: A Transition to Advanced Mathematics 2nd Edition Pearson; 2nd 2007 ISBN-13: 978-0321390530
Process Control Instrumentation Technology (8th Ed., Curtis Johnson)
[Richard E. Baker, Valdean C. Lembke, Thomas E. King] Advanced Financial Accounting 5th Edition Mcgraw-Hill College; 5th ISBN-13: 978-0072444124
Construction Contracts (3RD Ed., Jimmie Hinze)
[Godfrey Beddard] Applying Maths in the Chemical and Biomolecular Sciences: An Example-Based Approach 1st Edition Oxford University Press; 1st ISBN-13: 978-0199230914
Applied Statistics and Probability for Engineers (4th Ed., Douglas Montgomery & George Runger)
[Jimmy T. Arnold, Lee W. Johnson, R. Dean Riess] Introduction to Linear Algebra 5th Edition Pearson; 5th 2001 ISBN-13: 978-0201658590
Introduction to Mathematical Structures and Proofs (Larry J. Gerstein)
[David R. Gaskell Introduction to the Thermodynamics of Materials 3rd Edition Taylor & Francis ISBN 13: 9781560324324
[J. David Logan Applied Mathematics 3rd Edition 2006 Wiley-Interscience ISBN-13: 978-0471746621
A Graphical Approach to Precalculus (4th Ed., John Hornsby, Lial & Rockswold)
Donald R. Askeland (auth.) - The Science and Engineering of Materials_ Solutions Manual (1991, Springer Netherlands)
[John D. Cutnell, Kenneth W. Johnson] Physics, 9th Edition John Wiley and Sons;(2012) ISBN: 978-0-470-87952-8
Design of Analog CMOS Integrated Circuits, 1st Edition, by Razavi Douglas C. Montgomery
Single Variable Calculus (3rd Ed., Monty J. Strauss, Gerald L. Bradley & Karl J. Smith)
Tony Gaddis - Starting Out with Java_ From Control Structures through Objects, Instructor Solution Manual (2012, Addison-Wesley)

[Francis A. Carey, Robert M. Giuliano] Organic Chemistry, 8th Edition 2010 McGraw-Hill ISBN-13: 978-0077354770
CMOS VLSI Design: A Circuits and Systems Perspective, 3/E., Neil Weste, David Harris
Simulation Modeling and Analysis (4th Ed., Averill Law)
Oracle 10g Programming: A Primer (Rajshekhar Sunderraman)
[Curtis Howard] Orbital Mechanics For Engineering Students 1st Edition Butterworth-Heinemann 2004 ISBN: 9780080470542
Intermediate Algebra: Graphs & Models (3rd Ed., Marvin L. Bittinger, David J. Ellenbogen & Barbara L. Johnson)
Aircraft Structures for Engineering Students (4th Ed., T.H.G. Megson)
Java Software Structures: Designing and Using Data Structures (2nd Ed., Lewis, Chase)
A Quantum Approach to Condensed Matter Physics (Philip L. Taylor & Olle Heinonen)
File Structures: An Object-Oriented Approach with C++ (3rd Ed., Folk, Zoellick, Riccardi)
[Howard Curtis] Orbital Mechanics for Engineering Students Kindle 2nd Edition Butterworth-Heinemann 2009 ISBN: 9780080887845
[Paul M. Fischer, William J. Taylor and Rita H. Cheng] Advanced Accounting 10th edition Publisher: South-Western Publishing Co.2009 ISBN13: 9780324379051
Leslie D. Field, Sev Sternhell, John R. Kalman - Organic Structures from Spectra_ Solutions Manual (2007, John Wiley and Sons Ltd)
[Rousseau, Richard M. Felder, Ronald W.] Elementary Principles of Chemical Processes 3rd Edition John Wiley & Sons; 3rd 2004 ISBN-13: 978-0471375876
Principles of Managerial Finance plus MyFinanceLab Student Access Kit, 12th Ed., Lawrence J. Gitman)
Developmental Mathematics (K. Elayn Martin-Gay)
[Edward B. Saff, Arthur David Snider] Fundamentals of Complex Analysis with Applications to Engineering, Science, and Mathematics 3rd Edition Pearson; 3rd 2003 ISBN-13: 978-0139078743
Biostatistics for the Health Sciences (R. Clifford Blair & Richard Taylor)
Concrete Structures (Mehdi Setareh & Robert M. Darvas)
Willem Conradie, Valentin Goranko, Claudette Robinson - Logic and Discrete Mathematics_ A Concise Introduction, Solutions Manual (2015, Wiley)
[Falconer, Kenneth J.] Fractal Geometry: Mathematical Foundations and Applications 2nd Edition Wiley, 2003 ISBN 13: 9780470848623
Prealgebra & Introductory Algebra (2nd Ed., K. Elayn Martin-Gay)
Data Structures and Algorithm Analysis in C (2nd Ed., Mark Allen Weiss)
T. W. Graham Solomons, Craig B. Fryhle, Robert G. Johnson - Organic Chemistry_ Study Guide and Solutions Manual (2008, Wiley)
McCabe_ Julian C Smith_ Peter Harriott - Unit operations of chemical engineering [SOLUTIONS MANUAL] (1985, McGraw-Hill) [McGraw-Hill chemical engineering series] Warren L
[Robert L. Carter] Molecular Symmetry and Group Theory Wiley ISBN-13: 978-0471149552
[Goodman D.Yates, David J.Roy] Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers 2nd Edition Wiley; 2nd 2004 ISBN-13: 978-0471272144
[David C. Lay] Linear Algebra and Its Applications,Books a la Carte edition 4th Edition Pearson;2011 ISBN-13: 978-0321385178
[Michael R. Cullen Dennis G. Zill] Advanced Engineering Mathematics: 3rd Edition Jones & Bartlett Learning, LLC 2006 ASIN: B0089AEAJK
Introduction to Engineering Experimentation (2nd Ed., Wheeler & Ganji)
[Jordan, Stephen A. Ross, Randolph Westerfield, Bradford D.] Essentials of Corporate Finance 6th Edition McGraw-Hill; 6th ASIN: B0041W2KEC
[Rosen, Kenneth H.] Elementary Number Theory 5th Edition Addison Wesley; 5th 2004 ISBN-13: 978-0321237071
[Stephen Ross, Randolph Westerfield, Jeffrey Jaffe] Corporate Finance, 8th Edition McGraw-Hill/Irwin; 8th 2006 ISBN-13: 978-0073337180
[Douglas C. Montgomery] Introduction to Statistical Quality Control 4th Edition John Wiley & Sons 2007 ISBN-13: 978-0471381853
Intermediate Algebra with Applications and Visualization (2nd Ed., Gary K. Rockswold & Terry A. Krieger)
[Sundem, Stratton, Burgstahler, Schatzberg Horngren] Introduction to Management Accounting:Pearson Education (2008) ISBN-13: 978-8120335684
C++ Programming Today, 2/E, (Barbara Johnston)
ELEMENTARY NUMBER THEORY AND ITS APPLICATIONS, (5TH EDITION, Bart Goddard, Kenneth H. Rosen)
[Mark A. Weiss Data structures and Algorithm Analysis in C 2nd Edition 1996 Pearson ISBN-13: 978-0201498400
Starting Out with Java: From Control Structures through Data Structures, by Gaddis, Muganda)
[International Trade 2nd Ed by Robert Feenstra, Alan Taylor
[Lightfoot, R. Byron Bird, Warren E. Stewart, Edwin N.] Transport Phenomena Revised 2nd Edition John Wiley & Sons, Inc.; 2nd 2006 ISBN-13: 978-0470115398
[Jerry Banks, John S. Carson, Barry L. Nelson , David M. Nicol] Discrete-Event System Simulation 3rd Edition Prentice Hall 2000 ISBN-13: 978-0130887023
Robert J. Schilling, Sandra L. Harris - Instructor Solution Manual - Fundamentals of Digital Signal Processing Using MATLAB
Elementary Statistics: Picturing the World (4TH Ed., Larson & Farber)
American Radio Relay League - The ARRL Instructor's Manual for Technician and General License Courses (2008, Amer Radio Relay League)
Principles of Highway Engineering and Traffic Analysis (3/E, Fred Mannering, Kilareski & Washburn)
[T. W. Graham Solomons, Craig B. Fryhle Organic Chemistry, 10th Edition 2010 Wiley ISBN-13: 978-0470478394
Finite Mathematics and Calculus with Applications (8th Ed.,L. Lial, Greenwell & Ritchey)
Experiments with Economic Principles, 2/E, by Bergstrom, Miller
[Johnson,Curtis D.] Process Control Instrumentation Technology 8th Edition Pearson; 8th 2005 ISBN-13: 978-0131194571
C. Henry Edwards, David E. Penney - Instructor's Solutions Manuals for Calculus Early Transcendentals 7th Edition (2007)
Design and Analysis of Experiments, 7/E, by Montgomery
[Douglas C. Montgomery] Introduction to Statistical Quality Control, 7th Edition, + Kindle Edition Wiley;(2009) ISBN-13: 978-1118146811
[Richard Baker] Advanced Financial Accounting 8th edition Richard D. Irwin, Inc. 2009 ISBN13: 9780073526911
[Thiele, Warren L. Stutzman, Gary A.] Antenna Theory and Design 2nd Edition Wiley; 2nd 1997 ISBN-13: 978-0471025900
Systems for Planning and Control in Manufacturing (D. K. Harrison & D. J. Petty)
Applied Numerical Analysis, 7th Edition, by Gerald, Wheatley
[C. T. Sun] Mechanics of Aircraft Structures 2nd Edition Kindle Wiley; 2nd 2006 ISBN-13: 978-0471699668
[Fitzgerald, A. E.Charles Kingsley, JR., Stephen D. Umans] Electric Machinery 6th Edition McGraw-Hill; 6th 2005 ISBN-13: 978-0071230100
[Adel S. Sedra, Kenneth C. Smith] Microelectronic Circuits 5TH Edition McGraw-Hill ISBN-13: 978-0195142525
Mathematics with Applications (9th Ed., Margaret L. Lial, Hungerford & John Holcomb)
Precalculus with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
[Richard C. Dorf and James A. Svoboda] Introduction to Electric Circuits 7th Edition John Wiley & Sons, Inc.7th 2006 ISBN13: 9780471730422
Prealgebra (3rd Ed., Margaret L. Lial & Diana L. Hestwood)
Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5/e, Luger
[Kenneth M. Leet Emeritus, Chia-Ming Uang, Joel Lanning] Fundamentals of Structural Analysis 3rd Edition McGraw-Hill ASIN: B0086HY1O0
[Principles of Highway Engineering and Traffic Analysis 4 ED (US Units) by Mannering,Washburn,Kilareski
[Baker, Richard E.; Lembke, Valdean C.; King, Thomas E.] Advanced Financial Accounting 5th Edition Mcgraw-Hill ASIN: B008BWV85A
Starting Out with Java 5: Control Structures to Objects (Tony Gaddis)
College Algebra and Trigonometry (4TH Ed., Margaret L. Lial, John Hornsby & David I. Schneider)
[Wesley G. Crawford] Construction Surveying and Layout 2nd Edition Creative Construction ISBN 13: 9780964742109
[Ravi Anupindi, Sunil Chopra, Sudhakar D. Deshmukh, Jan Van Mieghem] Managing Business Process Flows, Principles of Operations Management 2nd Edition Prentice Hall; 2nd 2005 ISBN-13: 978-0130675460
Heating, Ventilating and Air Conditioning Analysis and Design (6th Ed., McQuiston)
[Frederick S. Hillier, Gerald J. Lieberman] Introduction to Operations Research 7th Edition McGraw Hill ISBN-13: 978-0072461213
Calculus and its Applications (11th Ed., Larry J Goldstein, Schneider, Lay & Asmar)
[Walter T. Harrison Jr., Charles T. Horngren , Bill Thomas] Financial Accounting 8th Edition Pearson; 8th 2009 ISBN-13: 978-0136108863
PIC Microcontroller (Muhammad Ali Mazidi, Rolin McKinlay & Danny Causey)
Concrete Abstract Algebra: From Numbers to GrÃ¶bner Bases (Niels Lauritzen)
[Carey, Francis A., Sundberg, Richard J.] Advanced Organic Chemistry Part A: Structure and Mechanisms Academic Press, 1996 ISBN 978-0-387-44899-2
Calculus, Early Transcendentals (7th Ed., C. Henry Edwards & David E. Penney)
Fundamentals of Probability, with Stochastic Processes (3rd Ed., Saeed Ghahramani)
[Cengage Learning studyguide 1] Ron Larson - solutions manual _ Precalculus with limits (2014, Cengage Learning)
Algebra and Trigonometry with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
Trigonometry (9th Ed., Margaret L. Lial, John Hornsby & David I. Schneider)
Ultrasonic Nondestructive Evaluation Systems: Models and Measurements (Lester W. Schmerr & Sung-Jin Song)
[James M. Reeve, Jonathan E. Duchac Carl S. Warren] Accounting 25th Edition Latest version included Loose Leaf Cengage Learning (2014) ISBN-13: 978-1285914794
[Ken A. Dill, Sarina Bromberg] Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience, 2nd Edition Garland Science; 2nd ISBN-13: 978-0815344308
Orbital Mechanics for Engineering Students (Howard Curtis)

James Wilson - A Hungerford's Algebra Solutions Manual (2002, University of Oregon, Portland State University.)
[John R. Hubbard, Anita Huray] Data Structures with Java United States Ed Edition Prentice Hall; ISBN-13: 978-0130933744
Advanced Engineering Mathematics,3/E, by Zill, Cullen
Introductory and Intermediate Algebra (3rd Ed., argaret L. Lial, John Hornsby & Terry McGinnis)
Calculus with Applications (9th Ed., Margaret L. Lial, Greenwell & Nathan P. Ritchey)
Intermediate Algebra (5TH Ed., K. Elayn Martin-Gay)
Mechanics of Aircraft Structures (2nd Ed, C. T. Sun)
[Randolph W Westerfield Robert R. Dockson Deans Chair in Bus. Admin] Essentials of Corporate Finance, 8th Edition McGraw-Hill Education; 8th 2013 ISBN-13: 978-0078034756
[Margaret L. Lial , Raymond N. Greenwell, Nathan P. Ritchey] Calculus with Applications 8th Edition Addison-Wesley; 8th 2004 ISBN-13: 978-0321238436
Customer Service: A Practical Approach (4th Ed., Harris)
[Endre Süli, David F. Mayers] An Introduction to Numerical Analysis 1st Edition Cambridge University Press;2003 ISBN-13: 978-0521007948
Technical Mathematics (2th Ed., Dale Ewen, Joan S. Gary & James E. Trefzger)
The Cosmic Perspective, 5/E, Jeffrey O. Bennett, Megan Donahue, Nicholas Schneider, Mark Voit
[Warren McCabe, Julian Smith, Peter Harriott emeritus] Unit Operations of Chemical Engineering 7th Edition McGraw-Hill Education; 7 2004 ISBN-13: 978-0072848236
[Lawrence J. Gitman] Principles of Managerial Finance 4th Edition Pearson Education (2005) ISBN 13: 9781740911016
[Sanjiv Das Rangarajan Sundaram] Derivatives: Principles and Practice 1st Edition McGraw Hill, 2013 ISBN 13: 9781259097096
Organic Chemistry, 7/E., by Carey, Atkins
[Curtis F. Gerald, Patrick O. Wheatley] Applied Numerical Analysis 7th Edition Pearson; 7 2003 ISBN-13: 978-0321133045
Thomas E. Copeland J. Fred Weston Kuldeep Shastri - Financial Theory and Corporate Policy FOURTH EDITION SOLUTIONS MANUAL
[Paul D. Kimmel, Jerry J. Weygandt , Donald E. Kieso] Accounting Principles 9th Edition for SouthWestern Illinois College-Belleville 9th Edition Wiley; 9th 2009 ISBN-13: 978-0470539118
Developmental Mathematics: Basic Mathematics and Algebra (Margaret L. Lial, John Hornsby, Terry McGinnis, Stanley A. Salzman & Diana L. Hestwood)
[David R. Anderson, Dennis J. Sweeney, Thomas A. Williams, Jeffrey D. Camm, R. Kipp Martin] An Introduction to Management Science: Quantitative Approaches to Decision Making, Revised 13th Edition South-Western; 13th 2011 ISBN-13: 978-1111532222
Elementary Math Modeling Updated (2nd Ed., Mary Ellen Davis & C. Henry Edwards)
[Kenneth S. Krane] Modern Physics, 3rd Edition, + Kindle Edition Wiley; (2012) ISBN-13: 978-1118061145
Calculus, An Applied Approach, 8/E, by Larson
[Ralph H. Petrucci, William S. Harwood, Geoffrey Herring] General Chemistry: Principles and Modern Applications 8th Edition Prentice Hall; 8th 2001 ISBN-13: 978-0130143297
Marilyn E. Noz, Gerald Q. Maguire - Radiation Protection in the Health Sciences_ With Problem Solutions Manual (2007)
A Graphical Approach to Algebra and Trigonometry (4th ed., John Hornsby, Margaret L. Lial & Gary K. Rockswold)
[Rotman, Joseph J] Journey Into Mathematics: An Introduction to Proofs Dover Publications Inc. 2006 ISBN-13: 978-0486453064
[Clough, R.W.] Dynamics of Structures International 2nd Revised Edition, 1993 ISBN-13: 978-0071132411
An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation (Desmond Higham)
[Douglas B. West] Introduction to Graph Theory 2nd Edition Pearson; 2nd ISBN-13: 978-0130144003
[Behzad Razavi] Fundamentals of Microelectronics 2nd Edition Wiley; 2nd 2013 ISBN-13: 978-1118156322
[Keown Arthur J., John H. Martin, John W. Petty, David F. Scott] Financial Management: Principles and Applications 10th Edition Prentice Hall; 10 2004 ISBN-13: 978-0131450653
Oracle 9i Programming: A Primer (Rajshekhar Sunderraman)
[Arthur Nilson , David Darwin , Charles Dolan] Design of Concrete Structures 14th Edition McGraw-Hill Education; 14th ISBN-13: 978-0073293493
[T W Graham Solomons; Craig B Fryhle; S A Snyder; Robert G Johnson; Jon Antilla] Organic chemistry, 11th Edition Wiley, [2014] ISBN: 9781118147900
Multivariate Data Analysis (6th Ed., Hair, Black, Babin, Anderson & Tatham)
[Randall D. Knight, Brian Jones, Stuart Field] College Physics: A Strategic Approach Volume 1 2nd Edition Addison-Wesley;(2009) ISBN-13: 978-0321611147
[Samuel Karlin, Howard M. Taylor] An Introduction to Stochastic Modeling, 3rd Edition Academic Press; 3rd 1998 ISBN-13: 978-0126848878
[Welty, James, Wicks, Charles E., Wilson, Robert E., Rorrer] Fundamentals of Momentum, Heat, and Mass Transfer 4th edition Wiley; 4th 2000 ISBN-13: 978-0471381495
Understanding UNIX/LINUX Programming: A Guide to Theory and Practice(Bruce Molay)
Introduction to Statistical Quality Control (6th Ed., Douglas C. Montgomery)
The 8051 Microcontroller (4th Ed., I. Scott MacKenzie, Raphael Chung-Wei Phan)
[Charles H. Gibson] Financial Reporting and Analysis - Using Financial Accounting Information 10th Edition Thomson/SouthWestern; 10th 2006 ASIN: B0058SMY1M
Modern Electronic Communication (9th Ed., Jeff Beasley, Gary M.Miller)
[Kenneth Kuttler] Calculus Multivariable: Theory and Applications World Scientific Pub Co Inc 2011 ISBN-13: 978-9814335713
Jerry D. Wilson, Cecilia A. Hernández - Instructor's Manual to Physics Laboratory Experiments (2014, Cengage Learning)
[Lawrence E. Spence, Stephen H. Friedberg, Arnold J. Insel] Linear Algebra, 4th Edition Fourth Pearson; 4th 2002 ISBN-13: 978-0130084514
Advanced Engineering Mathematics (2nd Ed, Michael Greenberg)
Essentials of College Algebra with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
Beginning and Intermediate Algebra (4th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)
[Faye C. McQuiston, Jerald D. Parker, Jeffrey D. Spitler] Heating, Ventilating and Air Conditioning Analysis and Design 6th Edition Wiley; 6th 2004 ISBN-13: 978-0471470151
[James K. Wight; James G. MacGregor] Reinforced Concrete: Mechanics and Design 5th Edition Prentice Hall, 2008 ISBN 13: 9780132281416
[Donald R. Askeland, Pradeep P. Fulay] The Science & Engineering of Materials, 5th Edition Thomson; 5th 2005 ISBN-13: 978-0534553968
An Introduction to Numerical Analysis (Endre Suli and David Mayers)
[C. Henry Edwards, David E. Penney Calculus, Early Transcendentals 6th Edition Kindle Prentice Hall;2002 ISBN-13: 978-0130084071
Essentials of College Algebra, Alternate Edition (Margaret L. Lial, Hornsby & Schneider)
[Raphael C.-W. Phan I. Scott Mackenzie] The 8051 Microcontroller 4th International Edition Pearson ASIN: B001KR5G5G
Saber Elaydi - Instructor's Solutions Manual to An Introduction to Difference Equations (2005, Springer)
Fundamentals of Complex Analysis with Applications to Engineering, Science, and Mathematics (3rd Ed., E. Saff & Arthur Snider)
[Michael Spivak] Calculus On Manifolds: A Modern Approach To Classical Theorems Of Advanced Calculus Westview Press; ISBN-13: 978-0805390216
Elementary Linear Algebra (2nd Ed., Spence, Insel & Friedberg)
[Tony Chan Carusone, David Johns, Kenneth Martin] Analog Integrated Circuit Design 2nd Edition Wiley; 2 2011 ISBN-13: 978-0470770108
Data Structures and Problem Solving Using C++ (2nd Ed., Mark Allen Weiss)
[Woolridge, Jeffrey M.] Introductory Econometrics: A Modern Approach 3rd Edition Thomson South-Western; ISBN-13: 978-7302143123
Algebra A Combined Approach (3rd Ed., K. Elayn Martin-Gay)
Construction Contracts (2nd Ed., Hinze)*
Discrete Mathematics (7th Ed., Richard Johnsonbaugh)
Stephen A. Ross, Randolph W. Westerfield, Jeffrey Jaffe - Instructor's Manual to accompany Corporate Finance, Sixth Edition (2001, McGraw-Hill Inc.,US)
[Kenneth W. Whitten, Raymond E. Davis, Larry Peck, George G. Stanley] Chemistry 10th Edition Cengage Learning 2013 ISBN-13: 978-1133610663
Francis A. Carey, Richard J. Sundberg - Solutions Manual for Advanced Organic Chemistry Part B_ Reactions and Synthesis (2007)
Probability and Statistics in Engineering (4th Ed., Hines, Montgomery, Goldsman & Borror)
[Pru Marriott, J R Edwards , Howard J Mellett] Introduction to Accounting 3rd Edition, Kindle Edition ISBN-13: 978-0761970378
Prealgebra and Introductory Algebra (2nd Ed., Margaret L. Lial, John Hornsby, Terry McGinnis & Diana L. Hestwood)
[Robert J. Schilling , Sandra L Harris, Fundamentals of Digital Signal Processing Using MATLAB 2nd Edition 2011 Cengage Learning ISBN-13: 978-0840069092
[R. Jacob Baker] CMOS Circuit Design, Layout, and Simulation, Second Edition 2nd Edition Wiley-IEEE Press; 2 ISBN-13: 978-0471700555
[Margaret L. Lial, Raymond N. Greenwell, Nathan P. Ritchey] Finite Mathematics 8th Edition Addison Wesley; 8th 2004 ISBN-13: 978-0321228260
[Bradford D. Jordan , Stephen A. Ross, Randolph Westerfield] Fundamentals of Corporate Finance, 8th Edition, Irwin/McGraw-Hill; 2008 ISBN-13: 978-0073282121
[Lawrence J. Gitman . Chad J.Zutter] Principles of Managerial Finance: Global Edition Pearson Education; 2011 ISBN-13: 978-0273754282
Tony Gaddis, Diane Christie - Starting Out with Java_ From Control Structures through Objects, Lab Manual Solutions (2012, Addison-Wesley)
[Ron Larson, Bruce H. Edwards] Calculus 10th Edition Kindle Cengage Learning; 10th ISBN-13: 978-1285057095
Electric Machinery (6th Ed., Fitzgerald)
Supply Chain Management (3rd Ed., Sunil Chopra & Peter Meindl)
Data Structures and Other Objects Using Java (3rd Ed., Michael Main)
[Athanasios Papoulis, S. Unnikrishna Pillai] Probability, Random Variables and Stochastic Processes 4th Edition McGraw-Hill Europe; 4th ISBN-13: 978-0071226615
Fundamentals of Investing plus MyFinanceLab Student Access Kit and OTIS Student Access Kit (10th Ed., Lawrence J. Gitman & Michael D.Joehnk)
Introduction to Programming Using Visual Basic 2008 (David I. Schneider), An, 7/E
[Kenneth Kuttler] Elementary Linear Algebra 2014 ISBN: 978-87-403-1425-0
Finite Math with Applications (9th Ed., Margaret L. Lial, Hungerford & John Holcomb)
Modeling and Simulation of Dynamic Systems (Woods, Lawrence)

