

STARTING OUT WITH

C++

From Control Structures  
through Objects

**BRIEF VERSION  
NINTH EDITION**



STARTING OUT WITH

C++

From Control Structures  
through Objects

**BRIEF VERSION  
NINTH EDITION**

**Tony Gaddis**  
Haywood Community College



330 Hudson Street, New York, NY 10013

**Senior Vice President Courseware Portfolio  
Management:** Marcia J. Horton  
**Director, Portfolio Management: Engineering,  
Computer Science & Global Editions:** Julian Partridge  
**Portfolio Manager:** Matt Goldstein  
**Portfolio Management Assistant:** Meghan Jacoby  
**Field Marketing Manager:** Demetrius Hall  
**Product Marketing Manager:** Yvonne Vannatta

**Marketing Assistant:** Jon Bryant  
**Managing Producer, ECS and Math:** Scott Disanno  
**Content Producer:** Amanda Brands  
**Composition:** SPi Global  
**Project Manager:** Billu Suresh/ SPi Global  
**Cover Designer:** Joyce Wells  
**Cover Image:** Viktor Malyshchys/123RF

Credits and acknowledgments borrowed from other sources and reproduced, with permission, appear on the Credits page in the endmatter of this textbook.

---

Copyright © 2019 Pearson Education, Inc. Hoboken, NJ 07030. All rights reserved. Manufactured in the United States of America. This publication is protected by copyright and permissions should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise.

For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions department, please visit [www.pearsoned.com/permissions/](http://www.pearsoned.com/permissions/).

Many of the designations by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

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 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 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, Ltd

Pearson Education North Asia, Ltd., Hong Kong  
Pearson Education de Mexico, S.A. de C.V.  
Pearson Education Malaysia, Pte. Ltd.  
Pearson Education, Inc., Hoboken

Library of Congress Cataloging-in-Publication Data  
FPO

1 18



ISBN-13: 978-0-13-489573-4  
ISBN-10: 0-13-489573-8

# Contents at a Glance

	<b>Preface</b>	<b>xv</b>
<b>CHAPTER 1</b>	<b>Introduction to Computers and Programming</b>	<b>1</b>
<b>CHAPTER 2</b>	<b>Introduction to C++</b>	<b>27</b>
<b>CHAPTER 3</b>	<b>Expressions and Interactivity</b>	<b>85</b>
<b>CHAPTER 4</b>	<b>Making Decisions</b>	<b>151</b>
<b>CHAPTER 5</b>	<b>Loops and Files</b>	<b>231</b>
<b>CHAPTER 6</b>	<b>Functions</b>	<b>305</b>
<b>CHAPTER 7</b>	<b>Arrays and Vectors</b>	<b>381</b>
<b>CHAPTER 8</b>	<b>Searching and Sorting Arrays</b>	<b>463</b>
<b>CHAPTER 9</b>	<b>Pointers</b>	<b>503</b>
<b>CHAPTER 10</b>	<b>Characters, C-Strings, and More about the <code>string</code> Class</b>	<b>557</b>
<b>CHAPTER 11</b>	<b>Structured Data</b>	<b>613</b>
<b>CHAPTER 12</b>	<b>Advanced File Operations</b>	<b>665</b>
<b>CHAPTER 13</b>	<b>Introduction to Classes</b>	<b>719</b>
<b>CHAPTER 14</b>	<b>More about Classes</b>	<b>817</b>
<b>CHAPTER 15</b>	<b>Inheritance, Polymorphism, and Virtual Functions</b>	<b>907</b>
	<b>Appendix A: The ASCII Character Set</b>	<b>989</b>
	<b>Appendix B: Operator Precedence and Associativity</b>	<b>991</b>
	<b>Quick References</b>	<b>993</b>
	<b>Index</b>	<b>995</b>

**Online** The following appendices are available at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis).

**Appendix C: Introduction to Flowcharting**

**Appendix D: Using UML in Class Design**

**Appendix E: Namespaces**

**Appendix F: Passing Command Line Arguments**

**Appendix G: Binary Numbers and Bitwise Operations**

**Appendix H: STL Algorithms**

**Appendix I: Multi-Source File Programs**

**Appendix J: Stream Member Functions for Formatting**

**Appendix K: Unions**

**Appendix L: Answers to Checkpoints**

**Appendix M: Answers to Odd Numbered Review Questions**

**Case Study 1: String Manipulation**

**Case Study 2: High Adventure Travel Agency—Part 1**

**Case Study 3: Loan Amortization**

**Case Study 4: Creating a String Class**

**Case Study 5: High Adventure Travel Agency—Part 2**

**Case Study 6: High Adventure Travel Agency—Part 3**

**Case Study 7: Intersection of Sets**

**Case Study 8: Sales Commission**

# Contents

## **Preface xv**

## **CHAPTER 1 Introduction to Computers and Programming 1**

- 1.1 Why Program? 1
- 1.2 Computer Systems: Hardware and Software 2
- 1.3 Programs and Programming Languages 8
- 1.4 What Is a Program Made of? 14
- 1.5 Input, Processing, and Output 17
- 1.6 The Programming Process 18
- 1.7 Procedural and Object-Oriented Programming 22
- Review Questions and Exercises 24*