[Montgomery, Douglas C.] Design and Analysis of Experiments 6th Edition John Wiley and Sons ISBN-13: 978-0470088104

Java Foundations: Introduction to Program Design and Data Structures, Lewis, DePasquale, Chase

Data Structures and Problem Solving Using Java (3rd Ed., Mark Allen Weiss)

Daniel C. Harris - Quantitative Chemical Analysis, Solutions Manual (2010, W.H. Freeman & Company)

Fluency with Information Technology, Brief Edition (Lawrence Snyder)

Econometrics of Financial Markets, by Adamek, Cambell, Lo, MacKinlay, Viceira

[T.H.G. Megson] Aircraft Structures for Engineering Students 4th Edition (Elsevier Aerospace Engineering) Butterworth-Heinemann; 4th ISBN-13: 978-0750667395

[Michael D. Greenberg] Advanced Engineering Mathematics 2nd Edition Dorling Kindersley Pvt Ltd;(2006) ISBN-13: 978-8177585469

[Theodore Bergstrom, John Mille] Experiments with Economic Principles: Microeconomics 2nd Edition McGraw-Hill/Irwin; 2nd 1999 ISBN-13: 978-0072295184

Walter Gautschi - Instructor's Solutions Manual to Numerical Analysis (2012, Springer)

Operating Systems (3rd Ed., Gary Nutt)

An Introduction to Statistical Signal Processing (Robert M. Gray)

[Daniel C. Harris] Quantitative Chemical Analysis 8th Edition 2010 W.H. Freeman ISBN-13: 978-1429264846

Introduction to Programming Using Visual Basic 2005 (6th Ed., David I.Schneider)

Prealgebra (5th Ed., K. Elayn Martin-Gay)

Applied Multivariate Statistical Analysis (6th Ed., Johnson & Wichern)

Introduction to Management Science with Student CD (9th Ed., Bernard W. Taylor)

Modern Welding Technology (6th Ed., Cary & Helzer)

[Sung-Mo Kang, Yusuf Leblebici] CMOS Digital Integrated Circuits Analysis & Design 3rd Edition McGraw-Hill Education; 2002 ISBN-13: 978-0072460537

[Daryl L. Logan] A First Course in the Finite Element Method 5th Edition CL Engineering;2011 ISBN-13: 978-0495668251

Data Structures in Java: From Abstract Data Types to the Java Collections Framework, by Gray

Heating and Cooling of Buildings: Design for Efficiency (2nd Ed., Jan F. Kreider, Peter S. Curtiss & Ari Rabl)

[Nicholas J. Garber, Lester A. Hoel] Traffic & Highway Engineering, 3rd Edition Brooks / Cole Thomson Learning 2002 ISBN 13: 9780534387433

[Ross Westerfield Jordan] Essentials of Corporate Finance 7th Edition McGraw-Hill; 7th ISBN-13: 978-0077550073

Pointers on C (Kenneth Reek)

Fundamentals of Database Systems/Oracle 9i Programming (5th Ed., Ramez Elmasri, Shamkant B. Navathe & Rajshekhar Sunderraman)

[South-Western Federal Taxation 2012 - Corporations, Partnerships, Estates and Trusts, 35th Ed by Hoffman, Maloney

[Richard L. Burden, J. Douglas Faires] Numerical Analysis 8th Edition Brooks Cole; 8th ISBN-13: 978-0534392000

Intermediate Algebra (10th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)

Data Structures and Algorithm Analysis in Java (2nd Ed., Mark Allen Weiss)

L. D. Field, H. L. Li, A. M. Magill - Instructors Guide and Solutions Manual to Organic Structures from 2D NMR Spectra (2015, Wiley)

Introductory Statistics for Engineering Experimentation (Peter Nelson, Copeland & Coffin)

Discrete Mathematics (5th ed., John Dossey, Albert Otto, Spence & Charles Vanden Eynden)

The Finite Element Method: Its Basis and Fundamentals (6th Ed., Zienkiewicz, Taylor Zhu)

[Robert C. Feenstra, Alan M. Taylor] International Trade 2nd Edition Worth Publishers; 2010 ISBN-13: 978-1429241045

[Birger Bergersen, Michael Plischke, Birger Bergersen] Equilibrium Statistical Physics 2nd Edition World Scientific Publishing Company ISBN-13:9789810216429

Leslie D. Field, Sev Sternhell, John R. Kalman - Organic Structures from Spectra_ Solutions Manual (2014, John Wiley and Sons Ltd)

[Douglas C. Montgomery] Introduction to Statistical Quality Control 5th Edition Wiley; 5th ISBN-13: 978-0471656319

Professionalism: Real Skills for Workplace Success (Anderson & Bolt)

[Francis A. Carey, Richard J. Sundberg] Advanced Organic Chemistry, Part A: Structure and Mechanisms 5th Edition Springer; 5th 2008 ISBN-13: 978-0387683461

Transport Phenomena in Multiphase Systems (Faghri & Zhang)

Probability Random Variables, and Stochastic Processes, 4th Ed., by Papoulis, Pillai

Francis A. Carey, Richard J. Sundberg - Solutions Manual for Advanced Organic Chemistry Part A_ Structure and Mechanisms (2007)

[Fred L. Mannering, Scott S. Washburn, Walter P. Kilareski] Principles of Highway Engineering and Traffic Analysis 4th Edition Wiley;4th ISBN-13: 978-8126531653

[Kassimali, Aslam] Structural Analysis 3rd Edition CL-Engineering 2004 ISBN-13:9780534391683

Essentials of Geometry for College Students (2nd Ed., Margaret L. Lial, Barbara A. Brown, Arnold R. Steffenson & L. Murphy Johnson)

[Roy D. Yates, David J. Goodman] Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers 2nd Edition Wiley; 2nd 2004 ISBN-13: 978-0471272144

John West-Burnham, Ingrid Bradbury - Performance Management Manual Pack_ Creating a Culture for Sustainable High Performance (Schools Management Solutions)

Introductory Quantum Optics (Christopher Gerry & Peter Knight)

[Warren Nagourney] Quantum Electronics for Atomic Physics Oxford University Press ISBN-13 9780199665488

Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers,2/E, by Roy D. Yates , David J. Goodman

Engineering Statistics (4th Ed., Douglas Montgomery, George Runger & Norma Faris Hubele)

[T.H.G. Megson] Structural and Stress Analysis 2nd Edition Butterworth-Heinemann 2005 ISBN: 9780080455341

Dynamics of Structures (3rd Ed., Chopra)

Electronic Devices - Conventional Current Version (8th Ed., Thomas L.Floyd)

Technical Calculus (5th Ed., Dale Ewen, Joan S. Gary & James E. Trefzger)

[Warren Nagourney] Quantum Electronics for Atomic Physics Oxford University Press ISBN-13: 978-0199532629

Introductory Circuit Analysis (11th Ed., Robert L. Boylestad)

Reinforced Concrete Design (7th Ed., Chu-Kia Wang, Charles G. Salmon & JosÃ© A. Pincheira)

Structures (6th Ed., Daniel Lewis Schodek & Martin Bechthold)

[Lay, David C.] Linear Algebra and Its Applications, 3rd Updated 3rd Edition Addison Wesley; 2005 ISBN-13: 978-0321287137

[Charles T. Horngren, Walter T. Harrison Jr., M. Suzanne Oliver] Accounting, 8th Edition Prentice Hall; 8th 2008 ISBN-13: 978-0136072973

[Steven R. Lay] Analysis: With an Introduction to Proof (4th Edition) 4th Edition Pearson; 4th 2004 ISBN-13: 978-0131481015

Introduction to Finance (Lawrence J. Gitman & Jeff Madura)

[Rosen Kenneth H] Discrete Mathematics and Its Applications Seventh Edition 7th Edition Kindle Canada McGraw-Hill Education, 7th 2011 ISBN 13: 9780073383095

Douglas A. Skoog, Donald M. West, F. James Holler, Stanley R. Crouch - Instructor's Solutions Manual to Fundamentals of Analytical Chemistry (2013, Cengage Learning)

OSP: An Environment for Operating System Projects (Michael Kifer & Scott A. Smolka)

Construction Mathematics (Surinder Virdi & Roy Baker)

Reinforced Concrete Design (6th Ed., George F. Limbrunner & Abi Aghayere)

Basic College Mathematics (7th Ed., Margaret L. Lial, Stanley A. Salzman & Diana L. Hestwood)

[Gibson, Charles H.] Financial Reporting Analysis : Using Financial Accounting Information 10th Edition South-Western Publishing Co. 2007 ISBN13: 9780324304459

Basic College Mathematics with Early Integers (K. Elayn Martin-Gay)

[Logan, Daryl L.] A First Course In The Finite Element Method 4th Edition Nick Book House (Fresno, CA, U.S.A.) ISBN 13: 9780534552985

[Ghahramani, Saeed] Fundamentals of Probability, with Stochastic Processes 3rd Edition Pearson; 3rd 2004 Chapman and Hall/CRC 2015 ISBN-13: 978-0131453401

Design and Analysis of Experiments (7th Ed., Douglas Montgomery)

Physics,7/E, by John Cutnell

Calculus (3rd Ed., Monty J. Strauss, Gerald L. Bradley & Karl J. Smith)

Elementary Structures for Architects and Builders (5th Ed., Ronald E. Shaeffer)

Linear Algebra with Applications 3rd Ed by Otto Bretscher

[Steven R. Lay] Analysis: With an Introduction to Proof 4th Edition Pearson; 4th 2004 ISBN-13: 978-0131481015

[Jr. Edwards C.H., David E. Penney] Calculus With Analytic Geometry 4th Edition Prentice Hall College Div; 4th ISBN-13: 978-0134579122

Elementary and Intermediate Algebra: Graphs & Models (3RD Ed., Marvin L. Bittinger, David J. Ellenbogen & Barbara L. Johnson)

[David R. Anderson , Dennis J. Sweeney, Thomas A. Williams , R. Kipp Martin] An Introduction to Management Science: A Quantitative Approach to Decision Making (Book Only) 12th Edition South-Western College 2007 ISBN-13: 978-0324399790

Process Control Instrumentation Technology (8th Ed., Johnson)

[Clifford A. Shaffer] Practical Introduction to Data Structures and Algorithm Analysis 2nd Edition Prentice Hall ISBN 13: 9780130284464

[John D. Cutnell, Kenneth W. Johnson] Physics 8th Edition Wiley; 8th 2009 ISBN-13: 978-0470223550

Discrete Mathematics (Sherwood Washburn, Thomas Marlowe & Charles T.Ryan)

[James F. Sepe, Mark W. Nelson J. David Spiceland] Intermediate Accounting 2013 McGraw-Hill ISBN-13: 978-1259122064

[John M. Prausnitz, Rudiger N. Lichtenthaler, Edmundo Gomes de Azevedo Molecular Thermodynamics of Fluid-Phase Equilibria 3rd Edition Prentice Hall 3rd edition 1998 ISBN-13: 978-0139777455

Douglas A. Skoog, Donald M. West, F. James Holler - Solutions Manual for Skoog _ West Fundamentals of Analytical Chemistry, 9th Edition (2013, Brooks Cole)

Hazardous Waste Management (2nd Ed., Michael D. LaGrega, Phillip L Buckingham & Jeffrey C Evans)*

[Walter Nicholson Microeconomic Theory: Basic Principles and Extensions 9th Edition South-Western College Pub; 9th 2004 ISBN-13: 978-0324270860

Probability Random Variables, and Stochastic Processes, 4th Edition by Papoulis, Pillai

Analysis and Design of Analog Integrated Circuits, 4th Ed., by Gray,Hurst, Lewis, Meyer

Introduction to Linear Algebra (5th ed., Lee W. Johnson, R. Dean Riess & Jimmy T. Arnold)

[Kassimali Aslam] Matrix Analysis of Structures 2nd Edition Kindle Cengage Learning; 2nd ISBN-13: 978-1111426200

[Dilip Kondepudi, Ilya Prigogine] Modern Thermodynamics: From Heat Engines to Dissipative Structures 1st Edition John Wiley & Sons; 1st ISBN-13: 978-0471973942

[Behzad Razavi] Design of Analog CMOS Integrated Circuits 1st Edition McGraw-Hill Education; 1st 2000 ISBN-13: 978-0072380323

Electronic and Optoelectronic Properties of Semiconductor Structures (Jasprit Singh)

Linear Algebra and Its Applications with CD-ROM, Update, 3/E, David C. Lay

Introductory Chemistry, 3/E, Nivaldo J. Tro

Calculus (6th Ed., Henry Edwards & David E. Penney)

[Douglas A. Skoog, F. James Holler, Stanley R. Crouch] Principles of Instrumental Analysis 6th Edition Brooks Cole 2006 ISBN-13: 978-0495012016

College Physics (6th Ed., Jerry D Wilson, Anthony J Buffa & Bo Lou)

Technical Mathematics with Calculus (2th Ed., Dale Ewen, Joan S. Gary James E. Trefzger)

[Peter Smith, Peter Watts Jones] Stochastic Processes: An Introduction Chapman and Hall ISBN-13: 978-1420099607

[Otto Bretscher Linear Algebra with Applications, 4th Edition US Pearson;(2008) ISBN-13: 978-0136009269

[Douglas C. Montgomery] Design and Analysis of Experiments 8th Edition, Kindle Wiley;(2012) ISBN-13: 978-1118146927

[Folland, Gerald B.] Advanced Calculus 1st Edition Pearson; 1st 2001 ISBN-13: 978-0130652652

[David R. Anderson, Dennis J. Sweeney, Thomas A. Williams] Statistics for Business and Economics, 9th Edition South-Western College Pub; 9th 2005 ISBN-13: 978-0324380255

Simulation Modeling and Analysis with Expertfit Software (4th Ed.,Averill Law)

A Graphical Approach to Precalculus with Limits: A Unit Circle Approach (4th Ed., John Hornsby, Margaret L. Lial & Gary K. Rockswold)

Analysis and Performance of Fiber Composites (3rd Ed., Bhagwan Agarwal, Lawrence Broutman & K. Chandrashekhara)

[Maurice Obstfeld, Kenneth Rogoff] Foundations of International Macroeconomics (The MIT Press) 1996 ISBN-13: 978-0262150477

Physics for Scientist and Engineers, 2/E by Knight

[Eugene F. Brigham, Michael C. Ehrhardt] Financial Management: Theory & Practice 12th Edition South-Western Publishing Co. ISBN13: 9780324422696

Starting Out with Java: From Control Structures through Objects (3rd Ed., Tony Gaddis)

[Gregory L. Baker, James A. Blackburn] The Pendulum: A Case Study in Physics 1st Edition Oxford University Press 2005 ISBN-13: 978-0198567547

[Johnson, Curtis D.] Process Control Instrumentation Technology 8th Edition Pearson; 8th 2005 ISBN-13: 978-0131194571

Digital Fundamentals (10th Ed., Thomas L. Floyd)

A Graphical Approach to College Algebra (4th Ed., John Hornsby, Margaret L. Lial & Gary K. Rockswold)

[Chip Design for Submicron VLSI: CMOS Layout and Simulation, 1st Edition Cengage Learning; 1st ISBN13: 978-0-534-46629-9

[Margaret L. Lial, Raymond N. Greenwell, Nathan P. Ritchey Finite Mathematics 8th Edition Addison Wesley;(2004) ISBN-13: 978-0321228260

[Phillip E. Allen, Douglas R. Holberg] CMOS Analog Circuit Design 2nd Edition Oxford University Press, Incorporated ISBN 13: 9780195107203

[Curtis Howard D.] Fundamentals of Aircraft Structural Analysis 1st Edition Richard d Irwin 1996 ISBN-13: 978-0256192605

The 8051 Microcontroller and Embedded Systems (2nd Ed., Mazidi, Mazidi & McKinlay)

Statistical Thermodynamics and Microscale Thermophysics (Van P. Carey)

[Douglas C. Montgomery, George C. Runger] Applied Statistics and Probability for Engineers Kindle Edition + 6th Edition Wiley;(2013) ISBN-13: 978-1118539712

Principles of Money, Banking, and Financial Markets (11th Ed.,Lawrence S. Ritter, William L. Silber & Gregory F. Udell)

Ross, Westerfield, Jaffe - Solutions Manual Corporate Finance (0)

fracture mechanics ; fundamentals and applications, 3/E, by T.L. Anderson

Theory of Applied Robotics: Kinematics, Dynamics and Control (Reza N. Jazar)

Mathematics for Elementary School Teachers (4th Ed., Phares O'Daffer, Randall Charles, Thomas Cooney, John A. Dossey & Jane Schielack)

[David C. Lay] Linear Algebra and Its Applications 3rd Edition Addison Wesley; 3rd ISBN-13: 978-0201709704

[with Student Solutions Manual CD-ROM] Gary Haggard, John Schlipf, Sue Whitesides - Discrete Mathematics for Computer Science (2006, Brooks)

[William W. Hines, Douglas C. Montgomery, David M. Goldsman, Connie M. Borror] Probability and Statistics in Engineering 4th Edition Wiley; 4th 2003 ISBN-13: 978-0471240877

Financial Accounting, 7/E, by Harrison

[Bradford D. Jordan, Thomas Miller] Fundamentals of Investments 5th Edition McGraw-Hill College; 5th ISBN-13: 978-0073382357

[Randall D. Knight (Professor Emeritus)] Physics for Scientists and Engineers: A Strategic Approach with Modern Physics and MasteringPhysics 2nd Edition Benjamin Cummings; 2nd 2007 ISBN-13: 978-0321513335

[Otto Bretscher] Linear Algebra with Applications 3rd Edition Prentice Hall; 3rd 2004 ISBN-13: 978-0131453340

Ann G. Ryan, Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining - Solutions Manual to Accompany Introduction to Linear Regression Analysis (2013, Wiley)

Beginning Algebra with Applications & Visualization, 2/E, Gary K. Rockswold, Terry A. Krieger

[Paul M. Fischer, Rita H. Cheng , William J. Tayler] Advanced Accounting (8th edition) Kindle USASouth-Western College Pub ISBN 13: 9780324058789

The Essential Cosmic Perspective Media Update, 4/E, Jeffrey O. Bennett, Megan Donahue, Nicholas Schneider, Mark Voit

[Ray W Clough, Joseph Penzien] Dynamics of Structures Computers and Structures inc; 2nd Revised edition (2010) ISBN-13: 978-0923907518

[Douglas C. Montgomery] Introduction to Linear Regression Analysis, Set 5th Edition Wiley;2013 ISBN-13: 978-1118780572

[Carl S. Warren, James M. Reeve, Jonathan Duchac] Accounting, 23th Edition South-Western College Pub (2009) ISBN 13: 9780324664287

[Neil Weste, David Harris] CMOS VLSI Design: Circuits and Systems Perspective 3rd Edition ddison-Wesley Longman, Inc.2005 ISBN13: 9780321149015

Alexander McFarlane Mood, Franklin A. Graybill, Duane C. Boes - Introduction to the Theory of Statistics_ Solutions Manual (1974)

Applied Physics (9th Ed., Dale Ewen, Ronald Nelson, Neill Schurter & Erik Gundersen)

[Frederick S. Hillier, Gerald J. Lieberman] Introduction to Operations Research with Student Access Card 9th Edition McGraw-Hill Science/Engineering/Math;(2009) ISBN-13: 978-0077298340

Algebra for College Students (6th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)

Data Structures and Algorithm Analysis in C++ (3rd Ed., Mark Allen Weiss)

Mathematical Proofs: A Transition to Advanced Mathematics (Gary Chartrand, Polimeni & Zhang)

[Bradford D. Jordan, Thomas Miller] Fundamentals of Investments 6th Edition 2011 McGraw-Hill College ISBN-13: 978-0073530710

[Adel S. Sedra, Kenneth C. Smith] Microelectronic Circuits 5TH Edition McGraw-Hill (2004) ISBN-13: 978-0195142525

Kinematics, Dynamics, and Design of Machinery, 2nd Ed., Waldron & Kinzel

[Douglas C. Montgomery, George C. Runger, Norma F. Hubele] Engineering Statistics 4th edition John Wiley & Sons, Inc.2007 ISBN13: 9780471735571

[John A. Dossey, Albert D. Otto , Lawrence E. Spence] Discrete Mathematics 5th Edition Pearson;(2005) ISBN-13: 978-0321305152

Business Mathematics, 11th Ed. (Charles D. Miller, Stanley A. Salzman Gary Clendenen)

[Stuart Warren, Jonathan Clayden, Nick Greeves] Organic Chemistry 2nd Edition Kindle Canada Oxford University Press; 2nd 2012 ISBN-13: 978-0199270293

Nelson D.L., Cox M.M. - Solutions Manual for Lehninger Principles of Biochemistry 5 ed. (Freeman, 2008)

Elementary Principles of Chemical Processes, 3rd Ed., by Felder,Rousseau

Probability and Statistics for Engineers (7th Ed., Johnson, Miller, Freund)

Essentials of College Algebra (Margaret L. Lial, John Hornsby & David Schneider)

[Dilip Kondepudi, Ilya Prigogine] Modern Thermodynamics: From Heat Engines to Dissipative Structures 1st Edition John Wiley & Sons; 1st 1998 ISBN-13: 978-0471973942

[C. Henry Edwards, David E. Penney] Calculus, Early Transcendentals 7th Edition Pearson; 7th ISBN-13: 978-0131569898

The Science and Engineering of Materials, 5/E, by Donald R.Askeland, Pradeep P. Phule

[Lawrence Revsine; Daniel W. Collins; W. Bruce Johnson] Financial Reporting and Analysis 3rd Edition Pearson ISBN-13:9780131430211

[Bauer, Wolfgang, Westfall, Gary] University Physics with Modern Physics 2nd Edition McGraw-Hill Education; (2013) ISBN-13: 978-0073513881

[Carl D. Meyer] Matrix Analysis and Applied Linear Algebra 1st Edition SIAM: ISBN-13: 978-0898714548

System Dynamics: Modeling and Simulation of Mechatronic Systems (4th Ed., Karnopp, Rosenberg)

Kenneth W. Whitten, Raymond E. Davis, M. Larry Peck, George G. Stanley, Wendy Keeney-Kennicutt - Solutions Manual for Whitten's Chemistry 10th (2013, Cengage Learning)

Applied Multivariate Statistical Analysis (6th Ed., Johnson & Wichern)

Beginning Algebra (10th Ed., Margaret L. Lial, John Hornsby & Terry McGinnis)

Beginning and Intermediate Algebra with Applications & Visualization, 2/E, Gary K. Rockswold,Terry A. Kriege

[David Halliday, Kenneth S. Krane Robert Resnick] Physics, Volume 1, 5th Edition, Kindle Edition Wiley; 5th ISBN-13: 978-0471320579

Aircraft Control and Simulation (2nd Ed., Brian Stevens & Frank Lewis)

[Arthur Nilson, David Darwin , Charles Dolan] Design of Concrete Structures 13th Edition McGraw-Hill (2003) ISBN-13: 978-0070581999

Brief Calculus and Its Applications (11/e, Larry J Goldstein, Schneider, Lay & Asmar)

[Ron Larson, Robert P. Hostetler, Bruce H. Edwards] Calculus 8th Edition Brooks Cole; 8th 2005 ISBN-13: 978-0618502981

Engineering Materials Vol. 2 : An Introduction to Microstructures, Processing and Design (3rd Ed., Michael Ashby & David R H Jones)
Linear Algebra, 4th Ed, by Stephen H. Friedberg , Arnold J. Insel , Lawrence E. Spence
[N. Felder M. Felder Kenny, Gary] Mathematical Methods in Engineering and Physics 1st Edition Kindle Canada Wiley; 1st 2015 ISBN-13: 978-1118449608
Statics: Analysis and Design of Systems in Equilibrium (Sheppard & Tongue)
[Margaret L. Lial, Raymond N. Greenwell, Nathan P. Ritchey] Books a la Carte Edition for Calculus with Applications 10th Edition Pearson;(2011) ISBN-13: 978-0321757876
Kinematics and Dynamics of Machinery (3rd Ed., Wilson & Sadler)
[Kenneth H. Rosen] Elementary Number Theory and Its Application, 6th Edition Pearson; 6th 2010 ISBN-13: 978-0321500311
General, Organic and Biological Chemistry: Structures of Life with , 2nd ed. (Karen C. Timberlake)
[John Wilson, Tim Hunt] Molecular Biology of the Cell, 5th Edition Routledge (2008) ISBN 13: 9780815341109
Flight Performance of Fixed and Rotary Wing Aircraft (Antonio Filippone)
[Knight (Professor Emeritus), Randall D.] Physics for Scientists and Engineers: A Strategic Approach with Modern Physics (3rd Edition) Pearson, 2012 ISBN 13: 9780321740908
[Razavi, Behzad] Fundamentals of Microelectronics 1st Edition Wiley; 1st 2008 ISBN-13: 978-0471478461
[Ron Larson, Bruce H. Edwards] Calculus: Early Transcendental Functions (MindTap) 6th Edition Cengage Learning;(2014) ISBN-13: 978-1285774770
Kernel Projects for Linux (Gary Nutt)
Introduction to Graph Theory, 2nd Ed, by Douglas B. West
[David Halliday, Robert Resnick, Kenneth S. Krane] Physics, Volume 2 5th Edition John Wiley & Sons; 5th ISBN-13: 978-0471401940
[Shreve, Steven] Stochastic Calculus for Finance The Binomial Asset Pricing Model Springer; 2005 ISBN-13: 978-0387249681
[Bergersen, Birger Bergersen, Michael Plischke, Birger] Equilibrium Statistical Physics 2nd Edition World Scientific Publishing Edition 2 ISBN-13:9789810216429
Starting Out with C++: From Control Structures through Objects (5th Ed., Tony Gaddis)
Managing Business Process Flows: Principles of Operations Management(2nd Ed., Anupind, Chopra, Deshmukh, et al)
Data Structures in Java (Thomas A. Standish)
Theory of Applied Robotics: Kinematics, Dynamics and Control (Reza N.Jazar)
Principles of Managerial Finance Brief plus MyFinanceLab Student Access Kit (5th Ed., Lawrence J. Gitman)
[H. Scott Fogler] Elements of Chemical Reaction Engineering 4th Edition Prentice Hall 2005 ISBN-13: 978-0130473943
[Stephen A. Ross, Randolph Westerfield, Bradford D. Jordan] Essentials of Corporate Finance 6th Edition McGraw-Hill; 6th 2007 ASIN: B0041W2KEC
[Aslam Kassimali] Structural Analysis, SI Edition, 4th Edition, Kindle Edition CL Engineering; 4th 2010 ISBN-13: 978-0495295679
[Anil K. Chopra] Dynamics of Structures 3rd Edition TBS, Pearson, 3rd 2006 ISBN 13: 9788131713297
[Kenneth J. Waldron, G. L. Kinzel] Kinematics, Dynamics, and Design of Machinery 2nd Edition Wiley; 2 2003 ISBN-13: 978-0471244172
Elements of Chemical Reaction Engineering (4th Ed., Fogler)
Process Dynamics: Modeling, Analysis and Simulation (Wayne Bequette)
Dennis G. Zill, Michael R. Cullen, Warren S. Wright, Carol D. Wright - Solutions Manual to Accompany Advanced Engineering Mathematics (2006, Jones & Bartlett)
Hoffmann, Laurence D._ Rosen, Kenneth H. - Solutions Manual For Use With Applied Calculus For Business, Economics, And The Social And Life Sciences, Expanded. (2004, McGraw-Hill Science En
Introduction to Programming Using Visual Basic 2008 (w/VS2008 DVD), An, 7/E David I. Schneider
[Carter Robert L.] Molecular Symmetry and Group Theory Wiley 1997 ISBN-13: 978-0471149552
[Pru Marriott, J R Edwards, Howard J Mellett] Introduction to Accounting 3rd Edition SAGE Publications Ltd; 3rd 2002 ISBN-13: 978-0761970385
[Warren L. Stutzman, Gary A. Thiele] Antenna Theory and Design 2nd Edition Wiley; 2nd 1997 ISBN-13: 978-0471025900
[Taylor, Samuel Karlin, Howard M.] An Introduction to Stochastic Modeling 3rd Edition Academic Press; 3rd 1998 ISBN-13: 978-0126848878
Precalculus (4TH Ed., Margaret L. Lial, John Hornsby & David I. Schneider)
George B. Arfken, Hans J. Weber, Frank E. Harris - Instructor's Manual_ Mathematical Methods for Physicists (2014, Elsevier)
[Barry L. Nelson] Stochastic Modeling: Analysis and Simulation 2010 Dover Publications ISBN-13: 978-0486477701
Chemistry: The Molecular Nature of Matter and Change (4th Ed., Martin Silberberg)
Phase Equilibria, Phase Diagrams and Phase Transformations – Their Thermodynamic Basis (2nd Ed., Mats Hillert)
Reinforced Concrete: Mechanics and Design (5th Ed., James G. MacGregor & James K. Wight)
[Gary L. Gray, Francesco Costanzo, Michael E. Plesha] Engineering Mechanics: Dynamics 2007 McGraw-Hil ISBN-13: 978-0073358253
[Howard Curtis] Orbital Mechanics for Engineering Students 2nd Edition Butterworth-Heinemann 2nd 2009 ISBN: 9780080887845
[Stephen A. Ross, Randoloh W. Westerfield, Jeffrey Jaffe] Corporate Finance, 9th Edition McGraw-Hill/Irwin; 9th 2009 ISBN-13: 978-0073382333
College Algebra (10th Ed., Margaret L. Lial, John Hornsby & David I. Schneider)
[Stephen A. Ross, Randolph W. Westerfield, Jeffrey Jaffe] Corporate Finance, 9th Edition McGraw-Hill/Irwin; 9th ISBN-13: 978-0077398309
Margaret L. Lial ,Raymond N. Greenwell ,Nathan P. Ritchey - Instructor's Solutions Manual to Calculus with Applications (2012, Pearson)
[Dennis G. Zill, Warren S. Wright] Advanced Engineering Mathematics 5th Edition Kindle Jones & Bartlett Learning; 5th 2012 ISBN-13: 978-1449691721
Equilibrium and Non-Equilibrium Statistical Thermodynamics (Michel Le Bellac, Fabrice Mortessagne & G. George Batrouni)
Practical Interfacing in the Laboratory: Using a PC for Instrumentation, Data Analysis and Control (Stephen E. Derenzo)
Introductory Algebra (3rd Ed., K. Elayn Martin-Gay)
[Martin H. Sadd Elasticity: Theory, Applications, and Numerics 2nd Edition 2009 Academic Press ISBN-13: 978-0123744463
The Management of Construction: A Project Lifecycle Approach (F. Lawrence Bennett)
[Analysis and Design of Analog Integrated Circuits 5th Ed (vol.1) ch1-4 by Gray, Meyer
Elementary Principles of Chemical Processes (3rd Ed., Felder & Rousseau)
Applying PIC18 Microcontrollers: Architecture, Programming, and Interfacing using C and Assembly (Barry B. Brey)
Financial Reporting and Analysis,3/E Lawrence Revsine
College Algebra with Modeling and Visualization (3rd Ed., Gary K. Rockswold)
[Paul M. Fischer, William J. Taylor, Rita H. Cheng] Fundamentals of Advanced Accounting 1st Edition Cengage Learning; 1st 2007 ISBN-13: 978-0324378900
ADVANCED CALCULUS Gerald B. Folland
[Jeffry D. Madura, Carey Bissonnette, Ralph H. Petrucci, F. Geoffrey Herring] General Chemistry: Principles and Modern Applications 10th Edition Kindle Pearson Prentice Hall; 10th 2010 ISBN-13: 978-0132064521
[Harris Randy] Modern Physics 2nd Edition Pearson; 2nd 2007 ISBN-13: 978-0805303087
Advanced Topics in Finite Element Analysis of Structures: With Mathematica and MATLAB Computations (Asghar Bhatti)
Finite Mathematics, 9/E, by Margaret L. Lial , Raymond N. Greenwell , Nathan P. Ritchey
Structural Concrete: Theory and Design (3rd Ed., M. Nadim Hassoun & Akthem Al-Manaseer)
[Frank K. Reilly, Keith C. Brown] Investment Analysis and Portfolio Management South-Western; 8th 2005 ISBN-13: 978-0324289053
Aircraft Digital Electronic and Computer Systems: Principles, Operation and Maintenance (Mike Tooley)
[Jonathan Clayden, Nick Greeves, Stuart Warren, Peter Wothers] Organic Chemistry 1st Edition Oxford University Press; 1st ISBN-13: 978-0198503460

heste...@gmail.com

unread,
Sep 2, 2018, 5:57:13 AM9/2/18
□

Reply to author

Sign in to reply to author

Forward

Sign in to forward

Delete

You do not have permission to delete messages in this group

Solutions Manual Linear Algebra and Its Applications 4th Edition by David C. Lay
Источник: https://groups.google.com/
8 5 2 6 . Filenames are not case sensitive. Folder: A container holding files and other folders. Folders also are known as directories. Subfolder: A folder contained inside another folder. Path: A sequence of folders, separated by backslashes (\), where each folder is a subfolder of the folder preceding it. Paths are used to identify the locations of folders and files. An example is: Programs\Ch07\Text_Files_for_Exercises

Filespec: An abbreviation of file specification, it is the combination of a drive letter followed by a colon, a path, and a filename. An example is:
C:\Programs\Ch07\Text_Files_for_Exercises\USPRES.TXT

In practice, you rarely have to type a filespec, since both Windows and Visual Basic provide Browse facilities that locate files and folders for you. Windows Explorer: A program used to view, organize, and manage the folders and files on your disks. The details are presented in Appendix B in the section Manage Files and Folders with Windows Explorer. To invoke Windows Explorer, right-click the Windows Start button and click on Explore in the context menu that appears. Displaying File Extensions: By default, Windows shows only the base names of files. The following steps get Windows to also display the extensions. (In this book we assume that extensions are always shown). Windows XP 1. From Windows Explorer, press Alt/T/O to display the Folder Options dialog box. 2. Click on the View tab in the dialog box. (The dialog box in Figure 1.3 will appear.) 3. If there is a check mark in the box next to "Hide extensions for known file types," click on the box to remove the check mark. 4. Click on the OK button to close the Folder Options dialog box. Windows Vista 1. Click on the Start button. 2. Type Folder Options into the Start Search box (just above the Start button) and press the Enter key. (The Folder Options dialog box appears.) 3. Click on the View tab in the Folder Options dialog box. (The dialog box will appear as in Figure 1.4.)

FIGURE 1.3

XP Folder Options dialog box.

4. If there is a check mark in the box next to "Hide extensions for known file types," click on the box to remove the check mark. 5. Click on the OK button to close the Folder Options dialog box.

FIGURE 1.4

Vista Folder Options dialog box.

1.3

Program Development Cycle

We learned in Section 1.1 that hardware refers to the machinery in a computer system (such as the monitor, keyboard, and CPU) and software refers to a collection of instructions, called a program, that directs the hardware. Programs are written to solve problems or perform tasks on a computer. Programmers translate the solutions or tasks into a language the computer can understand. As we write programs, we must keep in mind that the computer will only do what we instruct it to do. Because of this, we must be very careful and thorough with our instructions. Note: A program is also known as a project, application, or solution.

■ Performing a Task on the Computer The first step in writing instructions to carry out a task is to determine what the output should be—that is, exactly what the task should produce. The second step is to identify the data, or input, necessary to obtain the output. The last step is to determine how to process the input to obtain the desired output, that is, to determine what formulas or ways of doing things can be used to obtain the output. This problem-solving approach is the same as that used to solve word problems in an algebra class. For example, consider the following algebra problem: How fast is a car traveling if it goes 50 miles in 2 hours? The first step is to determine the type of answer requested. The answer should be a number giving the speed in miles per hour (the output). (Speed is also called velocity.) The information needed to obtain the answer is the distance and time the car has traveled (the input). The formula speed = distance/time is used to process the distance traveled and the time elapsed in order to determine the speed. That is, speed = 50 miles/2 hours = 25 miles/hour A pictorial representation of this problem-solving process is

Input

Processing

Output

We determine what we want as output, get the needed input, and process the input to produce the desired output. In the following chapters we discuss how to write programs to carry out the preceding operations. But first we look at the general process of writing programs.

Program Development Cycle

■ Program Planning A baking recipe provides a good example of a plan. The ingredients and the amounts are determined by what is to be baked. That is, the output determines the input and the processing. The recipe, or plan, reduces the number of mistakes you might make if you tried to bake with no plan at all. Although it's difficult to imagine an architect building a bridge or a factory without a detailed plan, many programmers (particularly students in their first programming course) try to write programs without first making a careful plan. The more complicated the problem, the more complex the plan may be. You will spend much less time working on a program if you devise a carefully thought out stepby-step plan and test it before actually writing the program. Many programmers plan their programs using a sequence of steps, referred to as the program development cycle. The following step-by-step process will enable you to use your time efficiently and help you design error-free programs that produce the desired output. 1. Analyze: Define the problem. Be sure you understand what the program should do, that is, what the output should be. Have a clear idea of what data (or input) are given and the relationship between the input and the desired output. 2. Design: Plan the solution to the problem. Find a logical sequence of precise steps that solve the problem. Such a sequence of steps is called an algorithm. Every detail, including obvious steps, should appear in the algorithm. In the next section, we discuss three popular methods used to develop the logic plan: flowcharts, pseudocode, and top-down charts. These tools help the programmer break a problem into a sequence of small tasks the computer can perform to solve the problem. Planning also involves using representative data to test the logic of the algorithm by hand to ensure that it is correct. 3. Choose the interface: Select the objects (text boxes, buttons, etc.). Determine how the input will be obtained and how the output will be displayed. Then create objects to receive the input and display the output. Also, create appropriate buttons and menus to allow the user to control the program. 4. Code: Translate the algorithm into a programming language. Coding is the technical word for writing the program. During this stage, the program is written in Visual Basic and entered into the computer. The programmer uses the algorithm devised in Step 2 along with a knowledge of Visual Basic. 5. Test and debug: Locate and remove any errors in the program. Testing is the process of finding errors in a program, and debugging is the process of correcting errors that are found. (An error in a program is called a bug.) As the program is typed, Visual Basic points out certain types of program errors. Other types of errors will be detected by Visual Basic when the program is executed; however, many errors due to typing mistakes, flaws in the algorithm, or incorrect use of the Visual Basic language rules can be uncovered and corrected only by careful detective work. An example of such an error would be using addition when multiplication was the proper operation.

6. Complete the documentation: Organize all the material that describes the program. Documentation is intended to allow another person, or the programmer at a later date, to understand the program. Internal documentation (comments) consists of statements in the program that are not executed, but point out the purposes of various parts of the program. Documentation might also consist of a detailed description of what the program does and how to use the program (for instance, what type of input is expected). For commercial programs, documentation includes an instruction manual and on-line help. Other types of documentation are the flowchart, pseudocode, and hierarchy chart that were used to construct the program. Although documentation is listed as the last step in the program development cycle, it should take place as the program is being coded.

1.4

Programming Tools

This section discusses some specific algorithms and develops three tools used to convert algorithms into computer programs: flowcharts, pseudocode, and hierarchy charts. You use algorithms every day to make decisions and perform tasks. For instance, whenever you mail a letter, you must decide how much postage to put on the envelope. One rule of thumb is to use one stamp for every five sheets of paper or fraction thereof. Suppose a friend asks you to determine the number of stamps to place on an envelope. The following algorithm will accomplish the task. 1. Request the number of sheets of paper; call it Sheets. 2. Divide Sheets by 5. 3. Round the quotient up to the next highest whole number; call it Stamps. 4. Reply with the number Stamps.

(input) (processing) (processing) (output)

The preceding algorithm takes the number of sheets (Sheets) as input, processes the data, and produces the number of stamps needed (Stamps) as output. We can test the algorithm for a letter with 16 sheets of paper. 1. 2. 3. 4.

Request the number of sheets of paper; Sheets = 16. Dividing 5 into 16 gives 3.2. Rounding 3.2 up to 4 gives Stamps = 4. Reply with the answer, 4 stamps. This problem-solving example can be pictured by

Input (16)

Processing (formulas)

Output (4)

Of the program design tools available, three popular tools are the following: Flowcharts: Graphically depict the logical steps to carry out a task and show how the steps relate to each other. Pseudocode: Uses English-like phrases with some Visual Basic terms to outline the task. Hierarchy charts: Show how the different parts of a program relate to each other.

■ Flowcharts A flowchart consists of special geometric symbols connected by arrows. Within each symbol is a phrase presenting the activity at that step. The shape of the symbol indicates the type of operation that is to occur. For instance, the parallelogram denotes input or output. The arrows connecting the symbols, called flowlines, show the progression in which the steps take place. Flowcharts should "flow" from the top of the page to the bottom. Although the symbols used in flowcharts are standardized, no standards exist for the amount of detail required within each symbol. Symbol

Name

Meaning

Flowline

Used to connect symbols and indicate the flow of logic.

Terminal

Used to represent the beginning (Start) or the end (End) of a task.

Input/Output

Used for input and output operations, such as reading and displaying. The data to be read or displayed are described inside.

Processing

Used for arithmetic and data-manipulation operations. The instructions are listed inside the symbol.

Decision

Used for any logic or comparison operations. Unlike the input/ouput and processing symbols, which have one entry and one exit flowline, the decision symbol has one entry and two exit paths. The path chosen depends on whether the answer to a question is "yes" or "no."

Connector

Used to joint different flowlines.

Offpage Connector

Used to indicate that the flowchart continues to a second page.

Predefined Process

Used to represent a group of statements that perform one processing task.

Annotation

Used to provide additional information about another flowchart symbol.

Start

Read sheets

input

Set stamps sheets / 5

processing

Round stamps up to next whole number

processing

Display stamps

output

End FIGURE 1.5

Flowchart for the postage stamp problem.

The table of the flowchart symbols shown on the previous page has been adopted by the American National Standards Institute (ANSI). Figure 1.5 shows the flowchart for the postage stamp problem. The main advantage of using a flowchart to plan a task is that it provides a pictorial representation of the task, which makes the logic easier to follow. We can clearly see every step and how each is connected to the next. The major disadvantage with flowcharts is that when a program is very large, the flowcharts may continue for many pages, making them difficult to follow and modify.

■ Pseudocode Pseudocode is an abbreviated plain English version of actual computer code (hence, pseudocode). The geometric symbols used in flowcharts are replaced by English-like statements that outline the process. As a result, pseudocode looks more like computer code than does a flowchart. Pseudocode allows the programmer to focus on the steps required to solve a problem rather than on how to use the computer language. The

programmer can describe the algorithm in Visual Basic-like form without being restricted by the rules of Visual Basic. When the pseudocode is completed, it can be easily translated into the Visual Basic language. The following is pseudocode for the postage stamp problem:

```
Program: Determine the proper number of stamps for a letter
Read Sheets
Set the number of stamps to Sheets / 5
Round the number of stamps up to the next whole number
Display the number of stamps
```

(input) (processing) (processing) (output)

Pseudocode has several advantages. It is compact and probably will not extend for many pages as flowcharts commonly do. Also, the plan looks like the code to be written and so is preferred by many programmers.

■ Hierarchy Chart The last programming tool we'll discuss is the hierarchy chart, which shows the overall program structure. Hierarchy charts are also called structure charts, HIPO (Hierarchy plus Input-Process-Output) charts, top-down charts, or VTOC (Visual Table of Contents) charts. All these names refer to planning diagrams that are similar to a company's organization chart. Hierarchy charts depict the organization of a program but omit the specific processing logic. They describe what each part, or module, of the program does and they show how the modules relate to each other. The details on how the modules work, however, are omitted. The chart is read from top to bottom and from left to right. Each module may be subdivided into a succession of submodules that branch out under it. Typically, after the activities in the succession of submodules are carried out, the module to the right of the original module is considered. A quick glance at the hierarchy chart reveals each task performed in the program and where it is performed. Figure 1.6 shows a hierarchy chart for the postage stamp problem.

Postage stamp program

Read sheets

Calculate stamps

Set stamps sheets / 5 FIGURE 1.6

Display stamps

Round stamps up to next whole number

Hierarchy chart for the postage stamp problem.

◆

13

14

◆

Chapter 1 An Introduction to Computers and Problem Solving

The main benefit of hierarchy charts is in the initial planning of a program. We break down the major parts of a program so we can see what must be done in general. From this point, we can then refine each module into more detailed plans using flowcharts or pseudocode. This process is called the divide-and-conquer method. The postage stamp problem was solved by a series of instructions to read data, perform calculations, and display results. Each step was in a sequence; that is, we moved from one line to the next without skipping over any lines. This kind of structure is called a sequence structure. Many problems, however, require a decision to determine whether a series of instructions should be executed. If the answer to a question is "Yes," then one group of instructions is executed. If the answer is "No," then another is executed. This structure is called a decision structure. Figure 1.7 contains the pseudocode and flowchart for a decision structure.

No

If condition is true Then Process step(s) 1 Else Process step(s) 2 End If

FIGURE 1.7

Is condition true?