## **CHAPTER 2 Introduction to C++ 27**

- 2.1 The Parts of a C++ Program 27
- 2.2 The `cout` Object 31
- 2.3 The `#include` Directive 36
- 2.4 Variables, Literals, and Assignment Statements 38
- 2.5 Identifiers 42
- 2.6 Integer Data Types 43
- 2.7 The `char` Data Type 49
- 2.8 The C++ `string` Class 53
- 2.9 Floating-Point Data Types 55
- 2.10 The `bool` Data Type 58
- 2.11 Determining the Size of a Data Type 59
- 2.12 More about Variable Assignments and Initialization 60
- 2.13 Scope 62
- 2.14 Arithmetic Operators 63
- 2.15 Comments 71
- 2.16 Named Constants 73
- 2.17 Programming Style 75
- Review Questions and Exercises 77*
- Programming Challenges 81*

**CHAPTER 3 Expressions and Interactivity 85**

- 3.1 The `cin` Object 85
- 3.2 Mathematical Expressions 91
- 3.3 When You Mix Apples and Oranges: Type Conversion 100
- 3.4 Overflow and Underflow 102
- 3.5 Type Casting 103
- 3.6 Multiple Assignment and Combined Assignment 106
- 3.7 Formatting Output 110
- 3.8 Working with Characters and `string` Objects 120
- 3.9 More Mathematical Library Functions 126
- 3.10 Focus on Debugging: Hand Tracing a Program 132
- 3.11 Focus on Problem Solving: A Case Study 134
- Review Questions and Exercises* 138
- Programming Challenges* 144

**CHAPTER 4 Making Decisions 151**

- 4.1 Relational Operators 151
- 4.2 The `if` Statement 156
- 4.3 Expanding the `if` Statement 164
- 4.4 The `if/else` Statement 168
- 4.5 Nested `if` Statements 171
- 4.6 The `if/else if` Statement 178
- 4.7 Flags 183
- 4.8 Logical Operators 184
- 4.9 Checking Numeric Ranges with Logical Operators 191
- 4.10 Menus 192
- 4.11 Focus on Software Engineering: Validating User Input 195
- 4.12 Comparing Characters and Strings 197
- 4.13 The Conditional Operator 201
- 4.14 The `switch` Statement 204
- 4.15 More about Blocks and Variable Scope 213
- Review Questions and Exercises* 216
- Programming Challenges* 222

**CHAPTER 5 Loops and Files 231**

- 5.1 The Increment and Decrement Operators 231
- 5.2 Introduction to Loops: The `while` Loop 236
- 5.3 Using the `while` Loop for Input Validation 243
- 5.4 Counters 245
- 5.5 The `do-while` Loop 246
- 5.6 The `for` Loop 251
- 5.7 Keeping a Running Total 261
- 5.8 Sentinels 264
- 5.9 Focus on Software Engineering: Deciding Which Loop to Use 265
- 5.10 Nested Loops 266
- 5.11 Using Files for Data Storage 269
- 5.12 Optional Topics: Breaking and Continuing a Loop 288
- Review Questions and Exercises* 292
- Programming Challenges* 297



**CHAPTER 6 Functions 305**

- 6.1 Focus on Software Engineering: Modular Programming 305
- 6.2 Defining and Calling Functions 306
- 6.3 Function Prototypes 315
- 6.4 Sending Data into a Function 317
- 6.5 Passing Data by Value 322
- 6.6 Focus on Software Engineering: Using Functions in a  
Menu-Driven Program 324
- 6.7 The return Statement 328
- 6.8 Returning a Value from a Function 330
- 6.9 Returning a Boolean Value 338
- 6.10 Local and Global Variables 340
- 6.11 Static Local Variables 348
- 6.12 Default Arguments 351
- 6.13 Using Reference Variables as Parameters 354
- 6.14 Overloading Functions 360
- 6.15 The `exit()` Function 364
- 6.16 Stubs and Drivers 367
- Review Questions and Exercises* 369
- Programming Challenges* 372

**CHAPTER 7 Arrays and Vectors 381**

- 7.1 Arrays Hold Multiple Values 381
- 7.2 Accessing Array Elements 383
- 7.3 No Bounds Checking in C++ 395
- 7.4 The Range-Based for Loop 398
- 7.5 Processing Array Contents 402
- 7.6 Focus on Software Engineering: Using Parallel Arrays 410
- 7.7 Arrays as Function Arguments 413
- 7.8 Two-Dimensional Arrays 424
- 7.9 Arrays with Three or More Dimensions 431
- 7.10 Focus on Problem Solving and Program Design: A Case Study 433
- 7.11 Introduction to the STL vector 435
- Review Questions and Exercises* 449
- Programming Challenges* 454

**CHAPTER 8 Searching and Sorting Arrays 463**

- 8.1 Focus on Software Engineering: Introduction to Search Algorithms 463
- 8.2 Focus on Problem Solving and Program Design: A Case Study 469
- 8.3 Focus on Software Engineering: Introduction to Sorting Algorithms 476
- 8.4 Focus on Problem Solving and Program Design: A Case Study 486
- 8.5 Sorting and Searching vectors 495
- Review Questions and Exercises* 498
- Programming Challenges* 499

**CHAPTER 9 Pointers 503**

- 9.1 Getting the Address of a Variable 503
- 9.2 Pointer Variables 505
- 9.3 The Relationship between Arrays and Pointers 512