Yes

Process step(s) 2

Process step(s) 1

Pseudocode and flowchart for a decision structure.

The sequence and decision structures are both used to solve the following problem.

■ Direction of Numbered NYC Streets Algorithm Problem: Given a street number of a one-way street in New York, decide the direction of the street, either eastbound or westbound. Discussion: There is a simple rule to tell the direction of a one-way street in New York: Even-numbered streets run eastbound. Input: Street number Processing: Decide if the street number is divisible by 2. Output: "Eastbound" or "Westbound" Figures 1.8 through 1.10 show the flowchart, pseudocode, and hierarchy chart for the New York City numbered streets problem. The solution to the next problem requires the repetition of a series of instructions. A programming structure that executes instructions many times is called a loop structure. We need a test (or decision) to tell when the loop should end. Without an exit condition, the loop would repeat endlessly (an infinite loop). One way to control the number of times a loop repeats (often referred to as the number of passes or iterations)

1.4

Programming Tools

Start

Get street

No

Is street even?

Display westbound

Yes

Display eastbound

End

FIGURE 1.8

Flowchart for the New York City numbered streets problem.

Program: Determine the direction of a numbered NYC street. Get street If street is even Then Display Eastbound Else Display Westbound End If FIGURE 1.9

Pseudocode for the New York City numbered streets problem.

Street direction program

Get street number FIGURE 1.10

Decide whether street number is even or odd

Display direction

Hierarchy chart for the New York City numbered streets problem.

Is condition true? Do While condition is true Process step(s) Loop

No

Yes

Process step(s)

FIGURE 1.11

Pseudocode and flowchart for a loop.

is to check a condition before each pass through the loop and continue executing the loop as long as the condition is true. See Figure 1.11.

■ Class Average Algorithm Problem: Calculate and report the grade-point average for a class. Discussion: The average grade equals the sum of all grades divided by the number of students. We need a loop to read and then add (accumulate) the grades for each student in the class. Inside the loop, we also need to total (count) the number of students in the class. See Figures 1.12 to 1.14. Input: Student grades Processing: Find the sum of the grades; count the number of students; calculate average grade = sum of grades / number of students. Output: Average grade

■ Comments 1. Tracing a flowchart is like playing a board game. We begin at the Start symbol and proceed from symbol to symbol until we reach the End symbol. At any time, we will be at just one symbol. In a board game, the path taken depends on the result of spinning a spinner or rolling a pair of dice. The path taken through a flowchart depends on the input. 2. The algorithm should be tested at the flowchart stage before being coded into a program. Different data should be used as input, and the output checked. This process is known as desk checking. The test data should include nonstandard data as well as typical data.

Start

Initialize counter and sum to 0

Is there more data?

counter and sum start at 0

No

Yes Get next grade

read next grade

Increment counter

add 1 to counter

Add grade to sum

accumulate sum of grades

Set average to sum/counter

find the average

Display average

display the answer

End FIGURE 1.12

Flowchart for the class average problem.

3. Flowcharts, pseudocode, and hierarchy charts are universal problem-solving tools. They can be used to construct programs in any computer language, not just Visual Basic. 4. Flowcharts are used throughout this text to provide a visualization of the flow of certain programming tasks and Visual Basic control structures. Major examples of pseudocode and hierarchy charts appear in the case studies.

◆

Program: Calculate and report the average grade of a class. Initialize Counter and Sum to 0 Do While there is more data Get the next Grade Increment the Counter Add the Grade to the Sum Loop Compute Average = Sum/Counter Display Average FIGURE 1.13

Pseudocode for the class average problem.

Class average problem

Get grade

Compute sum and number of grades FIGURE 1.14

Calculate average

Display average

Hierarchy chart for the class average problem.

5. There are four primary logical programming constructs: sequence, decision, loop, and unconditional branch. Unconditional branch, which appears in some languages as GoTo statements, involves jumping from one place in a program to another. Structured programming uses the first three constructs but forbids the fourth. One advantage of pseudocode over flowcharts is that pseudocode has no provision for unconditional branching and thus forces the programmer to write structured programs. 6. Flowcharts are time consuming to write and difficult to update. For this reason, professional programmers are more likely to favor pseudocode and hierarchy charts. Because flowcharts so clearly illustrate the logical flow of programming techniques, however, they are a valuable tool in the education of programmers. 7. There are many styles of pseudocode. Some programmers use an outline form, whereas others use a form that looks almost like a programming language. The pseudocode appearing in the case studies of this text focuses on the primary tasks to be performed by the program and leaves many of the routine details to be completed during the coding process. Several Visual Basic keywords, such as, If, Else, Do, and While, are used extensively in the pseudocode appearing in this text. 8. Many people draw rectangles around each item in a hierarchy chart. In this text, rectangles are omitted in order to make hierarchy charts easier to draw and thereby to encourage their use.

2.1

An Introduction to Visual Basic 2008

Visual Basic 2008 is the latest generation of Visual Basic, a language used by millions of software developers. Visual Basic was designed to make user-friendly programs easier to develop. Prior to the creation of Visual Basic, developing a friendly user interface usually required a programmer to use a language such as C or C++, often requiring hundreds of lines of code just to get a window to appear on the screen. Now the same program can be created with much less time and fewer instructions.

■ Why Windows and Why Visual Basic? What people call graphical user interfaces, or GUIs (pronounced "gooies"), have revolutionized the computer industry. Instead of the confusing textual prompts that earlier users once saw, today's users are presented with such devices as icons, buttons, and drop-down lists that respond to mouse clicks. Accompanying the revolution in how programs look was a revolution in how they feel. Consider a program that requests information for a database. Figure 2.1 shows how a program written before the advent of GUIs got its information. The program requests the six pieces of data one at a time, with no opportunity to go back and alter previously entered information. Then the screen clears and the six inputs are again requested one at a time. Enter name (Enter EOD to terminate): Mr. President Enter Address: 1600 Pennsylvania Avenue Enter City: Washington Enter State: DC Enter Zipcode: 20500 Enter Phone Number: 202-456-1414 FIGURE 2.1

Input screen of a pre-Visual Basic program to fill a database.

Figure 2.2 shows how an equivalent Visual Basic program gets its information.The boxes may be filled in any order. When the user clicks on a box with the mouse, the cursor moves to that box. The user can either type in new information or edit the existing information. When the user is satisfied that all the information is correct, he or she just clicks on the "Write to Database" button. The boxes will clear, and the data

FIGURE 2.2

Input screen of a Visual Basic program to fill a database.

for another person can be entered. After all names have been entered, the user clicks on the Exit button. In Figure 2.1, the program is in control; in Figure 2.2, the user is in control!

■ How You Develop a Visual Basic Program One of the key elements of planning a Visual Basic program is deciding what the user sees—in other words, designing the screen. What data will he or she be entering? How large a window should the program use? Where will you place the buttons the user clicks on to activate actions by the program? Will the program have places to enter text (text boxes) and places to display output? What kind of warning boxes (message boxes) should the program use? In Visual Basic, the responsive objects a program designer places on windows are called controls. Two features make Visual Basic different from traditional programming tools: 1. You literally draw the user interface, much like using a paint program. 2. Perhaps more important, when you're done drawing the interface, the buttons, text boxes, and other objects that you have placed in a blank window will automatically recognize user actions such as mouse movements and button clicks. That is, the sequence of procedures executed in your program is controlled by "events" that the user initiates rather than by a predetermined sequence of procedures in your program. In any case, only after you design the interface does anything like traditional programming occur. Objects in Visual Basic recognize events like mouse clicks; how the objects respond to them depends on the instructions you write. You always need to write instructions in order to make controls respond to events. This makes Visual Basic programming fundamentally different from conventional programming. Programs in traditional programming languages ran from the top down. For these programming languages, execution started from the first line and moved with the flow of the program to different parts as needed. A Visual Basic program works differently. Its core is a set of independent groups of instructions that are activated by the events they have been told to recognize. This event-driven methodology is a fundamental shift. The user decides the order in which things happen, not the programmer. Most of the programming instructions in Visual Basic that tell your program how to respond to events like mouse clicks occur in what Visual Basic calls event procedures. Essentially, anything executable in a Visual Basic program either is in an event procedure or is used by an event procedure to help the procedure carry out its job. In fact, to stress that Visual Basic is fundamentally different from traditional programming languages, Microsoft uses the term project, rather than program, to refer to the combination of programming instructions and user interface that makes a Visual Basic program possible. Here is a summary of the steps you take to design a Visual Basic program: 1. Design the appearance of the window that the user sees. 2. Determine the events that the controls on the window should recognize. 3. Write the event procedures for those events.

Now here is what happens when the program is running: 1. Visual Basic monitors the controls in the window to detect any event that a control can recognize (mouse movements, clicks, keystrokes, and so on). 2. When Visual Basic detects an event, it examines the program to see if you've written an event procedure for that event. 3. If you have written an event procedure, Visual Basic executes the instructions that make up that event procedure and goes back to Step 1. 4. If you have not written an event procedure, Visual Basic ignores the event and goes back to Step 1. These steps cycle continuously until the program ends. Usually, an event must happen before Visual Basic will do anything. Event-driven programs are reactive more than active—and that makes them more user friendly.

■ The Different Versions of Visual Basic Visual Basic 1.0 first appeared in 1991. It was followed by version 2.0 in 1992, version 3.0 in 1993, version 4.0 in 1995, version 5.0 in 1997, and version 6.0 in 1998. VB.NET, initially released in February 2002, was not backward compatible with the earlier versions of Visual Basic. It incorporated many features requested by software developers, such as true inheritance and powerful Web capabilities. Visual Basic 2005, released in November 2005, and Visual Basic 2008, released in November 2007 are significantly improved versions of VB.NET.

2.2

Visual Basic Controls

Visual Basic programs display a Windows-style screen (called a form) with boxes into which users type (and in which users edit) information and buttons that they click to initiate actions.

The boxes and buttons are referred to as controls. In this section, we examine forms and four of the most useful Visual Basic controls.

■ Starting a New Visual Basic Program For our purposes, Visual Basic programs are also known as applications, solutions, or projects. Each program is saved (as several files and subfolders) in its own folder. Before starting a new program, you should use Windows Explorer to create a folder to hold the folders for your programs. The process for invoking Visual Basic varies slightly with the edition of Visual Basic installed on the computer. To invoke Visual Basic from a computer that has Visual Basic Express installed, click the Windows Start button, hover over All Programs, and then click on Microsoft Visual Basic 2008 Express Edition. With the other editions of Visual Basic, hover over All Programs, hover over Microsoft Visual Studio 2008, and then click on Microsoft Visual Studio 2008 in the short list that is revealed. The window that appears after Visual Basic is invoked is called the Start Page and is similar to the window shown in Figure 2.3. (The information in the large section in the middle of the screen comes from the Microsoft Web site and will differ each time Visual Basic is invoked.) The first item on the Start Page menu bar is "File." Click on File, and then click on New Project to produce a New Project dialog box. Figure 2.4 shows the

Menu bar

FIGURE 2.3

The Visual Basic 2008 Express Edition Start Page.

New Project dialog box produced by Visual Basic Express. The "Windows Forms Application" icon should be selected as the installed template. If this is not the case, click on "Windows Forms Application" to select it. (The other editions of Visual Basic contain a pane identifying a Project type. You should select "Visual Basic" as the Project type.) Note: The number of project types and icons showing will vary depending on the version of Visual Basic you are using. Figure 2.4 was created from the Express Edition.

FIGURE 2.4

The Visual Basic Express New Project dialog box.

The name of the program, initially set to WindowsApplication1, can be specified at this time. Since we will have a chance to change the name later, let's just call the program WindowsApplication1 for now. Click on the OK button to invoke the Visual Basic programming environment See Figure 2.5. The Visual Basic programming environment is referred to as the Integrated Development Environment or IDE. Very likely, your screen will look different than Figure 2.5. The IDE is extremely configurable. Each window in Figure 2.5 can have its location and size altered. New windows can be displayed in the IDE, and any window can be closed or hidden behind a tab. For instance, in Figure 2.5 the Toolbox window is hidden behind a tab. The View menu is used to add additional windows to the IDE. If you would like your screen to look exactly like Figure 2.5, click on "Reset Windows Layout" in the Windows menu, and then click on "Yes." The Menu bar of the IDE displays the commands you use to work with Visual Basic. Some of the menus, like File, Edit, View, and Window, are common to most Windows applications. Others, such as Project, Build, and Debug, provide commands specific to programming in Visual Basic. The Toolbar holds a collection of icons that carry out standard operations when clicked. For example, you use the fifth icon, which looks like a stack of three diskettes, to save the files associated with the current program. To reveal the purpose of a Toolbar icon, hover the mouse pointer over the icon for a few seconds. The little information rectangle that pops up is called a tooltip. The Main area currently holds the rectangular Form window, or form for short. The form becomes a Windows window when a program is executed. Most information displayed by the program appears on the form. The information usually is displayed in Start Page tab

Form Designer tab

Main area

Properties window

Solution Explorer window

Menu bar Toolbar Toolbox tab

Form

Sizing handle

FIGURE 2.5

The Visual Basic integrated development environment in Form Designer mode.

controls that the programmer has placed on the form.You can change the size of the form by dragging one of its sizing handles. The Properties window is used to change how objects look and react. The Solution Explorer window displays the files associated with the program and provides access to the commands that pertain to them. (Note: If the Solution Explorer or the Properties window is not visible, click on it in the View menu.) The Toolbox holds icons representing objects (called controls) that can be placed on the form. If your screen does not show the Toolbox, hover the mouse over the Toolbox tab at the left side of the screen. The Toolbox will slide into view. Then click on the pushpin icon in the title bar at the top of the Toolbox to keep the Toolbox permanently displayed in the IDE. (Note: If there is no tab marked Toolbox, click on Toolbox in the View menu.) The controls in the Toolbox are grouped into categories such as All Windows Forms and Common Controls. Figure 2.6 shows the Toolbox after the plus sign to the left of "Common Controls" has been clicked. Nearly all the controls discussed in this text can be found in the list of common controls. (You can obtain a description of a control by hovering the mouse over the control.) The four controls discussed in this chapter are text boxes, labels, buttons, and list boxes. In order to see all the group names, click on each of the minus signs appearing to the left of a group name.

Pushpin

Group names

FIGURE 2.6

The Toolbox's common controls.

Text boxes: You use a text box to get information from the user, referred to as input, or to display information produced by the program, referred to as output. Labels: You place a label near a text box to tell the user what type of information is displayed in the text box. Buttons: The user clicks a button to initiate an action. List boxes: In the first part of the book, we use list boxes to display tables or several lines of output. Later, we use list boxes to make selections.

■ A Text Box Walkthrough 1. Double-click on the text box icon in the Common Controls portion of the Toolbox. A rectangle with two small squares, called sizing handles, appears at the upper left corner of the form. (You can alter the width of the text box by dragging one of the sizing handles.) Move the mouse arrow to any point of the text box other than a sizing handle, hold down the left mouse button, and drag the text box to the center of the form. See Figure 2.7. Note: The Tasks button will be discussed in Chapter 3. Tasks button

Sizing handles FIGURE 2.7

A text box with sizing handles.

2. Click anywhere on the form outside the rectangle to deselect the text box. 3. Click on the rectangle to restore the handles. An object showing its handles is said to be selected. A selected text box can have its width altered, location changed, and other properties modified. 4. Move the mouse arrow to the handle in the center of the right side of the text box. The cursor should change to a double arrow 142. Hold down the left mouse button, and move the mouse to the right. The text box is stretched to the right. Similarly, grabbing the text box on the left side and moving the mouse to the left stretches the text box to the left. You also can use the handles to make the text box smaller. Steps 1 and 4 allow you to place a text box of any width anywhere on the form. Note: The text box should now be selected; that is, its sizing handles should be showing. If not, click anywhere inside the text box to select it. 5. Press the delete key, Del, to remove the text box from the form. Step 6 gives an alternative way to place a text box of any width at any location on the form. 6. Click on the text box icon in the Toolbox. Then move the mouse pointer to any place on the form. (When over the form, the mouse pointer becomes a pair of crossed thin lines.) Hold down the left mouse button, and drag the mouse on a diagonal to generate a rectangle. Release the mouse button to obtain a selected

text box. You can now alter the width and location as before. Note: The text box should now be selected. If not, click anywhere inside the text box to select it. 7. Press F4 to activate the Properties window. (You also can activate the Properties window by clicking on it, clicking on the Properties window icon in the right part of the Toolbar, selecting Properties Window from the View menu, or clicking on the text box with the right mouse button and selecting Properties.) See Figure 2.8. The first line of the Properties window (called the Object box) reads "TextBox1 etc." TextBox1 is the current name of the text box. The first two buttons below the Object box permit you to view the list of properties either grouped into categories or alphabetically. Use the up- and down-arrow keys (or the up- and down-scroll arrows) to move through the list. The left column gives the property names, and the right column gives the current settings of the properties. We discuss four properties in this walkthrough.

Object box

Description pane

Categorized view

FIGURE 2.8

Alphabetic view

Properties window icon

Text box Properties window.

Note 1: The third and fourth buttons below the Object box, the Properties button and the Events button, determine whether properties or events are displayed in the Properties window. Normally the Properties button is highlighted. If not, click on the Properties button. Note 2: If the Description pane is not visible, right-click on the Properties window, and then click on "Description." The Description pane describes the currently highlighted property.

8. Move to the Text property with the up- and down-arrow keys. (Alternatively, scroll until the property is visible, and click on the property.) The Text property, which determines the words displayed in the text box, is now highlighted. Currently, there is no text displayed in the Settings box on the right. 9. Type your first name. Then press the Enter key, or click on another property. Your name now appears in both the Settings box and the text box. See Figure 2.9.

FIGURE 2.9

Setting the Text property to David.

10. Click at the beginning of your name in the Text Settings box, and add your title, such as Mr., Ms., or The Honorable. (If you mistyped your name, you can easily correct it now.) Then, press Enter. 11. Use the up-arrow key or the mouse to move to the ForeColor property. This property determines the color of the information displayed in the text box. 12. Click on the down arrow in the right part of the Settings box, and then click on the Custom tab to display a selection of colors. See Figure 2.10. Click on one of the

FIGURE 2.10

Setting the ForeColor property.

colors, such as blue or red. Notice the change in the color of your name. (Note: The sixteen white boxes at the bottom of the grid are used to create custom colors. See item J under "Design a Form" in Appendix B for details.) 13. Highlight the Font property with a single click of the mouse. The current font is named Microsoft Sans Serif. 14. Click on the ellipsis 1 Á 2 box in the right part of the Settings box to display a dialog box. See Figure 2.11. The three lists give the current name (Microsoft Sans Serif), current style (Regular), and current size (8) of the font. You can change any of these attributes by clicking on an item in its list or by typing into the box at the top of the list. Click on Bold in the style list, and click on 12 in the size list. Now click on the OK button to see your name displayed in a larger bold font. The text box will be longer so that it can accommodate the larger font.

FIGURE 2.11

The Font dialog box.

15. Click on the text box and resize it to be about 3 inches wide. Visual Basic programs consist of three parts: interface, values of properties, and code. Our interface consists of a form with a single object, a text box. We have set a few properties for the text box—the text (namely, your name), the foreground color, the font style, and the font size. In Section 2.3, we discuss how to place code into a program. Visual Basic endows certain capabilities to programs that are independent of any code we will write. We will now run the existing program without adding any code in order to experience these capabilities.

16. Press F5 to run the program. (Alternatively, a program can be run from the menu by pressing Alt/D/S or by clicking on the Start Debugging icon, the fourteenth or fifteenth icon on the Toolbar.) After a brief delay, a copy of the form appears that has neither the form nor the text box selected. 17. Your name is highlighted. Press the End key to move the cursor to the end of your name. Now type in your last name, and then keep typing. Eventually, the words will scroll to the left. 18. Press Home to return to the beginning of the text. You have a miniature word processor at your disposal. You can place the cursor anywhere you like to add or delete text. You can drag the cursor across text to select a block, place a copy of the block in the Clipboard with Ctrl+C, and then duplicate it elsewhere with Ctrl+V. 19. To end the program, press Alt + F4. Alternatively, you can end a program by clicking on the form's Close button at the right corner of the title bar. 20. Select the text box, activate the Properties window, select the ReadOnly property, click on the down-arrow button, and finally click on True. Notice that the background color is now gray. 21. Run the program, and try typing into the text box. You can't. Such a text box is used for output. Only code can display information in the text box. Note: In this textbook, whenever a text box will be used only for the purpose of displaying output, we will always set the ReadOnly property to True. 22. End the program. 23. Let's now save the program on a disk. Click on the Save All icon to save the work done so far. (The Save All icon is the fifth or sixth icon on the Toolbar. It shows three fanned diskettes. Alternately, you can click on Save All in the File menu.) You will be prompted for the name of the program and the path to the folder where you want the program to be saved. Type a name, such as "VBdemo". You can either type a path or use Browse to locate a folder. (This folder will automatically be used the next time you click on the Save All icon.) The files for the program will be held in a subfolder of the selected folder. Important: If the "Create directory for solution" check box is checked, then click on the check box to uncheck it. Finally, click on the Save button. 24. Create a new program as before by clicking on "New Project" on the File menu. (Or, you can click on the New Project icon, the first icon on the Toolbar.) A New Project dialog box will appear. 25. Give a name to the project, such as My Program, and then click on the OK button. 26. Place three text boxes on the form. (If you use the double-click technique, move the text boxes so that they do not overlap.) Notice that they have the names TextBox1, TextBox2, and TextBox3. 27. Run the program. Notice that the cursor is in TextBox1. We say that TextBox1 has the focus. (This means that TextBox1 is the currently selected object and any keyboard actions will be sent directly to this object.) Any text typed will display in that text box.

28. Press Tab once. Now, TextBox2 has the focus. When you type, the characters appear in TextBox2. 29. Press Tab several times, and then press Shift+Tab a few times. With Tab, the focus cycles through the objects on the form in the order the objects were created. With Shift+Tab, the focus cycles in the reverse order. 30. End the program you created. 31. We would now like to return to the first program. Click on "Open Project" from the File menu. An Open Project dialog box will appear stating that "You must choose to either save or discard changes in the current project before opening a project." There is no need to save this program, so click on the Discard button. Then a second Open Project dialog box will appear. 32. Find the folder corresponding to the program you saved earlier, double-click on the folder, and double-click on the file with extension sln. You have now recovered the first program. Note: As an alternative to using the Open Project dialog box in Steps 31 and 32 to return to the first program, click on the Start Page tab at the top of the Main area, and click on the program in the Recent Projects pane of the Start Page. 33. If you do not see the Form Designer for the program, double-click on Form1.vb in the Solution Explorer.

■ A Button Walkthrough 1. Click on the New Project icon to start a new program. (Give a name, such as ButtonProg, to the program, and click on OK.) 2. Double-click on the Button icon in the Toolbox to place a button on the form. (The Button icon is the second icon in the Common Controls portion of the Toolbox.) 3. Move the button to the center of the form. 4. Activate the Properties window, highlight the Text property, type "Please Push Me," and press Enter. See Figure 2.12. The button is too small.

FIGURE 2.12

Setting the Text property.

5. Click on the button to select it, and then enlarge it to accommodate the phrase “Please Push Me” on one line. 6. Run the program, and click on the button. The button appears to move in and then out. In Section 2.3, we will write code that is executed when a button is pushed. 7. End the program and select the button. 8. From the Properties window, edit the Text setting by inserting an ampersand (&) before the first letter, P. Press the Enter key, and notice that the first letter P on the button is now underlined. See Figure 2.13. Pressing Alt+P while the program is running triggers the same event as clicking the button. However, the button will not appear to move in and out. Here, P is referred to as the access key for the button. (The access key is always specified as the character following the ampersand.)