- 9.4 Pointer Arithmetic 516
- 9.5 Initializing Pointers 518
- 9.6 Comparing Pointers 519
- 9.7 Pointers as Function Parameters 521
- 9.8 Dynamic Memory Allocation 530
- 9.9 Returning Pointers from Functions 534
- 9.10 Using Smart Pointers to Avoid Memory Leaks 541
- 9.11 Focus on Problem Solving and Program Design: A Case Study 544
- Review Questions and Exercises* 550
- Programming Challenges* 553

**CHAPTER 10 Characters, C-Strings, and More about the string Class 557**

- 10.1 Character Testing 557
- 10.2 Character Case Conversion 561
- 10.3 C-Strings 564
- 10.4 Library Functions for Working with C-Strings 568
- 10.5 String/Numeric Conversion Functions 579
- 10.6 Focus on Software Engineering: Writing Your Own  
C-String-Handling Functions 585
- 10.7 More about the C++ string Class 591
- 10.8 Focus on Problem Solving and Program Design: A Case Study 603
- Review Questions and Exercises* 604
- Programming Challenges* 607

**CHAPTER 11 Structured Data 613**

- 11.1 Abstract Data Types 613
- 11.2 Structures 615
- 11.3 Accessing Structure Members 618
- 11.4 Initializing a Structure 622
- 11.5 Arrays of Structures 625
- 11.6 Focus on Software Engineering: Nested Structures 627
- 11.7 Structures as Function Arguments 631
- 11.8 Returning a Structure from a Function 634
- 11.9 Pointers to Structures 637
- 11.10 Focus on Software Engineering: When to Use.,  
When to Use ->, and When to Use \* 640
- 11.11 Enumerated Data Types 642
- Review Questions and Exercises* 653
- Programming Challenges* 659

**CHAPTER 12 Advanced File Operations 665**

- 12.1 File Operations 665
- 12.2 File Output Formatting 671
- 12.3 Passing File Stream Objects to Functions 673
- 12.4 More Detailed Error Testing 675
- 12.5 Member Functions for Reading and Writing Files 678
- 12.6 Focus on Software Engineering: Working with Multiple Files 686
- 12.7 Binary Files 688
- 12.8 Creating Records with Structures 693

- 12.9 Random-Access Files 697
- 12.10 Opening a File for Both Input and Output 705
  - Review Questions and Exercises* 710
  - Programming Challenges* 713

### **CHAPTER 13 Introduction to Classes 719**

- 13.1 Procedural and Object-Oriented Programming 719
- 13.2 Introduction to Classes 726
- 13.3 Defining an Instance of a Class 731
- 13.4 Why Have Private Members? 744
- 13.5 Focus on Software Engineering: Separating Class Specification from Implementation 745
- 13.6 Inline Member Functions 751
- 13.7 Constructors 754
- 13.8 Passing Arguments to Constructors 759
- 13.9 Destructors 767
- 13.10 Overloading Constructors 771
- 13.11 Private Member Functions 775
- 13.12 Arrays of Objects 777
- 13.13 Focus on Problem Solving and Program Design: An OOP Case Study 781
- 13.14 Focus on Object-Oriented Programming: Simulating Dice with Objects 788
- 13.15 Focus on Object-Oriented Design: The Unified Modeling Language (UML) 792
- 13.16 Focus on Object-Oriented Design: Finding the Classes and Their Responsibilities 794
  - Review Questions and Exercises* 803
  - Programming Challenges* 808

### **CHAPTER 14 More about Classes 817**

- 14.1 Instance and Static Members 817
- 14.2 Friends of Classes 825
- 14.3 Memberwise Assignment 830
- 14.4 Copy Constructors 831
- 14.5 Operator Overloading 837
- 14.6 Object Conversion 864
- 14.7 Aggregation 866
- 14.8 Focus on Object-Oriented Design: Class Collaborations 871
- 14.9 Focus on Object-Oriented Programming: Simulating the Game of Cho-Han 876
- 14.10 Rvalue References and Move Semantics 886
  - Review Questions and Exercises* 895
  - Programming Challenges* 900

### **CHAPTER 15 Inheritance, Polymorphism, and Virtual Functions 907**

- 15.1 What Is Inheritance? 907
- 15.2 Protected Members and Class Access 916
- 15.3 Constructors and Destructors in Base and Derived Classes 922
- 15.4 Redefining Base Class Functions 936
- 15.5 Class Hierarchies 941
- 15.6 Polymorphism and Virtual Member Functions 947
- 15.7 Abstract Base Classes and Pure Virtual Functions 963

- 15.8 Multiple Inheritance 970
  - Review Questions and Exercises* 977
  - Programming Challenges* 981

**Appendix A: The ASCII Character Set 989**

**Appendix B: Operator Precedence and Associativity 991**

**Quick References 993**

**Index 995**

**Online**

The following appendices are available at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis).