FIGURE 2.13

Designating P as an access key.

■ A Label Walkthrough 1. Click on the New Project icon to begin a new program. Feel free to select the default name, such as WindowsApplication1. 2. Double-click on the label icon to place a label on the form. (The label icon is a large letter A.) Move the label to the center of the form. 3. Activate the Properties window, highlight the Text property, type “Enter Your Phone Number:”, and press Enter. (Such a label would be placed next to a text box into which the user will type a phone number.) Notice that the label widened to accommodate the text. This happened because the AutoSize property of the label is set to True by default. 4. Change the AutoSize property to False. Press Enter. Notice that the label now has eight sizing handles when selected. 5. Make the label narrower and longer until the words occupy two lines. 6. Activate the Properties window, and click on the down arrow to the right of the setting for the TextAlign property. Experiment by clicking on the various rectangles and observing their effects. The combination of sizing and alignment permits you to design a label easily.

7. Run the program. Nothing happens, even if you click on the label. Labels just sit there. The user cannot change what a label displays unless you write code to make the change. 8. End the program.

■ A List Box Walkthrough 1. Click on the New Project icon to begin a new program. Feel free to select the default name, such as WindowsApplication1. 2. Place a list box on the form. (The list box icon is the ninth icon in the Common Controls group of the Toolbox.) 3. Press F4 to activate the Properties window and notice that the list box does not have a Text property. The word ListBox1 is actually the setting for the Name property. 4. Also place a text box, a button and a label on the form. 5. Click on the Object box of the Properties window. The name of the form and the names of the four controls are displayed. If you click on one of the names, that object will become selected and its properties displayed in the Properties window. 6. Run the program. Notice that the word ListBox1 has disappeared, but the words Button1 and Label1 are still visible. The list box is completely blank. In subsequent sections, we will write code to place information into the list box.

■ The Name Property Every control has a Name property. It is used in code to refer to the control. By default, controls are given names like TextBox1 and TextBox2. You can use the Properties window to change the Name property of a control to a more meaningful name. (The Name property is always the third property in the alphabetized list of properties. A control’s name must begin with a letter and can be a maximum of 215 characters. It can include numbers and underline (_) characters, but cannot include punctuation or spaces.) Also, it is good coding practice to have each name begin with a three- or four-letter prefix that identifies the type of the control. See Table 2.1. The form itself also has a Name property. Beginning with Section 2.3, we will use suggestive names and these prefixes whenever possible.

TABLE 2.1

Some controls and their three-letter prefixes.

Control	Prefix	Example
form	frm	frmLottery
button	btn	btnComputeTotal
label	lbl	lblInstructions
list box	lst	lstOutput
text box	txt	txtAddress

The Name property of the form itself also can be changed. Suppose you want to change the name of the form from Form1 to frmPayroll. The most efficient way to set the name of the form is to change the name of the file Form1.vb appearing in the Solution Explorer window to frmPayroll.vb. To make the change, right-click on Form1.vb in the

Solution Explorer window, click on Rename, type in the new name (frmPayroll.vb), and press Enter. Important: Make sure that the new filename has the extension “vb”. The Name and Text properties of a button are both initially set to something like Button1. However, changing one of these properties does not affect the setting of the other property, and similarly for the Name and Text properties of forms, text boxes, and labels. The Text property of a form specifies the words appearing in the form’s title bar.

■ Two Help Walkthroughs The Help menu provides four ways for you to obtain information about Visual Basic. You can browse topic titles using the Contents window, search for topics by keyword using the Index window, search the full text of topics using the Search page, or browse for topics by category using How Do I. The following walkthroughs demonstrate the Index and How Do I features. Index Walkthrough 1. Click on Index in the Help menu to invoke the Index window. 2. If the “Filtered by:” box does not say “Visual Basic” or “Visual Basic Express Edition,” click on the down arrow and select one of them from the drop-down list. 3. Type “buttons” into the “Look for” box, look down the list of subheadings of “buttons,” and then click on “introducing.” (The page that appears has the title “Interacting with the User: Using Buttons.”) 4. Read the first few paragraphs of the discussion. “How Do I” Walkthrough 1. Click on “How Do I” in the Help menu. The page that appears contains a short list of underlined links. 2. Click on the underlined link “Learn the Visual Basic Language (How Do I in

Visual Basic Express).” The page that appears contains another list of underlined links under the heading “In Visual Basic Express.” 3. Click on the underlined link “Introduction to the Visual Basic Programming Language.” 4. Read the introductory discussion of Visual Basic in the page that appears.

■ Fonts The default font for controls is Microsoft Sans Serif. Two other useful fonts are Courier New and Wingdings. Courier New is a fixed-width font; that is, each character has the same width. With such a font, the letter i occupies the same space as the letter m. Fixed-width fonts are used with tables when information is to be aligned in columns.

The Wingdings font consists of assorted small pictures and symbols, each corresponding to a character on the keyboard. For instance if one of the characters %, (, 1, or J is typed into the Text setting of a control whose Font is Wingdings, the control will display a bell, phone, open folder, or smiling face, respectively. To view the character set for a Windows font, click on the Windows Start button in the Windows task bar and successively select All Programs, Accessories, System Tools, and Character Map. Then click on Character Map, or press the Enter key. After selecting the font, click on any item to enlarge it. You can insert the keyboard character for the item into the Clipboard by pressing the Select button and then the Copy button. To place the character into the Text property of a control having that font, just move the cursor to the Settings box for the Text property and press Ctrl+ V.

■ Auto Hide The Auto Hide feature allows you to make more room for the Main area of the screen by hiding windows (such as the Toolbox, Solution Explorer, or Properties window). Let’s illustrate the feature with a walkthrough using the Toolbox window. We start by discussing the situation where the feature is disabled. 1. If the Toolbox window is currently visible and the pushpin icon in the window title is vertical, then the Auto Hide feature is disabled. (If the Toolbox window is not visible, press Alt/V/X to select Toolbox from the View menu. If the pushpin icon is horizontal, then click on the icon to make it vertical.) When the Auto Hide feature is disabled, the Toolbox window stays stationary and is always ready for use. 2. Click the mouse cursor somewhere outside of the Toolbox window and note that the Toolbox window stays fixed. 3. Click on the pushpin icon to make it horizontal. The Auto Hide feature is now enabled. 4. Move the mouse cursor somewhere outside of the Toolbox window and note that the window slides into a tab on the left side of the screen. The name and icon of the Toolbox window appear on the tab. 5. Hover the mouse cursor over the tab. The window slides into view and is ready for use. 6. Place a new control on the form, and then move the cursor away from the Toolbox window. The window automatically slides back to its tab on the edge of the screen. Note: We recommend that you keep the Toolbox, Solution Explorer, and Properties windows displayed unless you are creating a program with a very large form and need extra space.

■ Positioning and Aligning Controls Visual Basic provides several tools for positioning and aligning controls on a form. Proximity lines are short line segments that help you place controls a comfortable distance from each other and from the sides of the form. Snap lines are horizontal and vertical line segments that help you align controls. The Format menu is used to align controls, center controls horizontally and vertically in a form, and make a group of selected controls the same size.

A Positioning and Aligning Walkthrough 1. Begin a new program. 2. Place a button near the center of the form. 3. Drag the button toward the upper-right corner of the form until two short line segments appear. See Figure 2.14(a). The button is now a comfortable distance from the two sides of the form. 4. Place a second button below the first button and drag it upwards until a proximity line appears between the two buttons. The buttons are now a comfortable distance apart. 5. Resize and position the two buttons as shown in Figure 2.14(b). 6. Drag Button2 upwards until a blue line appears along the bottoms of the two buttons. See Figure 2.14(c). This blue line is called a snap line. The bottoms of the two buttons are now aligned. 7. Continue dragging Button2 upwards until a purple snap line appears just underneath the words Button1 and Button2. See Figure 2.14(d). The middles of the two buttons are now aligned. If we were to continue dragging Button2 upwards, a blue snap line would tell us when the tops are aligned. Step 10 shows another way to align the controls.

Proximity line

(b)

(a)

Snap line

(c)

(d) FIGURE 2.14

Positioning Controls.

8. Click on Button1 and then hold down the Ctrl key and click on Button2. After the mouse button is released, both buttons will be selected. Note: This process (called selection of multiple controls) can be repeated to select a group of any number of controls.

9. With the two buttons still selected, press F4 to open the Properties window. Then set the ForeColor property to Blue. Notice that the ForeColor property has been altered for both buttons at the same time. Actually, any property that is common to every control in selected multiple controls can be set simultaneously for the entire group. 10. With the two buttons still selected, open the Format menu in the Menu bar, hover over Align, and click on Tops. The tops of the two buttons are now aligned. Precisely, Button1 (the first button selected) will stay fixed, and Button2 will move up so that its top is aligned with the top of Button1. The most common uses of the submenus of the Format menu are as follows: Align: Align middles or corresponding sides of a group of selected controls. Make Same Size: Make the width and/or height of a group of selected controls the same. Horizontal Spacing: Equalize the horizontal spacing between a group of three or more selected controls arranged in a row. Vertical Spacing: Equalize the vertical spacing between a group of three or more selected controls arranged in a column. Center in Form: Center a selected control either horizontally or vertically in a form. When multiple controls are selected with the Ctrl key, the first control selected (called the primary control of the group) will have white sizing handles, while the other controls will have black sizing handles. All alignment and sizing statements initiated from the Format menu will keep the primary control fixed, and align (or size) the other controls with respect to the primary control. After multiple controls have been selected, they can be dragged as a group and deleted as a group. Exercises 35 and 36 show how the arrow keys can be used to move and size a control. The arrow keys also can be used to move and size multiple controls as a group. A group of controls also can be selected by clicking the mouse outside the controls, dragging it across the controls, and releasing it. The Select All command from the Edit menu (or the key combination Ctrl + A) causes all the controls on the form to be selected. Although these methods are easy to apply, they do not allow the programmer to choose the primary control.

■ Setting Tab Order Each time the Tab key is pressed while a program is running, the focus moves from one control to another. The following walkthrough explains how to determine the order in which the focus moves and how that order can be changed. 1. Start a new program. 2. Place a button, a text box, and a list box on a form. 3. Run the program, and successively press the Tab key. Notice that the controls receive the focus in the order they were placed on the form.

4. End the program. 5. Click on Tab Order in the View menu. The screen appears as in Figure 2.15(a). The controls are numbered from 0 to 2 in the order they were created. Each of the numbers is referred to as a tab index. 6. Click on the list box, then the button, and finally the text box. Notice that the tab indexes change as shown in Figure 2.15(b).

(a)

(b) FIGURE 2.15

7. Click again on Tab Order in the View menu to set the new tab order. 8. Run the program again, and successively press Tab. Notice that the controls receive the focus according to the new tab order. 9. End the program. 10. Add a label to the form, rerun the program, and successively press Tab. Notice that the label does not receive the focus. Whether or not a control can receive the focus is determined by the setting of its TabStop property. By default, the setting for TabStop property is True for buttons, text boxes, and list boxes, and False for labels. In this book we always use these default settings. Note: Even though labels do not receive the focus while tabbing, they are still assigned a tab index.

■ Comments 1. While you are working on a program, the program resides in memory. Removing a program from memory is referred to as closing the program. A program is automatically closed when you begin a new program. Also, it can be closed directly with the Close Project command from the File menu. 2. Three useful properties that have not been discussed are the following: (a) BackColor: This property specifies the background color for the form or a control.

(b) Visible: Setting the Visible property to False causes an object to disappear when the program is run. The object can be made to reappear with code. (c) Enabled: Setting the Enabled property of a control to False restricts its use. It appears grayed and cannot receive the focus. Controls sometimes are disabled temporarily if they do not apply to the current state of the program. 3. Most properties can be set or altered with code as the program is running instead of being preset from the Properties window. For instance, a button can be made to disappear with a line such as Button1.Visible = False. The details are presented in Section 2.3. 4. If you inadvertently double-click on a form, a window containing text will appear. (The first line is Public Class Form1.) This is the Code Editor, which is discussed in the next section. Press Ctrl+Z to undo the addition of this new code. To return to the Form Designer, click on the tab above the Main area labeled "Form1.vb [Design]." 5. We have seen two ways to place a control onto a form. A third method is to drag the control from the Toolbox to the form. 6. The Recent Projects pane in the Start Page not only displays a list of recently updated programs, it also lets you open existing programs and create new programs. To open any existing program or to create a new program, click on the appropriate word "Project…:" at the bottom of the pane. 7. The Getting Started pane in the Start Page displays a list of Help topics, Web sites, technical articles, and other sources that highlight features in the product. Practice Problems 2.2 1. What is the difference between the Text and the Name properties of a button? 2. The first two group names in the Toolbox are "All Windows Forms" and "Common Controls." How many groups are there? EXERCISES 2.2

1. Create a form with two buttons, run the program, and click on each button. Do you notice anything different about a button after it has been clicked? 2. While a program is running, a control is said to lose focus when the focus moves from that control to another control. In what three ways can the user cause a control to lose focus? In Exercises 3 through 24, carry out the task. Use a new program for each exercise. 3. 4. 5. 6.

Place "CHECKING ACCOUNT" in the title bar of a form. Create a text box containing the words "PLAY IT, SAM" in blue letters. Create a text box with a yellow background. Create a text box named txtGreeting and containing the word "HELLO" in large italic letters.

7. Create a label containing the sentence "After all is said and done, more is said than done." The sentence should occupy three lines, and each line should be centered horizontally in the label. 8. Create a read-only text box containing the words "Visual Basic" in bold white letters on a red background. 9. Create a text box named txtLanguage and containing the words "Visual Basic 2008" in Courier New font. 10. Create a yellow button named btnPush and containing the word "PUSH". 11. Create a white button containing the word "PUSH" in large italic letters. 12. Create a button containing the word "PUSH" in bold letters in which the letter P is underlined. 13. Create a button containing the word "PUSH" with the letter H as the access key. 14. Create a label containing the word "ALIAS" in white on a blue background. 15. Create a label named lblAKA and containing the centered italicized word "ALIAS". 16. Place BALANCE SHEET in the title bar of a form, where the form has a yellow background. 17. Create a label containing VISUAL on the first line and BASIC on the second line. Each word should be right justified. 18. Create a form named frmHello whose title bar reads "Hello World". 19. Create a label containing a picture of a diskette. (Hint: Use the Wingdings character 6.) Make the diskette as large as possible. 20. Create a label containing the bold word "ALIAS" in the Courier New font. 21. Create a list box with a yellow background. 22. Create a list box that will be invisible when the program is run. 23. Create a form named frmYellow with a yellow background. 24. Create a button containing a picture of a red bell. (Hint: Use the Wingdings character %.) Make the bell as large as possible. In Exercises 25 through 30, create the interface shown in that figure. (These exercises give you practice creating controls and assigning properties. The interfaces do not necessarily correspond to actual programs.) 25.

26.

27.

28.

29.

30.

31. Create a replica of your bank check on a form. Words common to all checks, such as "PAY TO THE ORDER OF," should be contained in labels. Items specific to your checks, such as your name at the top left, should be contained in text boxes. Make the check on the screen resemble your personal check as much as possible. 32. Create a replica of your campus ID on a form. Words that are on all student IDs, such as the name of the college, should be contained in labels. Information specific to your ID, such as your name and Social Security number, should be contained in text boxes. Simulate your picture with a text box containing a smiling face—a size 24 Wingdings J. 33. Consider the form shown in Exercise 25. Assume the Batman button was added to the form before the Robin button. What is the tab index of the Robin button? 34. Consider the form shown in Exercise 26. Assume the first control added to the form was the label. What is the tab index of the label? The following hands-on exercises develop additional techniques for manipulating and accessing controls placed on a form. 35. Place a text box on a form and select the text box. What is the effect of pressing the various arrow keys? 36. Place a text box on a form and select the text box. What is the effect of pressing the various arrow keys while holding down the Shift key? 37. Repeat Exercise 36 for selected multiple controls. 38. Repeat Exercise 35 for selected multiple controls. 39. Place a label and a list box on a form and change their font sizes to 12 at the same time. 40. Place a button in the center of a form and select it. Hold down the Ctrl key and press an arrow key. Repeat this process for each of the other arrow keys. Describe what happens.

41. Place a label and a text box on a form with the label to the left of and above the text box. Select the label. Hold down the Ctrl key and press the down-arrow key twice. With the Ctrl key still pressed, press the right-arrow key. Describe what happens. 42. Place two buttons on a form with one button to the right of and below the other button. Select the lower button, hold down the Ctrl key, and press the left-arrow key. With the Ctrl key still pressed, press the up-arrow key. Describe the effect of pressing the two arrow keys. 43. Experiment with the Align command on the Format menu to determine the difference between the center and the middle of a control. 44. Place four large buttons vertically on a form. Use the Format menu to make them the same size and to make the spacing between them uniform. 45. Place a text box on a form, select the text box, and open its Properties window. Double-click on the name (not the Settings box) of the ReadOnly property. Double-click again. What is the effect of double-clicking on a property whose possible settings are True and False? 46. Place a button on a form, select the button, and open its Properties window. Double-click on the name (not the Settings box) of the ForeColor property. Double-click repeatedly. Describe what is happening. Solutions to Practice Problems 2.2

1. The text is the words appearing on the button, whereas the name is the designation used to refer to the button. Initially, they have the same value, such as Button1. However, each can be changed independently of the other.

2. 11 groups. Figure 2.16 shows the Toolbox after each of the - signs preceding group names have been converted to + signs.

Until Chapter 9, all of the controls discussed in this book can be found in the Common Controls group. The group General, which holds text fragments rather than controls, will first be used in Section 3.3 to hold snippets of programming code.

FIGURE 2.16

Toolbox group names

2.3

Visual Basic Events

When a Visual Basic program runs, the form and its controls appear on the screen. Normally, nothing happens until the user takes an action, such as clicking a control or pressing a key. We call such an action an event. The programmer writes code that reacts to an event by performing some functionality. The three steps in creating a Visual Basic program are as follows: 1. Create the interface; that is, generate, position, and size the objects. 2. Set properties; that is, configure the appearance of the objects. 3. Write the code that executes when events occur. Section 2.2 covered Steps 1 and 2; this section is devoted to Step 3. Code consists of statements that carry out tasks. In this section, we limit ourselves to statements that change properties of a control or the form while a program is running. Properties of controls are changed in code with statements of the form controlName.property = setting

where controlName is the name of the control, property is one of the properties of the control, and setting is a valid setting for that property. Such statements are called assignment statements. They assign values to properties. Three examples of assignment statements are as follows: 1. The statement txtBox.Text = "Hello"

displays the word Hello in the text box. 2. The statement btnButton.Visible = True

makes the button visible. 3. The statement txtBox.ForeColor = Color.Red

sets the color of the characters in the text box named txtBox to red. Most events are associated with controls. The event "click on btnButton" is different from the event "click on lstBox.” These two events are specified btnButton.Click and lstBox.Click. The statements to be executed when an event occurs are written in a block of code called an event procedure. The first line of an event procedure (called the header) has the form Private Sub objectName_event(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles objectName.event

Since we do not make any use of the lengthy text inside the parentheses in this book, for the sake of readability we replace it with an ellipsis. However, it will automatically appear in our programs each time Visual Basic creates the header for an event procedure. The structure of an event procedure is

Private Sub objectName_event(...) Handles objectName.event statements End Sub

where the three dots (that is, the ellipsis) represent ByVal sender As System.Object, ByVal e As System.EventArgs

Words such as “Private,” “ByVal,” “As,” “Sub,” “Handles,” and “End” have special meanings in Visual Basic and are referred to as keywords or reserved words. The Visual Basic editor automatically capitalizes the first letter of a keyword and displays the word in blue. The word “Sub” in the first line signals the beginning of the procedure, and the first line identifies the object and the event occurring to that object. The last line signals the termination of the event procedure. The statements to be executed appear between these two lines. (Note: The word “Private” indicates that the event procedure cannot be invoked by another form. This will not concern us until much later in the book. The expression following Handles identifies the object and the event happening to that object. The expression "objectName_event" is the default name of the procedure and can be changed if desired. In this book, we always use the default name. The word “Sub” is an abbreviation of Subroutine.) For instance, the event procedure Private Sub btnButton_Click(...) Handles btnButton.Click txtBox.ForeColor = Color.Red End Sub

changes the color of the words in the text box to red when the button is clicked.

■ An Event Procedure Walkthrough The form in Figure 2.17, which contains two text boxes and a button, will be used to demonstrate what event procedures are and how they are created. Three event procedures will be used to alter the appearance of a phrase appearing in a text box. The event procedures are named txtFirst_TextChanged, btnRed_Click, and txtFirst_Leave.

FIGURE 2.17

OBJECT

PROPERTY

SETTING

frmDemo txtFirst txtSecond btnRed

Text

Demonstration

Text

Change Color to Red

The interface for the event procedure walkthrough.

1. Create the interface in Figure 2.17 in the Form Designer. The Name properties of the form, text boxes, and button should be set as shown in the Object column. The Text property of the form should be set to Demonstration, and the Text property of the button should be set to Change Color to Red. No properties need be set for the text boxes.

2. Click the right mouse button anywhere on the Main area, and click on View Code. The Form Designer IDE is replaced by the Code Editor IDE (also known as the Code view or the Code window). See Figure 2.18. Code Editor tab

Class Name box

FIGURE 2.18

Form Designer tab

Method Name box

The Visual Basic IDE in Code Editor mode.

The tab labeled frmDemo.vb corresponds to the Code Editor. You press the tab labeled frmDemo.vb [Design], when you want to return to the Form Designer. Just below the tabs are two drop-down list boxes. The left box is called the Class Name box, and the right box is called the Method Name box. (When you hover the mouse pointer over one of these list boxes, its type appears in a tooltip.) We will place our program code between the two lines shown. Let’s refer to this region as the program region. Figure 2.18 shows that the Code Editor IDE has a Toolbox, Solution Explorer, and Properties window that support Auto Hide. The Solution Explorer window for the Code Editor functions exactly like the one for the Form Designer. The Code Editor’s Toolbox has just one group, General, that is used to store code fragments which can be copied into a program when needed. The Code Editor’s Properties window will not be used in this textbook. 3. Click on the tab labeled “frmDemo.vb [Design]” to return to the Form Designer. (You also can invoke the Form Designer by clicking Designer in the View menu, or by right-clicking the Code Editor and clicking View Designer.) 4. Double-click on the first text box. The Code Editor reappears, but now the following two lines of code have been added to the program region and the cursor is located on the blank line between them.

Private Sub txtFirst_TextChanged(...) Handles txtFirst.TextChanged End Sub

The first line is the header for the event procedure named txtFirst_TextChanged. This procedure is triggered by the event txtFirst.TextChanged. That is, whenever there is a change in the text displayed in the text box txtFirst, the code between the two lines just shown will be executed. 5. Type the line txtFirst.ForeColor = Color.Blue

at the cursor location. When you type the first period, a list containing all the properties of text boxes appears. See Figure 2.19(a). (Each property is preceded by a little Properties window icon. The list also contains something called methods, which we will discuss later.) At this point, you can scroll up the list and doubleclick on ForeColor to automatically enter that property. See Figure 2.19(b). Or, you can keep typing. After you have typed "For," the list shortens to the single word ForeColor. At that point, you can press the Tab key to enter the word "ForeColor." This feature, known as Member Listing, is one of the features of Visual Basic that use a Microsoft context-sensitive help technology called IntelliSense. txtFirst.

txtFirst.