**Appendix C: Introduction to Flowcharting**

**Appendix D: Using UML in Class Design**

**Appendix E: Namespaces**

**Appendix F: Passing Command Line Arguments**

**Appendix G: Binary Numbers and Bitwise Operations**

**Appendix H: STL Algorithms**

**Appendix I: Multi-Source File Programs**

**Appendix J: Stream Member Functions for Formatting**

**Appendix K: Unions**

**Appendix L: Answers to Checkpoints**

**Appendix M: Answers to Odd Numbered Review Questions**

**Case Study 1: String Manipulation**

**Case Study 2: High Adventure Travel Agency—Part 1**

**Case Study 3: Loan Amortization**

**Case Study 4: Creating a String Class**

**Case Study 5: High Adventure Travel Agency—Part 2**

**Case Study 6: High Adventure Travel Agency—Part 3**

**Case Study 7: Intersection of Sets**

**Case Study 8: Sales Commission**

## LOCATION OF VIDEONOTES IN THE TEXT

<b>Chapter 1</b>	Introduction to Flowcharting, p. 20 Designing a Program with Pseudocode, p. 20 Designing the Account Balance Program, p. 25 Predicting the Result of Problem 33, p. 26
<b>Chapter 2</b>	Using <code>cout</code> , p. 32 Variable Definitions, p. 38 Assignment Statements and Simple Math Expressions, p. 63 Solving the Restaurant Bill Problem, p. 81
<b>Chapter 3</b>	Reading Input with <code>cin</code> , p. 85 Formatting Numbers with <code>setprecision</code> , p. 113 Solving the Stadium Seating Problem, p. 144
<b>Chapter 4</b>	The <code>if</code> Statement, p. 156 The <code>if/else</code> statement, p. 168 The <code>if/else if</code> Statement, p. 178 Solving the Time Calculator Problem, p. 223
<b>Chapter 5</b>	The <code>while</code> Loop, p. 236 The <code>for</code> Loop, p. 251 Reading Data from a File, p. 278 Solving the Calories Burned Problem, p. 297
<b>Chapter 6</b>	Functions and Arguments, p. 317 Value-Returning Functions, p. 330 Solving the Markup Problem, p. 372
<b>Chapter 7</b>	Accessing Array Elements with a Loop, p. 386 Passing an Array to a Function, p. 413 Solving the Chips and Salsa Problem, p. 455
<b>Chapter 8</b>	The Binary Search, p. 466 The Selection Sort, p. 482 Solving the Charge Account Validation Modification Problem, p. 500
<b>Chapter 9</b>	Dynamically Allocating an Array, p. 531 Solving the Pointer Rewrite Problem, p. 554
<b>Chapter 10</b>	Writing a C-String-Handling Function, p. 585 More About the <code>string</code> Class, p. 591 Solving the Backward String Problem, p. 607

## LOCATION OF VIDEONOTES IN THE TEXT *(continued)*

<b>Chapter 11</b>	Creating a Structure, p. 615 Passing a Structure to a Function, p. 631 Solving the Weather Statistics Problem, p. 659
<b>Chapter 12</b>	Passing File Stream Objects to Functions, p. 673 Working with Multiple Files, p. 686 Solving the File Encryption Filter Problem, p. 716
<b>Chapter 13</b>	Writing a Class, p. 726 Defining an Instance of a Class, p. 731 Solving the Employee Class Problem, p. 808
<b>Chapter 14</b>	Operator Overloading, p. 837 Class Aggregation, p. 866 Solving the NumDays Problem, p. 901
<b>Chapter 15</b>	Redefining a Base Class Function in a Derived Class, p. 936 Polymorphism, p. 947 Solving the Employee and ProductionWorker Classes Problem, p. 981

# Preface

Welcome to the Brief Version of *Starting Out with C++: From Control Structures through Objects, 9th edition*. This book is intended for use in a two-semester C++ programming sequence, or an accelerated one-semester course. Students new to programming, as well as those with prior course work in other languages, will find this text beneficial. The fundamentals of programming are covered for the novice, while the details, pitfalls, and nuances of the C++ language are explored in depth for both the beginner and more experienced student. The book is written with clear, easy-to-understand language, and it covers all the necessary topics for an introductory programming course. This text is rich in example programs that are concise, practical, and real-world oriented, ensuring that the student not only learns how to implement the features and constructs of C++, but why and when to use them.

## Changes in the Ninth Edition

This book's pedagogy, organization, and clear writing style remain the same as in the previous edition. Many improvements and updates have been made, which are summarized here:

- The material on the Standard Template Library (STL) has been completely rewritten and expanded into its own chapter. Previously, covered exceptions, templates, and gave brief coverage to the STL. In this edition, covers exceptions and templates, is a new chapter dedicated to the STL. The new chapter covers the following topics:
  - The array and vector classes
  - The various types of iterators
  - Emplacement versus insertion
  - The `map`, `multimap`, and `unordered_map` Classes
  - The `set`, `multiset`, and `unordered_set` Classes
  - Sorting and searching algorithms
  - Permutation algorithms
  - Set algorithms
  - Using function pointers with STL algorithms
  - Function objects, or functors
  - Lambda expressions
- Chapter 2 now includes a discussion of alternative forms of variable initialization, including functional notation, and brace notation (also known as uniform initialization).
- Chapter 3 now mentions the `round` function, introduced in C++ 11.

- Chapter 7 now introduces array initialization much earlier.
- In Chapter 8, the bubble sort algorithm has been rewritten and improved.
- A new example of sorting and searching a vector of strings has been added to Chapter 8.
- In Chapter 9, the section on smart pointers now gives an overview of `shared_ptrs` and `weak_ptrs`, in addition to the existing coverage of `unique_ptrs`.
- In Chapter 10, a new *In the Spotlight* section on string tokenizing has been added.
- Chapter 10 now covers the `string`-to-number conversion functions that were introduced in C++ 11.
- The material on unions that previously appeared in Chapter 11 has been moved to Appendix K, available on the book's companion Website.
- Chapter 13 has new sections covering:
  - Member initialization lists.
  - In-place initialization.
  - Constructor delegation.
- Several new topics were added to Chapter 14, including:
  - Rvalue references and move semantics.
  - Checking for self-assignment when overloading the `=` operator.
  - Using member initialization lists in aggregate classes.
- Chapter 15 includes a new section on constructor inheritance.
- Several new programming problems have been added throughout the book.

## Organization of the Text

This text teaches C++ in a step-by-step fashion. Each chapter covers a major set of topics and builds knowledge as the student progresses through the book. Although the chapters can be easily taught in their existing sequence, some flexibility is provided. The diagram shown in Figure P-1 suggests possible sequences of instruction.

Chapter 1 covers fundamental hardware, software, and programming concepts. You may choose to skip this chapter if the class is already familiar with those topics. Chapters 2 through 7 cover basic C++ syntax, data types, expressions, selection structures, repetition structures, functions, and arrays. Each of these chapters builds on the previous chapter and should be covered in the order presented.

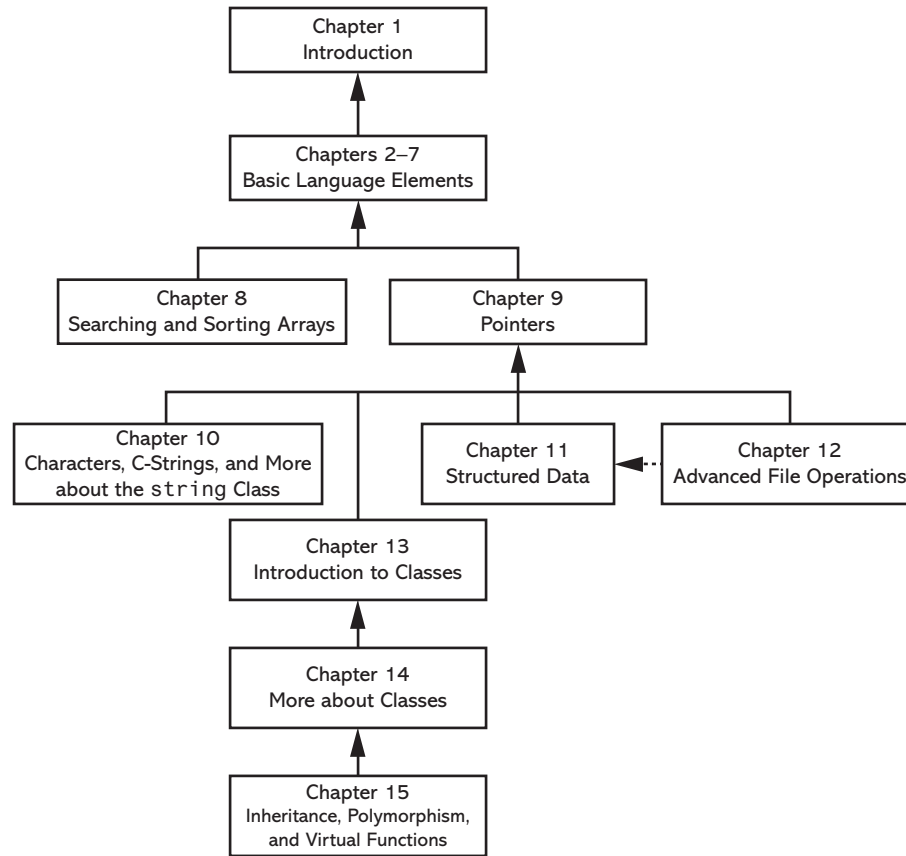
After Chapter 7 has been covered, you may proceed to Chapter 8, or jump to Chapter 9.

After Chapter 9 has been covered, Chapter 10, 11, 12 or 13 may be covered. (If you jump to Chapter 12 at this point, you will need to postpone Sections 12.8, 12.9, and 12.10 until Chapter 11 has been covered.) After Chapter 13, you may cover Chapters 14 in sequence.

This text's approach starts with a firm foundation in structured, procedural programming before delving fully into object-oriented programming and advanced data structures.



**Figure P-1** Chapter dependency chart



## Brief Overview of Each Chapter

### Chapter 1: Introduction to Computers and Programming

This chapter provides an introduction to the field of computer science and covers the fundamentals of programming, problem solving, and software design. The components of programs, such as key words, variables, operators, and punctuation, are covered. The tools of the trade, such as pseudocode, flow charts, and hierarchy charts, are also presented.

### Chapter 2: Introduction to C++

This chapter gets the student started in C++ by introducing data types, identifiers, variable declarations, constants, comments, program output, simple arithmetic operations, and C-strings. Programming style conventions are introduced and good programming style is modeled here, as it is throughout the text.

### Chapter 3: Expressions and Interactivity

In this chapter, the student learns to write programs that input and handle numeric, character, and string data. The use of arithmetic operators and the creation of mathematical expressions are covered in greater detail, with emphasis on operator precedence. Debugging is introduced, with a section on hand tracing a program. Sections are also included on simple output formatting, on data type conversion and type casting, and on using library functions that work with numbers.

### Chapter 4: Making Decisions

Here, the student learns about relational operators, relational expressions, and how to control the flow of a program with the `if`, `if/else`, and `if/else if` statements. The conditional operator and the `switch` statement are also covered. Crucial applications of these constructs are covered, such as menu-driven programs and the validation of input.