(a)

(b) FIGURE 2.19

IntelliSence at work.

6. Return to the Form Designer and double-click on the button. The Code window reappears, and the first and last lines of the event procedure btnRed_Click appear in the program region. Type the line that sets the ForeColor property of txtFirst to Red. The event procedure will now appear as follows: Private Sub btnRed_Click(...) Handles btnRed.Click txtFirst.ForeColor = Color.Red End Sub

7. Click on the down-arrow button to the right of the Class Name box and then click on txtFirst in the drop-down list.

8. Click on the down-arrow button to the right of the Method Name box and then click on Leave in the drop-down list box. (The event txtFirst.Leave is triggered when the focus is removed from the text box.) The first and last lines of the event procedure txtFirst_Leave will be displayed. In this procedure, type the line that sets the ForeColor property of txtFirst to Black. The Code Editor will now look as follows: Public Class frmDemo Private Sub txtFirst_Leave(...) Handles txtFirst.Leave txtFirst.ForeColor = Color.Black End Sub Private Sub txtFirst_TextChanged(...) Handles txtFirst.TextChanged txtFirst.ForeColor = Color.Blue End Sub Private Sub btnRed_Click(...) Handles btnRed.Click txtFirst.ForeColor = Color.Red End Sub End Class

9. Hover the cursor over the word "ForeColor." Visual Basic now displays information about the foreground color property. This illustrates another help feature of Visual Basic. 10. Now run the program by pressing F5. 11. Type something into the text box. In Figure 2.20, the blue word "Hello" has been typed. (Recall that a text box has the focus whenever it is ready to accept typing—that is, whenever it contains a blinking cursor.)

FIGURE 2.20

Text box containing input.

12. Click on the second text box. The contents of the first text box will become black. When the second text box was clicked, the first text box lost the focus; that is, the event Leave happened to txtFirst. Thus, the event procedure txtFirst_Leave was invoked, and the code inside the procedure was executed. 13. Click on the button. This invokes the event procedure btnRed_Click, which changes the color of the words in txtFirst to Red.

14. Click on the first text box, and type the word "Friend" after the word "Hello." As soon as typing begins, the text in the text box is changed and the TextChanged event is triggered. This event causes the color of the contents of the text box to become blue. 15. You can repeat Steps 11 through 14 as many times as you like. When you are finished, end the program by pressing Alt+F4, clicking the Stop Debugging icon on the Toolbar, or clicking the Close button (X) on the form.

■ Properties and Event Procedures of the Form You can assign properties to the form itself in code. However, a statement such as frmDemo.Text = "Demonstration"

will not work. The form is referred to by the keyword Me. Therefore, the proper statement is Me.Text = "Demonstration"

To display a list of the events associated with frmDemo, select "(frmDemo Events)" from the Class Name box and then open the Method Name box.

■ The Header of an Event Procedure As mentioned earlier, in a header for an event procedure such as Private Sub btnOne_Click(...) Handles btnOne.Click

btnOne_Click is the name of the event procedure, and btnOne.Click identifies the event that triggers the procedure. The name can be changed at will. For instance, the header can be changed to Private Sub ButtonPushed(...) Handles btnOne.Click

Also, an event procedure can be triggered by more than one event. For instance, if the previous line is changed to Private Sub ButtonPushed(...) Handles btnOne.Click, btnTwo.Click

the event will be triggered if either btnOne or btnTwo is clicked. We have been using ellipses 1 Á 2 as place holders for the phrase ByVal sender As System.Object, ByVal e As System.EventArgs

In Chapter 5, we will gain a better understanding of this type of phrase. Essentially, the word "sender" carries a reference to the object that triggered the event, and the letter "e" carries some additional information that the sending object wants to communicate. We will not make use of either "sender" or "e".

■ Context-Sensitive Help Consider the program created in the event procedure walkthrough. If you click on the word txtFirst and then press the F1 key, a discussion of the Textbox control will appear. This feature of Visual Basic is called context-sensitive help. You can click on any element of a program and press F1 to obtain information relevant to that element. For instance, if you click on Sub and press F1, you will obtain information about procedures.

■ Text Files Programs often make use of data stored in files. In Chapter 10 we work with sophisticated data files called databases. Until then we will rely on simple data files known as text files. Text files can be created, viewed, and modified with sophisticated word processors such as Word, or with elementary word processors such as the Windows accessories WordPad and Notepad. They differ from files normally created with Word in that they have no formatting. They are pure text and nothing else – hence the name text file. The Visual Basic IDE provides simple ways to create and manage text files. The following walkthrough creates a text file: 1. Click on the New Project icon to start a new program. 2. Give a name, such as TextFileProg, to the program, and click on OK. 3. If the Solution Explorer window is not visible, click on Solution Explorer in the View menu. 4. Highlight the name of the program at the top of the Solution Explorer window. 5. Press Alt/P/W to invoke Add New Item from the Project menu. Alternately, click on the Add New Item icon in the Toolbar. (An Add New Item dialog box will appear.) 6. Select the Text File icon in the dialog box, and enter the filename PAYROLL.TXT in the Name text box. (Had we omitted the extension, Visual Basic would have automatically added the extension ".txt" to the name.) 7. Click the Add button. (The Text Editor, an elementary word processor, will appear in the Main area, and the filename will appear in the Solution Explorer window.) 8. Type the following information into the Text Editor. (It gives the names of employees, their hourly wages, and the number of hours worked in the past week. This text file will be used in Section 3.3.) Mike Jones 9.35 35 John Smith 10.75 33

9. Right-click on the tab containing the filename PAYROLL.TXT and click on "Save PAYROLL.TXT." 10. Right-click on the tab containing the filename PAYROLL.TXT and click on Close.

■ Viewing All Possible Event Procedures for a Control All the event procedures associated with the selected control can be displayed and described in the Form Designer Properties window by clicking on the Events button (pictured as a lightning bolt) on the toolbar at the top of the Properties window. Figure 2.21 shows some of the 58 events for the button from the preceding event procedure walkthrough. A description of the selected event appears in the Description pane. Don't be alarmed by the large number of events. There is only one event that is used extensively throughout this book – the Click event for buttons. Note: After viewing events in the Properties window, click on the Properties button (to the left of the Events button) to return to displaying properties in the window.

FIGURE 2.21

Events displayed in the Properties window.

■ Opening a Program Beginning with the next chapter, each example contains a program. These programs can be downloaded from my Prentice-Hall website for this book. See the discussion on page xv for details. The process of placing a program stored on a disk into the Visual Basic environment is referred to as opening the program. Let's open the downloaded program 7-1-3 from Chapter 7. That program allows you to enter the name of a football team, and then displays the numbers of Super Bowls the team won. 1. From Visual Basic, click on Open Project in the File menu. (An Open Project dialog box will appear.) 2. Display the contents of the Ch07 subfolder downloaded from the website. 3. Double-click on 7-1-3. 4. Double-click on 7-1-3.sln. 5. If the Solution Explorer window is not visible, click on Solution Explorer in the View menu.

6. In the Solution Explorer window, click on frmBowl.vb. (Five buttons will appear at the top of the Solution Explorer window. See Figure 2.22.) At any time you can click on the View Code button to invoke the Code Editor, or you can click on the View Designer button to invoke the Form Designer. Show All Files

FIGURE 2.22

Refresh

View Code

View Designer

Solution Explorer window.

7. Press F5 to run the program. 8. Type in the name of a football team (such as 49ers or Giants), and press the "Display Games Team Won" button. (See Figure 2.23.) You can repeat this process as many times as desired.

FIGURE 2.23

Output for program 7-1-3.

9. To end the program, click the Close button (X). The program just executed uses a text file named SBWINNERS.TXT. To view the text file, open the folder bin, open the subfolder Debug, and click on SBWINNERS.TXT.

(If the bin folder is not visible, click on the Show All Files button. If SBWINNERS.TXT is not listed in the Debug subfolder, click the Refresh button and reopen the folders. After reading Chapter 3, you will understand why text files are placed in the Debug subfolder of the bin folder.) The first line of the file gives the winner of the first Super Bowl, the second line gives the winner of the second Super Bowl, and so on. To close the text file, right-click on the tab labeled SBWINNERS.TXT, and click on Close.

■ Comments 1. The Visual Basic editor automatically indents the statements inside procedures. In this book, we indent by two spaces. To instruct your editor to indent by two spaces, select Options from the Tools menu, and uncheck the "Show all settings" box in the Options window that appears. Expand "Text Editor Basic" or "Text Editor," click on "Editor," enter 2 into the "Indent size:" box, and click on OK. 2. The event control.Leave is triggered when the specified control loses the focus. Its counterpart is the event control. Enter which is triggered when the specified control gets the focus. A related statement is control.Focus()

which moves the focus to the specified control. 3. We have ended our programs by clicking the Stop Debugging icon or pressing Alt + F4. A more elegant technique is to create a button, call it btnQuit, with caption Quit and the following event procedure: Private Sub btnQuit_Click(...) Handles btnQuit.Click End End Sub

4. For statements of the form object.Text = setting

the expression for setting must be surrounded by quotes. (For instance, lblName. Text = "Name".) For properties where the proper setting is one of the words True or False, these words should not be surrounded by quotation marks. 5. Names of existing event procedures associated with an object are not automatically changed when you rename the object. You must change them yourself. However, the event that triggers the procedure (and all other references to the control) will change automatically. For example, suppose an event procedure is Private Sub btnOne_Click(...) Handles btnOne.Click btnOne.Text = "Press Me" End Sub

and, in the Form Designer, you change the name of btnOne to btnTwo. Then, when you return to the Code Editor the procedure will be Private Sub btnOne_Click(...) Handles btnTwo.Click btnTwo.Text = "Press Me" End Sub

6. The Code Editor has many features of a word processor. For instance, the operations cut, copy, paste, undo, and redo can be carried out with the sixth through ninth icons from the Toolbar. These operations, and several others, also can be initiated from the Edit menu. 7. The Code Editor can detect certain types of errors. For instance, if you type txtFirst.Text = hello

and then move away from the line, the automatic syntax checker will underline the word "hello" with a blue squiggle to indicate that something is wrong. When the mouse cursor is hovered over the offending wording, the editor will display a message explaining what is wrong. If you try to run the program without correcting the error, the dialog box in Figure 2.24 will appear.

FIGURE 2.24

Error dialog box.

8. When you double-click on a control in the Form Designer, the header for the most used event procedure is placed in the Code Editor. The event that appears most frequently in this book is the Click event for button controls. 9. Font properties, such as the name, style, and size, are usually specified at design time. The setting of the properties can be displayed in code with statements such as 1stBox.Items.Add(txtBox.Font.Name) 1stBox.Items.Add(txtBox.Font.Bold) 1stBox.Items.Add(txtBox.Font.Size)

However, a font's name, style, and size properties cannot be altered in code with statements of the form txtBox.Font.Name = "Courier New" txtBox.Font.Bold = True txtBox.Font.Size = 16

10. When you make changes to a program, asterisks appear as superscripts on the tabs labeled "frmName.vb [design]" and "frmName.vb." The asterisks disappear when the program is saved or run. Note: If the program has been saved to disk, all files for the program will be automatically updated on the disk whenever the program is saved or run.

Practice Problem 2.3 1. What event procedure is displayed when you double-click on each of the following controls in the Form Designer? (a) text box (b) button (c) label (d) list box 2. Give a statement that will prevent the user from typing into txtBox. EXERCISES 2.3

In Exercises 1 through 6, describe the contents of the text box after the button is clicked. 1. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = "Hello" End Sub

2. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.ForeColor = Color.Red txtBox.Text = "Hello" End Sub

3. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.BackColor = Color.Orange txtBox.Text = "Hello" End Sub

4. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = "Goodbye" txtBox.Text = "Hello" End Sub

5. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = "Hello" txtBox.Visible = False End Sub

6. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.BackColor = Color.Yellow txtBox.Text = "Hello" End Sub

In Exercises 7 through 10, assume that the three objects on the form were created in the order txtFirst, txtSecond, and lblOne. Determine the output displayed in lblOne when the program is run and the Tab key is pressed. Note: Initially, txtFirst has the focus. 7. Private Sub txtFirst_Leave(...) Handles txtFirst.Leave lblOne.ForeColor = Color.Green lblOne.Text = "Hello" End Sub

◆

◆

8. Private Sub txtFirst_Leave(...) Handles txtFirst.Leave lblOne.BackColor = Color.White lblOne.Text = "Hello" End Sub

9. Private Sub txtSecond_Enter(...) Handles txtSecond.Enter lblOne.BackColor = Color.Gold lblOne.Text = "Hello" End Sub

10. Private Sub txtSecond_Enter(...) Handles txtSecond.Enter lblOne.Visible = False lblOne.Text = "Hello" End Sub

In Exercises 11 through 16, determine the errors. 11. Private Sub btnOutput_Click(...) Handles btnOutput.Click Form1.Text = "Hello" End Sub

12. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Text = Hello End Sub

13. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtFirst.ForeColor = Red End Sub

14. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox = "Hello" End Sub

15. Private Sub btnOutput_Click(...) Handles btnOutput.Click txtBox.Font.Size = 20 End Sub

16. Private Sub btnOutput_Click(...) Handles btn1.Click, btn2.Click Me.Color = Color.Yellow End Sub

In Exercises 17 through 28, write a line (or lines) of code to carry out the task. 17. 18. 19. 20.

Display “E.T. phone home.” in lblTwo. Display “Play it, Sam.” in lblTwo. Display “The stuff that dreams are made of.” in red letters in txtBox. Display “Life is like a box of chocolates.” in txtBox with blue letters on a gold background. 21. Disable txtBox. 22. Change the words in the form’s title bar to “Hello World.” 23. Make lblTwo disappear.

24. 25. 26. 27. 28. 29. 30. 31.

Change the color of the letters in lblName to red. Enable the disabled button btnOutcome. Give the focus to btnCompute. Give the focus to txtBoxTwo. Change the background color of the form to White. Describe the Enter event in your own words. Describe the Leave event in your own words. The label control has an event called DoubleClick that responds to a doubleclicking of the left mouse button. Write a simple program to test this event. Determine whether you can trigger the DoubleClick event without also triggering the Click event. 32. Write a simple program to demonstrate that a button’s Click event is triggered when you press the Enter key while the button has the focus.

In Exercises 33 through 38, the interface and initial properties are specified. Write the program to carry out the stated task. 33. When one of the three buttons is pressed, the words on the button are displayed in the text box with the stated alignment. Note: Rely on IntelliSense to provide you with the proper settings for the TextAlign property. OBJECT

PROPERTY

SETTING

frmAlign txtBox btnLeft btnCenter btnRight

Text ReadOnly Text Text Text

Text Alignment True Left Justify Center Right Justify

34. When one of the buttons is pressed, the face changes to a smiling face (Wingdings character “J”) or a frowning face (Wingdings character “L”). OBJECT

PROPERTY

SETTING

frmFace lblFace

Text Font Name Font Size Text Text Text

Face Wingdings 24 K Smile Frown

btnSmile btnFrown

◆

◆

35. Pressing the buttons alters the background and foreground colors in the text box. OBJECT

PROPERTY

SETTING

frmColors lblBack btnRed btnBlue txtBox

Text Text Text Text Text TextAlign Text Text Text

Colorful Text Background Red Blue Beautiful Day Center Foreground White Yellow

lblFore btnWhite btnYellow

36. When one of the three text boxes receives the focus, its text becomes red. When it loses the focus, the text returns to black. The buttons set the alignment in the text boxes to Left or Right. Note: Rely on IntelliSense to provide you with the proper settings for the TextAlign property. OBJECT

PROPERTY

SETTING

frm123 txtOne txtTwo txtThree btnLeft btnRight

Text Text Text Text Text Text

One, Two, Three One Two Three Left Right

37. When the user moves the focus to one of the three small text boxes at the bottom of the form, an appropriate saying is displayed in the large text box. Use the sayings “I like life, it’s something to do.”; “The future isn’t what it used to be.”; and “Tell the truth and run.” OBJECT

PROPERTY

SETTING

frmQuote txtQuote txtLife txtFuture txtTruth

Text ReadOnly Text Text Text

Sayings True Life Future Truth

2.3

Visual Basic Events

38. The user can disable or enable the text box by clicking on the appropriate button. After the user clicks the Enable button, the text box should receive the focus. OBJECT

PROPERTY

SETTING

frmTextBox txtBox btnDisable btnEnable

Text

Text Box

Text Text

Disable Text Box Enable Text Box

In Exercises 39 through 44, write a program with a Windows-style interface to carry out the task. 39. The form contains four square buttons arranged in a rectangular array. Each button has the caption “Push Me.” When the user clicks on a button, the button disappears and the other three become or remain visible. 40. A form contains two text boxes and one large label between them with no preset caption. When the first text box receives the focus, the label reads “Enter your full name.” When the second text box receives the focus, the label reads “Enter your phone number, including area code.” 41. Use the same form and properties as in Exercise 34, with the captions for the buttons replaced with Vanish and Reappear. Clicking a button should produce the stated result. 42. Simulate a traffic light with three small square text boxes placed vertically on a form. Initially, the bottom text box is solid green and the other text boxes are dark gray. When the Tab key is pressed, the middle text box turns yellow and the bottom text box turns dark gray. The next time Tab is pressed, the top text box turns red and the middle text box turns dark gray. Subsequent pressing of the Tab key cycles through the three colors. Hint: First place the bottom text box on the form, then the middle text box, and finally the top text box. 43. The form contains a single read-only text box and two buttons. When the user clicks on one of the buttons, the sentence “You just clicked on a button.” is displayed in the text box. The program should consist of a single event procedure. 44. The form contains two text boxes into which the user types information. When the user clicks on one of the text boxes, it becomes blank and its contents are displayed in the other text box. Note: A text box can be cleared with the statement txtBox.Clear() or the statement txtBox.Text = "".

◆

Solutions to Practice Problem 2.3 1. (a) (b) (c) (d)

TextChanged Click Click SelectedIndexChanged

2. Three possibilities are txtBox.Enabled = False txtBox.ReadOnly = True txtBox.Visible = False

CHAPTER 2 SUMMARY 1. The Visual Basic Form Designer displays a form that can hold a collection of controls for which various properties can be set. Some examples of controls are text boxes, labels, buttons, and list boxes. Some useful properties are Text (sets the text displayed in a control), Name (used to give a meaningful name to a control), Font.Name (selects the name of the font used), Font.Size (sets the size of the text displayed), Font.Bold (displays boldface text), Font.Italic (displays italics text), BackColor (sets the background color), ForeColor (sets the color of the text), ReadOnly (determines whether text can be typed into a text box when the program is running), TextAlign (sets the type of alignment for the text in a control), Enabled (determines whether a control can respond to user interaction), and Visible (determines whether an object can be seen or is hidden). 2. An event procedure is executed when something happens to a specified object. Some events are object.Click (object is clicked), object.TextChanged (a change occurred in the value of the object's Text property), object.Leave (object loses the focus), and object.Enter (object receives the focus). Note: The statement object.Focus() moves the focus to the specified object. 3. Text files, which can be created and managed in the Visual Basic IDE, provide a convenient device for holding data to be accessed by a program. 4. Programming assistance can be accessed through the Help menu, or via IntelliSense and context-sensitive help. 5. Tab order, the order in which the user moves the focus from one control to another by pressing the Tab key while the program is running, can be set from the View menu.

3 Variables, Input, and Output 3.1

3.1

Numbers

Much of the data processed by computers consists of numbers. In computerese, numbers are called numeric literals. This section discusses the operations that are performed with numbers and the ways numbers are displayed.

■ Arithmetic Operations The five standard arithmetic operations in Visual Basic are addition, subtraction, multiplication, division, and exponentiation. Addition, subtraction, and division are denoted in Visual Basic by the standard symbols +, - , and /, respectively. However, the notations for multiplication and exponentiation differ from the customary mathematical notations as follows: Mathematical Notation a # b or a * b ar

Visual Basic Notation a*b a–r

(The asterisk [*] is the upper character of the 8 key. The caret [¿] is the upper character of the 6 key.) One way to show a number on the screen is to display it in a list box. If n is a number, then the instruction lstBox.Items.Add(n)

displays the number n as the last item in the list box. Add is called a method. (Generally, a method is a process that performs a task for a particular object.) If the parentheses contain a combination of numbers and arithmetic operations, the Add method carries out the operations and displays the result. Another important method is Clear. The statement lstBox.Items.Clear()

erases all the items displayed in the list box lstBox. Example 1 The following program applies each of the five arithmetic operations. Preceding the program is the form design and a table showing the names of the objects on the form and the settings, if any, for properties of these objects. This form design is also used in the discussion and examples in the remainder of this section. The word "Run" in the phrasing [Run Á] indicates that F5 should be pressed to execute the program. Notice that in the output 3 / 2 is displayed in decimal form. Visual Basic never displays numbers as common fractions. In the evaluation of 2 * 13 + 42, the operation inside the parentheses is calculated first. Note: All programs appearing in examples and case studies are provided on the companion website for this book. See the discussion on page xv for details.

OBJECT

PROPERTY

SETTING

frmArithmetic lstResults btnCompute

Text

3-1-1

Text

Compute

Private Sub btnCompute_Click(...) Handles btnCompute.Click lstResults.Items.Clear() lstResults.Items.Add(3 + 2) lstResults.Items.Add(3 - 2) lstResults.Items.Add(3 * 2) lstResults.Items.Add(3 / 2) lstResults.Items.Add(3 ^ 2) lstResults.Items.Add(2 * (3 + 4)) End Sub

[Run, and then click the button.]

■ Variables In applied mathematics problems, quantities are referred to by names. For instance, consider the following high school algebra problem: “If a car travels at 50 miles per hour, how far will it travel in 14 hours? Also, how many hours are required to travel 410 miles?” The solution to this problem uses the well-known formula distance = speed * time elapsed Here’s how this problem would be solved with a computer program: Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim speed As Double

Chapter 3 Variables, Input, and Output Dim timeElapsed As Double Dim distance As Double lstResults.Items.Clear() speed = 50 timeElapsed = 14 distance = speed * timeElapsed lstResults.Items.Add(distance) distance = 410 timeElapsed = distance / speed lstResults.Items.Add(timeElapsed) End Sub

[Run, and then click the button. The following is displayed in the list box.] 700 8.2

Skip the second, third, and fourth lines of the event procedure for now. We will return to them soon. The sixth line sets the speed to 50, and the seventh line sets the time elapsed to 14. The eighth line multiplies the value for the speed by the value for the time elapsed and sets the distance to this product. The next line displays the answer to the distance-traveled question. The three lines before the End Sub statement answer the time-required question in a similar manner. The names speed, timeElapsed, and distance, which hold values, are referred to as variables. Consider the variable timeElapsed. In the seventh line, its value was set to 14. In the eleventh line, its value was changed as the result of a computation. On the other hand, the variable speed had the same value, 50, throughout the program. In general, a variable is a name that is used to refer to an item of data. The value assigned to the variable may change during the execution of the program. In Visual Basic, variable names can be up to 16,383 characters long, must begin with a letter or an underscore, and can consist only of letters, digits, and underscores. (The shortest variable names consist of a single letter.) Visual Basic does not distinguish between uppercase and lowercase letters used in variable names. Some examples of variable names are total, numberOfCars, taxRate_2008, and n. As a convention, we write variable names in lowercase letters except for the first letters of additional words (as in gradeOnFirstExam). This convention is called camel casing. If var is a variable and n is a literal, then the statement var = n

assigns the number n to the variable var. (Such a statement is another example of an assignment statement.) A variable is declared to be of a certain type depending on the sort of data that can be assigned to it. The most versatile type for holding numbers is called Double. A variable of type Double can hold whole, fractional, or mixed numbers between about -1.8 # 10308 and 1.8 # 10308. Dim statements (also called declaration statements) declare the names and types of the variables to be used in the program. The second, third, and fourth lines of this event procedure declare three variables of type Double and give them the names speed, timeElapsed, and distance.