### Chapter 5: Loops and Files

This chapter covers repetition control structures. The `while` loop, `do-while` loop, and `for` loop are taught, along with common uses for these devices. Counters, accumulators, running totals, sentinels, and other application-related topics are discussed. Sequential file I/O is also introduced. The student learns to read and write text files, and use loops to process the data in a file.

### Chapter 6: Functions

In this chapter, the student learns how and why to modularize programs, using both `void` and value returning functions. Argument passing is covered, with emphasis on when arguments should be passed by value versus when they need to be passed by reference. Scope of variables is covered, and sections are provided on local versus global variables and on static local variables. Overloaded functions are also introduced and demonstrated.

**Chapter 7: Arrays and Vectors**

In this chapter, the student learns to create and work with single and multi-dimensional arrays. Many examples of array processing are provided including examples illustrating how to find the sum, average, highest, and lowest values in an array, and how to sum the rows, columns, and all elements of a two-dimensional array. Programming techniques using parallel arrays are also demonstrated, and the student is shown how to use a data file as an input source to populate an array. STL vectors are introduced and compared to arrays.

**Chapter 8: Searching and Sorting Arrays**

Here, the student learns the basics of sorting arrays and searching for data stored in them. The chapter covers the Bubble Sort, Selection Sort, Linear Search, and Binary Search algorithms. There is also a section on sorting and searching STL vector objects.

**Chapter 9: Pointers**

This chapter explains how to use pointers. Pointers are compared to and contrasted with reference variables. Other topics include pointer arithmetic, initialization of pointers, relational comparison of pointers, pointers and arrays, pointers and functions, dynamic memory allocation, and more.

**Chapter 10: Characters, C-Strings, and More about the string Class**

This chapter discusses various ways to process text at a detailed level. Library functions for testing and manipulating characters are introduced. C-strings are discussed, and the technique of storing C-strings in char arrays is covered. An extensive discussion of the `string` class methods is also given.

**Chapter 11: Structured Data**

The student is introduced to abstract data types and taught how to create them using structures, unions, and enumerated data types. Discussions and examples include using pointers to structures, passing structures to functions, and returning structures from functions.

**Chapter 12: Advanced File Operations**

This chapter covers sequential access, random access, text, and binary files. The various modes for opening files are discussed, as well as the many methods for reading and writing file contents. Advanced output formatting is also covered.

**Chapter 13: Introduction to Classes**

The student now shifts focus to the object-oriented paradigm. This chapter covers the fundamental concepts of classes. Member variables and functions are discussed. The student learns about private and public access specifications, and reasons to use each. The topics of constructors, overloaded constructors, and destructors are also presented. The chapter presents a section modeling classes with UML, and how to find the classes in a particular problem.

### Chapter 14: More about Classes

This chapter continues the study of classes. Static members, friends, memberwise assignment, and copy constructors are discussed. The chapter also includes in-depth sections on operator overloading, object conversion, and object aggregation. There is also a section on class collaborations and the use of CRC cards.

### Chapter 15: Inheritance, Polymorphism, and Virtual Functions

The study of classes continues in this chapter with the subjects of inheritance, polymorphism, and virtual member functions. The topics covered include base and derived class constructors and destructors, virtual member functions, base class pointers, static and dynamic binding, multiple inheritance, and class hierarchies.

### Appendix A: The ASCII Character Set

A list of the ASCII and Extended ASCII characters and their codes

### Appendix B: Operator Precedence and Associativity

A chart showing the C++ operators and their precedence

## Features of the Text

### Concept Statements

Each major section of the text starts with a concept statement. This statement summarizes the ideas of the section.

### Example Programs

The text has hundreds of complete example programs, each designed to highlight the topic currently being studied. In most cases, these are practical, real-world examples. Source code for these programs is provided so that students can run the programs themselves.

### Program Output

After each example program, there is a sample of its screen output. This immediately shows the student how the program should function.



### In the Spotlight

Each of these sections provides a programming problem and a detailed, step-by-step analysis showing the student how to solve it.



### VideoNotes

A series of online videos, developed specifically for this book, is available for viewing at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis). Icons appear throughout the text alerting the student to videos about specific topics.



### Checkpoints

Checkpoints are questions placed throughout each chapter as a self-test study aid. Answers for all Checkpoint questions can be downloaded from the book's Website at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis). This allows students to check how well they have learned a new topic.



### Notes

Notes appear at appropriate places throughout the text. They are short explanations of interesting or often misunderstood points relevant to the topic at hand.

**Warnings**

Warnings are notes that caution the student about certain C++ features, programming techniques, or practices that can lead to malfunctioning programs or lost data.

**Case Studies**

Case studies that simulate real-world applications appear in many chapters throughout the text. These case studies are designed to highlight the major topics of the chapter in which they appear.

**Review Questions and Exercises**

Each chapter presents a thorough and diverse set of review questions, such as fill-in-the-blank and short answer, that check the student's mastery of the basic material presented in the chapter. These are followed by exercises requiring problem solving and analysis, such as the *Algorithm Workbench*, *Predict the Output*, and *Find the Errors sections*. Answers to the odd-numbered review questions and review exercises can be downloaded from the book's Website at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis).

**Programming Challenges**

Each chapter offers a pool of programming exercises designed to solidify the student's knowledge of the topics currently being studied. In most cases, the assignments present real-world problems to be solved. When applicable, these exercises include input validation rules.