In general, a statement of the form Dim varName As Double

declares a variable named varName to be of type Double. Actually, the Dim statement causes the computer to set aside a location in memory with the name varName. Since varName is a numeric variable, the Dim statement initially places the number zero in that memory location. (We say that zero is the initial value or default value of the variable.) Each subsequent assignment statement having varName to the left of the equal sign will change the value of the number. The initial value can be set to a value other than zero. To specify a nonzero initial value, follow the declaration statement with an equal sign followed by the initial value. The statement Dim varName As Double = 50

declares the specified variable as a variable of type Double and gives it the initial value 50. The statement lstBox.Items.Add(varName)

looks into this memory location for the current value of the variable and displays that value in the list box. A combination of literals, variables, and arithmetic operations that can be evaluated to yield a number is called a numeric expression. Expressions are evaluated by replacing each variable by its value and carrying out the arithmetic. Some examples of expressions are 2 * distance + 7, n + 1, and 1a + b2>3. Example 2 The following program displays the default value of a variable and the value of an expression: Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim a As Double Dim b As Double = 3 lstResults.Items.Clear() lstResults.Items.Add(a) lstResults.Items.Add(b) a = 5 lstResults.Items.Add(a * (2 + b)) End Sub

[Run, and then click the button. The following is displayed in the list box.] 0 3 25

If var is a variable, then the assignment statement var = expression first evaluates the expression on the right and then assigns its value to the variable on the left. For instance, the event procedure in Example 2 can be written as Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim a As Double Dim b As Double Dim c As Double lstResults.Items.Clear() a = 5 b = 3 c = a * (2 + b) lstResults.Items.Add(c) End Sub

The expression a * 12 + b2 is evaluated to 25, and then this value is assigned to the variable c.

■ Incrementing the Value of a Variable Because the expression on the right side of an assignment statement is evaluated before an assignment is made, a statement such as var = var + 1

is meaningful. It first evaluates the expression on the right (that is, it adds 1 to the original value of the variable var) and then assigns this sum to the variable var. The effect is to increase the value of the variable var by 1. In terms of memory locations, the statement retrieves the value of var from var's memory location, uses it to compute var + 1, and then places the sum back into var's memory location. This type of calculation is so common that Visual Basic provides a special operator to carry it out. The statement var = var + 1 can be replaced with the statement var += 1

In general, if n has a numeric value, then the statement var += n

adds n to the value of var.

■ Built-In Functions: Math.Sqrt, Int, Math.Round There are several common operations that we often perform on numbers other than the standard arithmetic operations. For instance, we may take the square root of a number or round a number. These operations are performed by built-in functions. Functions associate with one or more values called the input, and a single value called the output. The function is said to return the output value. The three functions considered here have numeric input and output. The function Math.Sqrt calculates the square root of a number. The function Int finds the greatest integer less than or equal to a number. Therefore, Int discards the decimal part of positive numbers. The value of Math.Round(n, r) is the number n rounded to r decimal places. The parameter r can be omitted. If so, n is rounded to a whole number. Some examples follow:

Math.Sqrt192 is 3. Math.Sqrt102 is 0. Math.Sqrt122 is 1.414214.

Int12.72 is 2. Int132 is 3. Int1 -2.72 is - 3.

Math.Round12.72 is 3. Math.Round12.317, 22 is 2.32. Math.Round12.317, 12 is 2.3.

The terms inside the parentheses can be numbers (as shown), numeric variables, or numeric expressions. Expressions are first evaluated to produce the input. Example 3 The following program evaluates each of the functions for a specific input given by the value of the variable n: Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim n As Double Dim root As Double n = 6.76 root = Math.Sqrt(n) lstResults.Items.Clear() lstResults.Items.Add(root) lstResults.Items.Add(Int(n)) lstResults.Items.Add(Math.Round(n, 1)) End Sub

[Run, and then click the Compute button. The following is displayed in the list box.] 2.6 6 6.8

Example 4 an expression:

The following program evaluates each of the preceding functions at

Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim a As Double Dim b As Double a = 2 b = 3 lstResults.Items.Clear() lstResults.Items.Add(Math.Sqrt(5 * b + 1)) lstResults.Items.Add(Int(a ^ b + 0.8)) lstResults.Items.Add(Math.Round(a / b, 3)) End Sub

[Run, and then click the button. The following is displayed in the list box.] 4 8 0.667

■ The Integer Data Type In this text, we sometimes need to use variables of type Integer. An Integer variable is declared with a statement of the form Dim varName As Integer

and can be assigned only whole numbers from about - 2 billion to 2 billion. Integer variables are used primarily for counting.

■ Multiple Declarations Several variables of the same type can be declared with a single Dim statement. For instance, the two Dim statements in Example 2 can be replaced by the single statement Dim a, b As Double

Two other types of multiple-declaration statement are Dim a As Double, b As Integer Dim c As Double = 2, b As Integer = 5

■ Parentheses Parentheses cannot be used to indicate multiplication, as is commonly done in algebra. For instance, the expression x1y + z2 is not valid. It must be written as x * 1y + z2. Parentheses should be used when necessary to clarify the meaning of an expression. When there are no parentheses, the arithmetic operations are performed in the following order: (1) exponentiations; (2) multiplications and divisions; (3) additions and subtractions. In the event of ties, the leftmost operation is carried out first. Table 3.1 summarizes these rules. Note: If you use parentheses liberally, you will not have to rely on the precedence table for arithmetic operations. For instance, write (2 * 3) + 4 instead of 2 * 3 + 4. Write (2 ^ 3) + 4 instead of 2 ^ 3 + 4.

TABLE 3.1

Level of precedence for arithmetic operations.

()	Inner to outer, left to right
¿	Left to right in expression
* /	Left to right in expression
+-	Left to right in expression

■ Three Types of Errors Grammatical errors, such as misspellings, omissions, or incorrect punctuations, are called syntax errors. Most syntax errors are spotted by the Code Editor when they are entered. The editor underlines the syntax error with a blue squiggly line and displays a description of the error when the mouse cursor is hovered over the squiggly line. Some incorrect statements and their errors are as follows: Statement lstBox.Itms.Add(3) lstBox.Items.Add(2+ Dim m; n As Integer

Reason for Error The word Items is misspelled. The number following the plus sign is missing. The semicolon should be a comma.

Errors that occur while a program is running are called run-time errors or exceptions. They usually result from the inability of the computer to carry out the intended task. For instance, if the file DATA.TXT is not in the root folder of the C drive, then a statement that refers to the file by the filespec "C:\DATA.TXT" will cause the program to stop executing and produce a message box with the title FileNotFoundException was unhandled.

Also, a yellow arrow will appear at the left side of the line of code that caused the error. At that point, you should end the program. A third type of error is the so-called logical error. Such an error occurs when a program does not perform the way it was intended. For instance, the line average = firstNum + secondNum / 2

is syntactically correct. However, the missing parentheses in the line of code are responsible for an incorrect value being generated. Appendix D discusses debugging tools that can be used to detect and correct logical errors.

■ The Error List Window Syntax errors are not only indicated in the Code Editor, but also are listed in the Error List window. (Note: If the Error List window is not visible in the IDE, click on Error List in the View menu.) Example 5

The following program contains three errors.

Private Sub btnCompute_Click(...) Handles btnCompute.Click Dim m; n As Double lstResults.Items.Add(5 lstResults.Items.Add(a) End Sub

[Press Alt/V/I to display the Error List window.]

The Private Sub statement is in line 3. (Line 1 contains the Public Class statement and line 2 is a blank line.) Click on one of the three errors in the Error List window and then press F1 to see a discussion of that error.

◆

■ Comments 1. Declaring variables at the beginning of each event procedure is regarded as good programming practice because it makes programs easier to read and helps prevent certain types of errors. 2. Keywords (reserved words) cannot be used as names of variables. For instance, the statements Dim private as Double and Dim sub As Double are not valid. To obtain a complete list of Visual Basic keywords, look up reserved keywords in Help's Index. 3. Names given to variables are sometimes referred to as identifiers. 4. In math courses, literals are referred to as constants. However, the word "constant" has a special meaning in programming languages. 5. Numeric literals used in expressions or assigned to variables must not contain commas, dollar signs, or percent signs. Also, mixed numbers, such as 8 1/2, are not allowed. 6. Although requesting the square root of a negative number does not terminate the execution of the program, it can produce unexpected results. For instance, the statement lstBox.Items.Add(Math.Sqrt(-1))

displays NaN. Note: NaN is an abbreviation for "Not a Number." 7. If the value of numVar is 0 and numVar has type Double, then the statements numVarInv = 1 / numVar lstBox.Items.Add(numVarInv) lstBox.Items.Add(1 / numVarInv)

cause the following items to be displayed in the list box: Infinity 0

8. When n is halfway between two successive whole numbers (such as 1.5, 2.5, 3.5, and 4.5), then it rounds to the nearest even number. For instance, Math.Round(2.5) is 2 and Math.Round(3.5) is 4. 9. In addition to the five arithmetic operators discussed at the beginning of this section, the Mod operator and the integer division operator (\) are two other useful operators. Let m and n be positive whole numbers. When you use long division to divide m by n, you obtain an integer result and an integer remainder. The result is m \ n and the remainder is m Mod n. For instance, 14 \ 3 is 4 and 14 Mod 3 is 2 19 \ 5 is 3 and 19 Mod 5 is 4 10 \ 2 is 5 and 10 Mod 2 is 0. 10. In scientific notation, numbers are written in the form b # 10r, where b is a number of magnitude from 1 up to (but not including) 10, and r is an integer. Visual Basic displays very large numbers in scientific notation where b # 10r is written as bEr. (The letter E is an abbreviation for exponent.) For instance, when the statement lstBox.Items.Add(123 * 10 ^ 15) is executed, 1.23E+17 is displayed in the list box.

11. When you first enter a statement such as Dim n As Double, a squiggle will appear under the variable name and the Error List window will record a warning. The squiggle merely indicates that the variable has not yet been assigned a value. If the squiggle is still present after the entire event procedure has been entered, this will tell you that the variable was never used and that the declaration statement should be removed. Practice Problems 3.1 1. Evaluate 2 + 3 * 4. 2. Explain the difference between the assignment statement var1 = var2

and the assignment statement var2 = var1

3. Complete the table by filling in the value of each variable after each line is executed. a Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim a, b, c As Double 0 a = 3 3 b = 4 3 c = a + b a = c * a lstResults.Items.Add(a - b) b = b * b End Sub

b

c

0 0 4

0 0 0

4. Write a statement that increases the value of the numeric variable var by 5%. EXERCISES 3.1

In Exercises 1 through 6, evaluate the numeric expression without the computer, and then use Visual Basic to check your answer. 1. 3 * 4 3. 1>12¿32 5. 15 - 32 * 4

2. 7¿2 4. 3 + 14 * 52 6. 3 * 11 - 22¿52

In Exercises 7 through 10, evaluate the expression. 7. 7\3 9. 7 Mod 3

8. 14 Mod 4 10. 14\4

In Exercises 11 through 16, determine whether the name is a valid variable name. 11. sales.2008 13. fOrM_1040 15. expenses?

12. room&Board 14. 1040B 16. INCOME 2008

In Exercises 17 through 22, evaluate the numeric expression where a = 2, b = 3, and c = 4. 17. 1a * b2 + c 19. 11 + b2 * c 21. b¿1c - a2

18. a * 1b + c2 20. a¿c 22. 1c - a2¿b

In Exercises 23 through 28, write an event procedure to calculate and display the value of the expression. 23. 7 # 8 + 5 25. 5.5% of 20 27. 1713 + 1622

24. 11 + 2 # 923 26. 15 - 312 + 342 28. 4 1/2 - 3 5/8

In Exercises 29 and 30, complete the table by filling in the value of each variable after each line is executed. 29. x

y

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim x, y As Double x = 2 y = 3 * x x = y + 5 lstResults.Items.Clear() lstResults.Items.Add(x + 4) y = y + 1 End Sub

30. bal

inter

withDr

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim bal, inter, withDr As Double bal = 100 inter = 0.05 withDr = 25 bal += inter * bal bal = bal - withDr End Sub

In Exercises 31 through 38, determine the output displayed in the list box by the lines of code. 31. Dim amount As Double amount = 10 lstOutput.Items.Add(amount - 4)

32. Dim a, b As Integer a = 4

3.1

b = 5 * a lstOutput.Items.Add(a + b)

33. Dim n As Integer = 7 n += 1 lstOutput.Items.Add(1) lstOutput.Items.Add(n) lstOutput.Items.Add(n + 1)

34. Dim num As Integer = 5 num = 2 * num lstOutput.Items.Add(num)

35. Dim a, b As Integer lstOutput.Items.Add(a + 1) a = 4 b = a * a lstOutput.Items.Add(a * b)

36. Dim tax As Double tax = 200 tax = 25 + tax lstOutput.Items.Add(tax)

37. Dim x As Double = 3 x += 2 lstOutput.Items.Add(x * x) lstOutput.Items.Add(x + 3 * x)

38. Dim n As Double = 2, m As Double = 5 lstOutput.Items.Add(3 * n) n += n lstOutput.Items.Add(n + m) lstOutput.Items.Add(n - m)

In Exercises 39 through 44, identify the errors. 39. Dim a, b, c As Double a = 2 b = 3 a + b = c lstOutput.Items.Add(c)

40. Dim a, b, c, d As Double a = 2 b = 3 c = d = 4 lstOutput.Items.Add(5((a + b) / (c + d)

41. Dim balance, deposit As Double balance = 1,234 deposit = $100 lstOutput.Items.Add(balance + deposit)

42. Dim interest, balance As Double 0.05 = interest balance = 800 lstOutput.Items.Add(interest * balance)

43. Dim 9W As Double 9W = 2 * 9W lstOutput.Items.Add(9W)

44. Dim n As Double = 1.2345 lstOutput.Items.Add(Round(n, 2))

In Exercises 45 and 46, rewrite the code using one line. 45. Dim quantity As Integer quantity = 12

46. Dim m As Integer Dim n As Double m = 2 n = 3

In Exercises 47 through 52, find the value of the given function. 47. Int(10.75) 50. Math.Sqrt(64)

48. Int19 - 22 51. Math.Round(3.1279, 3)

49. Math.Sqrt13 * 122 52. Math.Round1- 2.62

In Exercises 53 through 58, find the value of the given function where a and b are numeric variables of type Double, a = 5 and b = 3. 53. Int1-a > 22 56. Math.Sqrt14 + a2

54. Math.Round(a / b) 57. Math.Round1a + .52

55. Math.Sqrt1a - 52 58. Int1b * 0 .52

In Exercises 59 through 66, write an event procedure starting with a Private Sub btnCompute_Click1 Á 2 Handles btnCompute.Click statement, ending with an End Sub statement, and having one line for each step. Lines that display data should use the given variable names. 59. The following steps calculate a company's profit: (a) Declare all variables. (b) Assign the value 98456 to the variable revenue. (c) Assign the value 45000 to the variable costs. (d) Assign the difference between the variables revenue and costs to the variable profit. (e) Display the value of the variable profit in a list box. 60. The following steps calculate the amount of a stock purchase: (a) Declare all variables. (b) Assign the value 25.625 to the variable costPerShare. (c) Assign the value 400 to the variable numberOfShares. (d) Assign the product of costPerShare and numberOfShares to the variable amount. (e) Display the value of the variable amount in a list box.

61. The following steps calculate the price of an item after a 30% reduction: (a) Declare all variables. (b) Assign the value 19.95 to the variable price. (c) Assign the value 30 to the variable discountPercent. (d) Assign the value of (discountPercent divided by 100) times price to the variable markDown. (e) Decrease price by markdown. (f) Display the value of price in a list box. 62. The following steps calculate a company's break-even point, the number of units of goods the company must manufacture and sell in order to break even: (a) Declare all variables. (b) Assign the value 5000 to the variable fixedCosts. (c) Assign the value 8 to the variable pricePerUnit. (d) Assign the value 6 to the variable costPerUnit. (e) Assign the value fixedCosts divided by (the difference of pricePerUnit and costPerUnit) to the variable breakEvenPoint. (f) Display the value of the variable breakEvenPoint in a list box. 63. The following steps calculate the balance after three years when $100 is deposited in a savings account at 5% interest compounded annually: (a) Declare all variables. (b) Assign the value 100 to the variable balance. (c) Increase the variable balance by 5% of its value. (d) Increase the variable balance by 5% of its value. (e) Increase the variable balance by 5% of its value. (f) Display the value of the variable balance in a list box. 64. The following steps calculate the balance at the end of three years when $100 is deposited at the beginning of each year in a savings account at 5% interest compounded annually: (a) Declare all variables. (b) Assign the value 100 to the variable balance. (c) Increase the variable balance by 5% of its value, and add 100. (d) Increase the variable balance by 5% of its value, and add 100. (e) Increase the variable balance by 5% of its value. (f) Display the value of the variable balance in a list box. 65. The following steps calculate the balance after 10 years when $100 is deposited in a savings account at 5% interest compounded annually: (a) Declare all variables. (b) Assign the value 100 to the variable balance. (c) Multiply the variable balance by 1.05 raised to the 10th power. (d) Display the value of the variable balance in a list box. 66. The following steps calculate the percentage profit from the sale of a stock: (a) Declare all variables. (b) Assign the value 10 to the variable purchasePrice.

(c) Assign the value 15 to the variable sellingPrice. (d) Assign, to the variable percentProfit, 100 times the value of the difference between sellingPrice and purchasePrice divided by purchasePrice. (e) Display the value of the variable percentProfit in a list box. In Exercises 67 through 72, write an event procedure to solve the problem and display the answer in a list box. The program should use variables for each of the quantities. 67. Suppose each acre of farmland produces 18 tons of corn. How many tons of corn can be produced on a 30-acre farm? 68. Suppose a ball is thrown straight up in the air with an initial velocity of 50 feet per second and an initial height of 5 feet. How high will the ball be after 3 seconds? Note: The height after t seconds is given by the expression -16t2 + v0t + h0, where v0 is the initial velocity and h0 is the initial height. 69. If a car left Washington, D.C., at 2 o'clock and arrived in New York at 7 o'clock, what was its average speed? Note: New York is 233 miles from Washington. 70. A motorist wants to determine her gas mileage. At 23,352 miles (on the odometer), the tank is filled. At 23,695 miles the tank is filled again with 14 gallons. How many miles per gallon did the car average between the two fillings? 71. A U.S. geological survey showed that Americans use an average of 1600 gallons of water per person per day, including industrial use. How many gallons of water are used each year in the United States? Note: The current population of the United States is about 304 million people. 72. According to FHA specifications, each room in a house should have a window area equal to at least 10 percent of the floor area of the room. What is the minimum window area for a 14-ft by 16-ft room? Solutions to Practice Problems 3.1 1. 24. Multiplications are performed before additions. If the intent is for the addition to be performed first, the expression should be written 12 + 32 * 4. 2. The first assignment statement assigns the value of the variable var2 to the variable var1, whereas the second assignment statement assigns var1's value to var2. 3.

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click Dim a, b, c As Double a = 3 b = 4 c = a + b a = c * a lstResults.Items.Add(a - b) b = b * b End Sub

a

b

c

0 3 3 3 21 21 21

0 0 4 4 4 4 16

0 0 0 7 7 7 7

3.2

Strings

Each time an assignment statement is executed, only one variable (the variable to the left of the equal sign) has its value changed. 4. Each of the three following statements increases the value of var by 5%. var = var + 0.05 * var var = 1.05 * var var += 0.05 * var

3.2

Strings

The most common types of data processed by Visual Basic are numbers and strings. Sentences, phrases, words, letters of the alphabet, names, telephone numbers, addresses, and Social Security numbers are all examples of strings. Formally, a string literal is a sequence of characters that is treated as a single item. String literals can be assigned to variables, displayed in text boxes and list boxes, and combined by an operation called concatenation (denoted by &).

■ Variables and Strings A string variable is a name used to refer to a string. The allowable names of string variables are the same as those of numeric variables. The value of a string variable is assigned or altered with assignment statements and displayed in a list box like the value of a numeric variable. String variables are declared with statements of the form Dim varName As String

Example 1 The following code shows how assignment statements and the Add method are used with strings. The string variable president is assigned a value by the third line and this value is displayed by the sixth line. The quotation marks surrounding each string literal are not part of the literal and are not displayed by the Add method. (The form for this example contains a button and a list box.) Private Sub btnDisplay_Click(...) Handles btnDisplay.Click Dim president As String president = "George Washington" lstOutput.Items.Clear() lstOutput.Items.Add("president") lstOutput.Items.Add(president) End Sub

[Run, and then click the button. The following is displayed in the list box.] president George Washington

If x, y, Á , z are characters and strVar is a string variable, then the statement strVar = "xy...z"

assigns the string literal xy Á z to the variable and the statement

◆

◆

Chapter 3 Variables, Input, and Output lstBox.Items.Add("xy...z")

or lstBox.Items.Add(strVar)

displays the string xy Á z in a list box. If strVar2 is another string variable, then the statement strVar2 = strVar

assigns the value of the variable strVar to the variable strVar2. (The value of strVar will remain the same.) String literals used in assignment or lstBox.Items.Add statements must be surrounded by quotation marks, but string variables are never surrounded by quotation marks.

■ Option Explicit and Option Strict Option Explicit and Option Strict are two options that affect programming. Throughout this book, we assume that both options are in effect. Having these two options working is considered good programming practice. Option Explicit requires that all variables be declared with Dim statements. The absence of this option can lead to errors resulting from the misspelling of names of variables. Option Strict requires explicit conversions in most cases where a value or variable of one type is assigned to a variable of another type. The absence of this option can lead to data loss. Visual Basic provides a way to enforce Option Explicit and Option Strict for all programs you create. Press Alt/ Tools/Options to open the Options dialog box. In the left pane, click on the symbol (+ or 饗) to the left of Projects and Solutions to expand that entry. Then click on the subentry VB Defaults. Four default project settings will appear on the right. (See Figure 3.1.) If the settings for Option Explicit and Object Strict are not already set to On, change them to On. Note: Option Infer is discussed in Chapter 6.

FIGURE 3.1

Option default project settings.