**Group Projects**

There are several group programming projects throughout the text, intended to be constructed by a team of students. One student might build the program's user interface, while another student writes the mathematical code, and another designs and implements a class the program uses. This process is similar to the way many professional programs are written and encourages team work within the classroom.

**Software Development Project: Serendipity Booksellers**

Available for download from the book's Website at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis). This is an ongoing project that instructors can optionally assign to teams of students. It systematically develops a "real-world" software package: a point-of-sale program for the fictitious Serendipity Booksellers organization. The Serendipity assignment for each chapter adds more functionality to the software, using constructs and techniques covered in that chapter. When complete, the program will act as a cash register, manage an inventory database, and produce a variety of reports.

**C++ Quick Reference Guide**

For easy access, a quick reference guide to the C++ language is printed on the inside back cover of the book.

**C++11**

Throughout the text, new C++11 language features are introduced. Look for the C++11 icon to find these new features.

## Supplements

### Student Online Resources

Many student resources are available for this book from the publisher. The following items are available on the Computer Science Portal at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis):

- The source code for each example program in the book
- Access to the book's VideoNotes
- A full set of appendices, including answers to the Checkpoint questions and answers to the odd-numbered review questions
- A collection of valuable Case Studies
- The complete Serendipity Booksellers Project

### Online Practice and Assessment with MyProgrammingLab

MyProgrammingLab helps students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyProgrammingLab improves the programming competence of beginning students who often struggle with the basic concepts and paradigms of popular high-level programming languages.

A self-study and homework tool, a MyProgrammingLab course consists of hundreds of small practice exercises organized around the structure of this textbook. For students, the system automatically detects errors in the logic and syntax of their code submissions and offers targeted hints that enable students to figure out what went wrong—and why. For instructors, a comprehensive gradebook tracks correct and incorrect answers and stores the code inputted by students for review.

MyProgrammingLab is offered to users of this book in partnership with Turing's Craft, the makers of the CodeLab interactive programming exercise system. For a full demonstration, to see feedback from instructors and students, or to get started using MyProgrammingLab in your course, visit [www.myprogramminglab.com](http://www.myprogramminglab.com).

### Instructor Resources

The following supplements are available only to qualified instructors:

- Answers to all Review Questions in the text
- Solutions for all Programming Challenges in the text
- PowerPoint presentation slides for every chapter
- Computerized test bank
- Answers to all Student Lab Manual questions

- Solutions for all Student Lab Manual programs

Visit the Pearson Instructor Resource Center ([www.pearsonhighered.com/irc](http://www.pearsonhighered.com/irc)) for information on how to access instructor resources.

### Textbook Web site

Student and instructor resources, including links to download Microsoft® Visual Studio Express and other popular IDEs, for all the books in the Gaddis *Starting Out with* series can be accessed at the following URL:

<http://www.pearsonhighered.com/gaddis>

## Which Gaddis C++ book is right for you?

The Starting Out with C++ Series includes three books, one of which is sure to fit your course:

- *Starting Out with C++: From Control Structures through Objects*
- *Starting Out with C++: Early Objects*
- *Starting Out with C++: Brief Version*

The following chart will help you determine which book is right for your course.

<b>■ FROM CONTROL STRUCTURES THROUGH OBJECTS</b> <b>■ BRIEF VERSION</b>	<b>■ EARLY OBJECTS</b>
<p><b>LATE INTRODUCTION OF OBJECTS</b></p> <p>Classes are introduced in Chapter 13 of the standard text and Chapter 11 of the brief text, after control structures, functions, arrays, and pointers. Advanced OOP topics, such as inheritance and polymorphism, are covered in the following two chapters.</p> <p><b>INTRODUCTION OF DATA STRUCTURES AND RECURSION</b></p> <p>Linked lists, stacks and queues, and binary trees are introduced in the final chapters of the standard text. Recursion is covered after stacks and queues, but before binary trees. These topics are not covered in the brief text, though it does have appendices dealing with linked lists and recursion.</p>	<p><b>EARLIER INTRODUCTION OF OBJECTS</b></p> <p>Classes are introduced in Chapter 7, after control structures and functions, but before arrays and pointers. Their use is then integrated into the remainder of the text. Advanced OOP topics, such as inheritance and polymorphism, are covered in Chapters 11 and 15.</p> <p><b>INTRODUCTION OF DATA STRUCTURES AND RECURSION</b></p> <p>Linked lists, stacks and queues, and binary trees are introduced in the final chapters of the text, after the chapter on recursion.</p>

## Acknowledgments

There have been many helping hands in the development and publication of this text. We would like to thank the following faculty reviewers for their helpful suggestions and expertise.