3.2

Strings

■ Using Text Boxes for Input and Output The content of a text box is always a string. Therefore, statements such as strVar = txtBox.Text

and txtBox.Text = strVar

can be used to assign the contents of the text box to the string variable strVar and vice versa. Numbers typed into text boxes are stored as strings. With Option Strict set to On, such strings must be converted to numeric values before they can be assigned to numeric variables or used in numeric expressions. The functions CDbl and CInt convert strings representing numbers into numbers of type Double and Integer, respectively. Going in the other direction, the function CStr converts a number into a string representation of the number. Therefore, statements such as dblVar = CDbl(txtBox.Text)

and txtBox.Text = CStr(dblVar)

can be used to assign the contents of a text box to the Double variable dblVar and vice versa. CDbl, CInt, and CStr, which stand for "convert to Double," "convert to Integer," and "convert to String," are referred to as data-conversion or type-casting functions. Example 2 by the user:

The following program computes the sum of two numbers supplied

OBJECT

PROPERTY

SETTING

frmAdd lblFirstNum

Text AutoSize Text

Addition False First Number:

AutoSize Text

False Second Number:

Text Text ReadOnly

Compute Sum Sum: True

txtFirstNum lblSecondNum txtSecondNum btnCompute lblSum txtSum

```
Private Sub btnCompute_Click() Handles btnCompute.Click Dim num1, num2, sum As Double num1 = CDbl(txtFirstNum.Text) num2 = CDbl(txtSecondNum.Text) sum = num1 + num2
txtSum.Text = CStr(sum) End Sub
```

◆

79

80

◆

Chapter 3 Variables, Input, and Output

[Run, type 45 into the first text box, type 55 into the second text box, and click on the button.]

■ Auto Correction The Auto Correction feature of IntelliSense suggests corrections when errors occur, and allows you to select a correction to be applied to the code. When a prohibited statement is entered, a blue squiggly error line appears under the offending part of the statement. If the squiggly line has a short red line segment at its right end, the Auto Correction feature is available for the error. When you hover the cursor over the squiggly line, a small box containing an exclamation mark inside a red circle appears. Clicking on the small box produces an Auto Correction helper box that describes the error and makes a suggestion for fixing it. Figure 3.2 shows a typical Auto Correction helper box for a data-type-conversion error. If you click on the line beginning "Replace," the change will be made for you.

FIGURE 3.2

An Auto Correction helper box.

Источник: https://dokumen.pub/an-introduction-to-programming-using-visual-basic-2008-with-visual-studio-expression-edition-dvd-7th-ed-9780136060727-2008023532-0136060722.html

An introduction to programming using Visual Basic 2008: with Visual Studio Expression edition DVD [7th ed] 9780136060727, 2008023532, 0136060722

Table of contents :

Citation preview

TO

AN INTRODUCTION PROGRAMMING USING ®

VISUAL BASIC 2008 with Visual Studio® Express Edition DVD

SEVENTH EDITION

This page intentionally left blank

TO

AN INTRODUCTION PROGRAMMING USING ®

VISUAL BASIC 2008 with Visual Studio® Express Edition DVD

SEVENTH EDITION

David I. Schneider University of Maryland

Upper Saddle River, NJ 07458

Library of Congress Cataloging-in-Publication Data Schneider, David I. An introduction programming using Visual Basic 2008 / David I. Schneider. —7th ed. p. cm. Includes index. ISBN 978-0-13-606072-7 1. Microsoft Visual BASIC. 2. BASIC (Computer program language) I. Title. QA76.73.B3S3336 2008 005.2'762—dc22 2008023532

Editorial Director, Computer Science and Engineering: Marcia J. Horton Executive Editor: Tracy Dunkelberger Editorial Assistant: Melinda Haggerty Director of Marketing: Margaret Waples Marketing Manager: Christopher Kelly Associate Editor: ReeAnne Davies Senior Managing Editor: Scott Disanno Production Editor: Irwin Zucker Art Director: Kenny Beck Cover Art: Franklin Hammond/Images.com Art Editor: Gregory Dulles Manufacturing Manager: Alan Fischer Manufacturing Buyer: Lisa McDowell

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs. Pearson Education Ltd., London Pearson Education Singapore, Pte. Ltd. Pearson Education Canada, Inc. Pearson Education—Japan Pearson Education Australia PTY, Limited Pearson Education North Asia Ltd., Hong Kong Pearson Educación de Mexico, S.A. de C.V. Pearson Education Malaysia, Pte. Ltd. Pearson Education Inc., Upper Saddle River, New Jersey 10 9 8 7 6 5 4 3 2 1 ISBN 10: 0-13-606072-2 ISBN 13: 978-0-13-606072-7

This page intentionally left blank

PREFACE

Since its introduction in 1991, Visual Basic has been the most widely used programming language in the world. The latest incarnation of Visual Basic, called Visual Basic 2008, brings the language into the Internet age by incorporating the .Net Framework. Visual Basic programmers are enthusiastically embracing the new features of the language. Likewise, students who are learning their first programming language will find VB 2008 the ideal tool to understand the development of computer programs. My objectives when writing this text were as follows: 1.

To develop focused chapters. Rather than covering many topics superficially, I concentrate on important subjects and cover them thoroughly.

2.

To use examples and exercises that students can relate to, appreciate, and feel comfortable with. I frequently use real data. Examples do not have so many embellishments that students are distracted from the programming techniques illustrated.

3.

To produce compactly written text that students will find both readable and informative. The main points of each topic are discussed first, and then the peripheral details are presented as comments.

4.

To teach good programming practices that are in step with modern programming methodology. Problem solving techniques and structured programming are discussed early and used throughout the book. The style follows object-oriented programming principles.

5.

To provide insights into the major applications of computers.

Unique and Distinguishing Featur es Exercises for Most Sections. Each section that teaches programming has an exercise set. The exercises both reinforce the understanding of the key ideas of the section and challenge the student to explore applications. Most of the exercise sets require the student to trace programs, find errors, and write programs. The answers to all the odd-numbered exercises in Chapters 2 through 8 and selected odd numbered exercises from Chapters 9, 10, and 11 are given at the end of the text. Practice Problems. Practice Problems are carefully selected exercises located at the end of a section, just before the exercise set. Complete solutions are given following the exercise set. The practice problems often focus on points that are potentially confusing or are best appreciated after the student has worked on them. The reader should seriously attempt the practice problems and study their solutions before moving on to the exercises. Programming Projects. Beginning with Chapter 3, every chapter contains programming projects. The programming projects not only reflect the variety of ways that computers are used in the business community, but also present some games and general-interest topics. The large number and wide range of difficulty of the programming projects provide the flexibility to adapt the course to the interests and abilities of the students. Some programming projects in later chapters can be assigned as end-of-the-semester projects.

Comments. Extensions and fine points of new topics are deferred to the “Comments” portion at the end of each section so that they will not interfere with the flow of the presentation. Case Studies. Each of the four case studies focuses on an important programming application. The problems are analyzed and the programs are developed with top-down charts and pseudocode. The programs can be found on the companion website at www.prenhall.com/schneider. Chapter Summaries. In Chapters 2 through 11, the key results are stated and the important terms are summarized at the end of the chapter. Arrays. Arrays are introduced gently in two sections. The first section presents the basic definitions and avoids procedures. The second section presents the techniques for manipulating arrays and shows how to pass arrays to procedures. “How To” Appendix. Appendix B provides a compact, step-by-step reference on how to carry out standard tasks in the Visual Basic environment. Appendix on Debugging. Placing of the discussion of Visual Basic's sophisticated debugger in Appendix D allows the instructor flexibility in deciding when to cover this topic. Appendix on Converting from Visual Basic 6.0 to Visual Basic 2008. This appendix is included primarily to assist the instructor who is familiar with VB 6.0 and is new to using VB 2008. Companion Website. The companion website (http://www.prenhall.com/schneider) contains all the examples, case studies, and data files referenced in the book. (Details can be found on page xv). Instructor Supplements. In addition to the materials on the companion website, solutions to every exercise and programming project, a test item file for each chapter, and PowerPoint lecture slides for each chapter are available to instructors. Contact your local Pearson sales representative for information on how to access these resources.

What's New in the Seventh Edition 1.

The version of Visual Basic has been upgraded from Visual Basic 2005 to Visual Basic 2008 and relevant new features of Visual Basic 2008 have been incorporated.

2.

Suggestions from students and reviewers have been incorporated as much as possible.

3.

Chapter 1 has been shortened considerably and combined with Chapter 2. (Some of the material from the previous Chapter 1 has been moved to the Appendices.)

4.

Chapter 3 has been split into two chapters.

5.

Chapter 4 (General Procedures) of the previous edition has been moved to follow the chapter on decision structures.

6.

The concept of scope is discussed earlier in the book.

7.

The real-life data in the examples and exercises have been updated and revised.

8.

Text files are dealt with inside the Visual Basic environment instead of with Notepad.

9.

All screen captures have been updated to show the Microsoft® Windows Vista operating system.

10.

Message boxes are invoked with MessageBox.Show instead of MsgBox.

ACKNOWLEDGMENTS

Many talented instructors, students, and programmers provided helpful comments and constructive suggestions during the preparation of this text. For their contributions to the quality of the first six editions of the book I extend my gratitude to A. Abonomah, University of Akron; Timothy Babbitt, Rochester Institute of Technology; William Barnett, Northwestern State University; Sherry Barriclow, Grand Valley State University; Robert Berman, Wayne State University; William Burrows, University of Washington; David Chao, San Francisco State University; Christopher Chisolm, University of Nebraska, Omaha; Robert Coil, Cincinnati State Technical and Community College; Gary Cornell, University of Connecticut; Ronit Dancis; John DaPonte, Southern Connecticut State University; Ward Deutschman, Briarcliff; Ralph Duffy, North Seattle Community College; Charles Fairchild; Pat Fenton, West Valley College; David Fichbohm, Golden Gate University; Robert Fritz, American River College; Matthew Goddard, New Hampshire Technical College; Mickie Goodro, Casper College; Wade T. Graves, Grayson Community College; Christine Griffin; Gary Haw, MIPS Software Development Inc.; Shelly Hawkins, Microsoft; Tom Janicki, Kent State University; Dana Johnson, North Dakota State University; Dan Joseph, Rochester Institute of Technology; Del Kimber, Clemson University; Wanda Kunkle, Rowan College; Paul Lecoq, San Francisco Community College; David Leitch, DeVry Institute; David Letcher, The College of New Jersey; Kieran Mathieson, Oakland University; Charlie Miri, Delaware Tech; George Nezlek, DePaul University; Ron Notes, Hebrew Academy of Greater Washington; Mike Paul, Berry University; T. S. Pennington, Maple Woods Community College; Arland Richmond, Computer Learning Center; David Rosser, Essex County College; Arturo Salazar, San Francisco State; Susanne Peterson, Microsoft; Janie Schwark, Microsoft; Mike Talber, Portland Community College; Steve Turek, DeVry Institute of Technology, Kansas City; Jac Van Deventer, Washington State University; Randy Weinberg, St. Cloud State University; Laurie Werner, Miami University; Melinda, White, Santa Fe Community College; Ronald Williams, Central Piedmont Community College; Amit Kalani, CIStems Solutions LLC; Priti Kalani, MobiEcast Corporation; Chris Panell, Heald College; Kevin Parker, Idaho State University; TJ Racoosin, rSolutions; and Bill Tinker, Aries Software. The current edition benefited greatly from the valuable comments of the following knowledgeable reviewers: Paul Bladek, Edmonds Community College (WA) Jason Blazzard, Lewis Clark State College & Washington State Douglas Bock, Southern Illinois University Joe Brady, University of Delaware Peter Chan, Canyon College (CA) Debra Chapman, University of South Alabama Chakib Chraibi, Barry University Ronald Greenberg, Loyola University of Chicago Dennis Higgins, SUNY- Oneonta Duke Hutchins, Bowling Green State University Stephen Klein, Ramapo College (NJ) Kathleen Lynch, University of Wisconsin (Oshkosh) Joe Mast, Eastern Mennonite University Tom McCollough, Hillsborough CC (FL) Jim McKeown, Dakota State University

Acknowledgments Josh Pauli, Dakota State University Markita Price, University of Missouri Paula Ruby, Arkansas State University Sara Rushinek, University of Miami Cheryl Schlitter, University of Colorado - Colorado Springs Syd Shewchuk, Heald College Debbie Tesch, Xavier University Sangsuree Vasupongayya, Portland State University Keith Weber, Idaho State University Many people are involved in the successful publication of a book. I wish to thank the dedicated team at Pearson whose support and diligence made this textbook possible, especially Marcia Horton, Editor-in-Chief, and Scott Disanno, Senior Managing Editor. I am grateful to Jeremy Schneider, Paul Mailhot, and Zane Lang for their careful proofreading. Production editor Irwin Zucker did a fantastic job producing the book and keeping it on schedule. I extend special thanks to my editor Tracy Dunkelberger. Her ideas and enthusiasm helped immensely with the preparation of the book.

THE COMPANION WEBSITE FOR STUDENTS AND INSTRUCTORS How to Access the Companion Website All the programs from the examples, case studies, text files for the exercises, database files and BMP images can be downloaded by students and instructors at www.prenhall.com/schneider.

Students The companion website, located at www.prenhall.com/schneider, contains all the programs from the examples and case studies set forth in this textbook, all the text files needed for the exercises, all databases needed for the exercises, and several picture files. All these files are contained in the folders "Ch02", "Ch03", "Ch04", and so on. Each chapter file contains a subfolder named "Text_Files_for_Exercises" which contains the text files needed for that chapter's exercises. The folder "Ch09" has a subfolder named "Pictures" that contains the picture files. The folder "Ch10" has a subfolder named "MajorDatabases" containing all the databases needed for the exercises. Each program is contained in a folder with a name in the form chapter–section–number. For instance, the program in Chapter 3, Section 1, Example 2 is contained in the folder "3-1-2". Many of the programs make use of a text file in the subfolder Debug of the program's folder named bin.

Instr uctors Instructor resources including solutions to the exercises, PowerPoint lecture slides, all the example programs and data files used by students throughout the book, and multiple-choice and true/false questions. Contact your local Pearson sales representative to gain access to this material.

Notice: This book contains many screen captures. When you run one of the programs downloaded from the website, what you see on your monitor might not look exactly like the screen capture shown in the book. To make them appear the same, you must check that your monitor is set to display 96 DPI (Dots Per Inch). To determine and/or change the DPI setting for your monitor, see the first item under "Utilize the Windows Environment" in Appendix B on pages 608-9.

USING THIS BOOK FOR A SHORT OR CONDENSED COURSE This book provides more than enough material for a complete semester course. The topics must be trimmed for courses lasting considerably less than a full semester. The following syllabus provides one possible way to present an abbreviated introduction to programming.

Chapter 1 An Introduction to Computers and Problem Solving 1.1 An Introduction to Computers 1.2 Windows, Folders, and Files Chapter 2 2.1 2.2 2.3

Visual Basic, Controls, and Events An Introduction to Visual Basic 2008 Visual Basic Controls Visual Basic Events

Chapter 3 3.1 3.2 3.3

Variables, Input, and Output Numbers Strings Input and Output

Chapter 4 4.1 4.2 4.3

Decisions Relational and Logical Operators If Blocks Select Case Blocks

Chapter 5 5.1 5.2 5.3

General Procedures Sub Procedures, Part I Sub Procedures, Part II Function Procedures

Chapter 6 6.1 6.2 6.3

Repetition Do Loops Processing Lists of Data With Do Loops For. . Next Loops

Chapter 7 Arrays 7.1 Creating And Accessing Arrays 7.2 Using Arrays Chapter 9 Additional Controls and Objects 9.1 List Boxes, Combo Boxes, and the File-Opening Control 9.2 Seven Elementary Controls xvi

1.1

An Introduction to Computers

An Introduction to Programming Using Visual Basic 2008 is a book about problem solving using computers. The programming language used is Visual Basic 2008 (hereafter shortened to Visual Basic), but the principles taught apply to many modern programming languages. The examples and exercises present a sampling of the ways that computers are used in the real world. Here are some questions that you might have about computers and programming. Question: What are the main components of a personal computer? Answer: Hidden from view inside the system unit are several components, including the microprocessor, memory, and hard drive of the computer. The central processing unit (CPU), sometimes referred to as the microprocessor, can be thought of as the computer's brain, which carries out all of the computations. The memory, often referred to as random access memory (RAM), stores instructions and data while they are being used by the computer. When the computer's power is turned off, the contents of memory are lost. A hard disk drive is used to store instructions and data when they are not being used in memory and when the computer is turned off. Inside the system unit there are also device cards, such as a graphics card, sound card, network card, and a modem. A graphics card is used to send an image to the monitor, and a sound card is used to send audio to a set of speakers attached to the computer. Network cards can be used to connect to a local area network (LAN) of computers, while a modem uses a telephone line or cable to connect to another computer. The personal computer also has several input and output devices, which are used to communicate with the computer. Standard input devices include the keyboard and mouse. Standard output devices include the monitor and printer. Instructions are entered into the computer by typing them on the keyboard, clicking a mouse, or loading them from a file located on a disk drive or downloaded from a network. Information processed by the computer can be displayed on the monitor, printed on the printer, or recorded on a disk drive. Question: What are some topics covered in this text that students could use immediately? Answer: Computer files can be created to hold lists of names, addresses, and phone numbers, which can be alphabetized and displayed in their entirety or selectively. Mathematical computations can be carried out for science, business, and engineering courses. Personal financial transactions, such as bank deposits and loans, can be recorded, organized, and analyzed. Line charts, pie charts, and bar charts can be created to enhance the data in a term paper. Question: How do we communicate with the computer? Answer: Many languages are used to communicate with the computer. At the lowest level, there is machine language, which is understood directly by the microprocessor, but is awkward for humans. Visual Basic is an example of a higher-level language. It consists of instructions to which people can relate, such as Click, If, and Do. Question: How do we get computers to perform complicated tasks? Answer: Tasks are broken down into a sequence of instructions that can be expressed in a computer language. (This text uses the language Visual Basic.) This sequence of instructions is called a program. Programs can range in size from two or three instructions to millions of instructions. Instructions are typed on the keyboard, or read in from a file

on a disk, and stored in the computer's memory. The process of executing the instructions is called running the program. Question: Are there certain features that all programs have in common? Answer: Most programs do three things: take in data, manipulate them, and give desired information. These operations are referred to as input, processing, and output. The input data might be held in a portion of the program, reside on a disk drive, or be provided by the computer operator in response to requests made by the computer while the program is running. The processing of the input data occurs inside the computer and can take from a fraction of a second to many hours. The output data are either displayed on the monitor, printed on the printer, or recorded on a disk. As a simple example, consider a program that computes sales tax. An item of input data is the cost of the thing purchased. The processing consists of multiplying the cost by a certain percentage. An item of output data is the resulting product, the amount of sales tax to be paid. Question: What are the meanings of the terms "hardware" and "software?" Answer: Hardware refers to the physical components of the computer, including all peripherals, the central processing unit, disk drives, and all mechanical and electrical devices. Programs are referred to as software. Question: What are the meanings of the terms "programmer" and "user?" Answer: A programmer is a person who solves problems by writing programs on a computer. After analyzing the problem and developing a plan for solving it, he or she writes and tests the program that instructs the computer how to carry out the plan. The program might be run many times, either by the programmer or by others. A user is any person who uses a program. While working through this text, you will function both as a programmer and as a user. Question: What is meant by problem solving? Answer: Problems are solved by carefully reading them to determine what data are given and what outputs are requested. Then a step-by-step procedure is devised to process the given data and produce the requested output. This procedure is called an algorithm. Finally, a computer program is written to carry out the algorithm. Algorithms are discussed in Section 1.4. Question: What types of problems are solved in this text? Answer: Carrying out business computations, creating and maintaining records, alphabetizing lists, and displaying tabular data are some of the types of problems we will solve. Question: When did the concept of computers and programming first appear? Answer: The first general-purpose computer, called the Analytical Engine, was described by Charles Babbage in 1837. Although it was never built, its logical design was essentially modern. (In 1939, the first general-purpose computer was actually built by John Atanasoff.) In 1843, Augusta Ada Byron showed how to carry out certain complex mathematical computations with the Analytical Engine, thereby establishing herself as the first computer programmer. For a historical perspective of the evolution of modern computers and programming languages, see Appendix E, A Biographical History of Computing.

Question: How did Visual Basic 2008 evolve? Answer: In the early 1960s, two mathematics professors at Dartmouth College developed BASIC to provide their students with an easily learned language that could tackle complicated programming projects. As the popularity of BASIC grew, refinements were introduced that permitted structured programming, which increases the reliability of programs. Visual Basic 1.0 is a version of BASIC developed in 1991 by the Microsoft Corporation to allow easy, visual-oriented development of Windows applications. Visual Basic 2008 is a language similar to the original Visual Basic, but more powerful. It is targeted for what is known as the .NET run time, which is a program that executes Visual Basic 2008 as well as programs from other languages that are targeted for the .NET run time. This will ultimately allow programs written in Visual Basic to be run on devices other than computers, such as cell phones and handheld devices. Other features of Visual Basic 2008 include full object-oriented programming capabilities and the development of Web services. Object-oriented programming is discussed in Chapter 11. Question: Are there any prerequisites to learning Visual Basic 2008? Answer: Since Visual Basic is used to write Windows applications, you should be familiar with Windows and understand how folders and files are managed with Windows. The key concepts are presented succinctly in Section 1.2 and discussed in detail in Appendix F. Question: Will it matter whether I use Windows XP or Windows Vista as the operating system? Answer: Visual Basic runs fine with either Windows XP or Windows Vista. However, the screens will vary in appearance. Figure 1.1 shows how the appearance of a Visual Basic program differs with the operating system. In this book, all screens have the Windows Vista appearance.

(a) Windows XP FIGURE 1.1

1.2

(b) Windows Vista

A Visual Basic window.

Windows, Folders, and Files

This preliminary section presents some terms used in this book.

■ Windows and its Little Windows Windows gets its name from the way it organizes the screen into rectangular regions. When you run a program, the program runs inside a bordered rectangular box.

Unfortunately Windows jargon calls these windows, so there's only a lowercase "w" to distinguish them from the operating system called Windows. Figure 1.2 shows the window that results from running one of the programs in this book. In Visual Basic terminology, such a window is also called a form.

Title bar

Minimize button

FIGURE 1.2

Maximize button

Close button

A Visual Basic window.

■ Mouse Actions Hover: Linger the mouse at a particular place and wait for a message (such as a tool tip) to appear. Drag an object: Move the mouse pointer until it is at the object, press the left mouse button and hold it down, move the mouse pointer until the object moves to where you want it to be, and finally, release the mouse button. (Sometimes this whole activity is called drag-and-drop.) Right-click: Press and release the right mouse button once. Click: Press and release the left mouse button once. (sometimes referred to as singleclick or left-click) Double-click: Click the left mouse button twice in quick succession. Note: An important Windows convention is that clicking selects an object so you can give Windows further directions about it, but double-clicking tells it to perform a default operation. For example, double-clicking on a folder will open that folder.

■ Files and Folders (A detailed discussion of files and folders can be found in Appendix F.) Disk: Either the hard disk, a diskette, a CD, or DVD. Each disk drive is identified by a letter followed by a colon.

File: Either a program file or a data file. Its name typically consists of letters, digits, and spaces. The name of the file is also called the base name. Extension of a file name: One or more letters, preceded by a period, that identify the type of file. For example, files created with Word have the extension doc or docx. Filename: The combination of the base name, the period, and the extension. The only characters that cannot be used in filenames are \, /, :, *, ?, , ", and 8 9 7 4

Introduction to Programming Using Visual Basic 2008, An (w/VS2008 DVD) : United States Edition

3.17 (12 ratings by Goodreads)

By (author) David I. Schneider

Description

Now in its Seventh Edition, Schneider's "tried and true" text - consistently praised by both students and instructors - is designed for students with no prior computer programming experience. Now updated for Visual Basic 2008, Schneider focuses on teaching problem-solving skills and sustainable programming skills. A broad range of real-world examples, section-ending exercises, case studies, and programming projects gives students significant hands-on experience.

Supplements:

Instructor: Solutions, Power Point Lecture Slides, Test Files

Student: Source Code

www.prenhall.com/schneider
show more

Product details

- Mixed media product 4