### Reviewers for the 9th Edition

Svetlana Marzelli

*Atlantic Cape Community College*

Jie Meichsner

*St. Cloud State University*

Ron Del Porto

*Penn State Erie, The Behrend College*

Lisa Rudnitsky

*Baruch College*

### Reviewers for Previous Editions

Ahmad Abuhejleh

*University of Wisconsin—River Falls*

David Akins

*El Camino College*

Steve Allan

*Utah State University*

Vicki Allan

*Utah State University*

Karen M. Arlien

*Bismark State College*

Mary Astone

*Troy University*

Ijaz A. Awan

*Savannah State University*

Robert Baird

*Salt Lake Community College*

Don Biggerstaff

*Fayetteville Technical Community College*

Michael Bolton

*Northeastern Oklahoma State University*

Bill Brown

Chia-Chin Chang

*Lakeland College*

William Duncan

*Louisiana State University*

Pranshu Gupta

*DeSales University*

Charles Hardnett

*Gwinnett Technical College*

*Pikes Peak Community College*

Robert Burn

*Diablo Valley College*

Charles Cadenhead

*Richland Community College*

Randall Campbell

*Morningside College*

Wayne Caruolo

*Red Rocks Community College*

Cathi Chambley-Miller

*Aiken Technical College*

C.C. Chao

*Jacksonville State University*

Joseph Chao

*Bowling Green State University*

Royce Curtis

*Western Wisconsin Technical College*

Joseph DeLibero

*Arizona State University*

Michael Dixon

*Sacramento City College*

Jeanne Douglas

*University of Vermont*

Michael Dowell

*Augusta State University*

Qiang Duan

*Penn State University—Abington*

William E. Duncan

*Louisiana State University*



- Daniel Edwards  
*Oblone College*
- Judy Etchison  
*Southern Methodist University*
- Dennis Fairclough  
*Utah Valley State College*
- Xisheng Fang  
*Oblone College*
- Mark Fienup  
*University of Northern Iowa*
- Richard Flint  
*North Central College*
- Ann Ford Tyson  
*Florida State University*
- Jeanette Gibbons  
*South Dakota State University*
- James Gifford  
*University of Wisconsin–Stevens Point*
- Leon Gleiberman  
*Touro College*
- Barbara Guillott  
*Louisiana State University*
- Ranette Halverson, Ph.D.  
*Midwestern State University*
- Ken Hang  
*Green River Community College*
- Carol Hannahs  
*University of Kentucky*
- Dennis Heckman  
*Portland Community College*
- Ric Heishman  
*George Mason University*
- Michael Hennessy  
*University of Oregon*
- Ilga Higbee  
*Black Hawk College*
- Patricia Hines  
*Brookdale Community College*
- Mike Holland  
*Northern Virginia Community College*
- Mary Hovik  
*Lehigh Carbon Community College*
- Richard Hull  
*Lenoir-Rhyne College*
- Kay Johnson  
*Community College of Rhode Island*
- Chris Kardaras  
*North Central College*
- Willard Keeling  
*Blue Ridge Community College*
- A.J. Krygeris  
*Houston Community College*
- Sheila Lancaster  
*Gadsden State Community College*
- Ray Larson  
*Inver Hills Community College*
- Michelle Levine  
*Broward College*
- Jennifer Li  
*Oblone College*
- Norman H. Liebling  
*San Jacinto College*
- Cindy Lindstrom  
*Lakeland College*
- Zhu-qu Lu  
*University of Maine, Presque Isle*
- Heidar Malki  
*University of Houston*
- Debbie Mathews  
*J. Sargeant Reynolds Community College*
- Rick Matzen  
*Northeastern State University*
- Robert McDonald  
*East Stroudsburg University*
- James McGuffee

*Austin Community College*

Dean Mellas  
*Cerritos College*

Lisa Milkowski  
*Milwaukee School of Engineering*

Marguerite Nedreberg  
*Youngstown State University*

Lynne O’Hanlon  
*Los Angeles Pierce College*

Frank Paiano  
*Southwestern Community College*

Theresa Park  
*Texas State Technical College*

Mark Parker  
*Shoreline Community College*

Tino Posillico  
*SUNY Farmingdale*

Frederick Pratter  
*Eastern Oregon University*

Susan L. Quick  
*Penn State University*

Alberto Ramon  
*Diablo Valley College*

Bazlur Rasheed  
*Sault College of Applied Arts and  
Technology*

Farshad Ravanshad  
*Bergen Community College*

Susan Reeder  
*Seattle University*

Sandra Roberts  
*Snead College*

Lopa Roychoudhuri  
*Angelo State University*

Dolly Samson  
*Weber State University*

Ruth Sapir  
*SUNY Farmingdale*

Jason Schatz  
*City College of San Francisco*

Dr. Sung Shin  
*South Dakota State University*

Bari Siddique  
*University of Texas at Brownsville*

William Slater  
*Collin County Community College*

Shep Smithline  
*University of Minnesota*

Richard Snyder  
*Lehigh Carbon Community College*

Donald Southwell  
*Delta College*

Caroline St. Claire  
*North Central College*

Kirk Stephens  
*Southwestern Community College*

Cherie Stevens  
*South Florida Community College*

Dale Suggs  
*Campbell University*

Mark Swanson  
*Red Wing Technical College*

Ann Sudell Thorn  
*Del Mar College*

Martha Tillman  
*College of San Mateo*

Ralph Tomlinson  
*Iowa State University*

David Topham  
*Oklahoma College*

Robert Tureman  
*Paul D. Camp Community College*

Arisa K. Ude  
*Richland College*

Peter van der Goes  
*Rose State College*

Stewart Venit  
*California State University, Los Angeles*

Judy Walters  
*North Central College*

John H. Whipple  
*Northampton Community College*

Aurelia Williams  
*Norfolk State University*

Chadd Williams  
*Pacific University*

Vida Winans  
*Illinois Institute of Technology*

I would also like to thank my family and friends for their support in all of my projects. I am extremely fortunate to have Matt Goldstein as my editor, and Kristy Alaura as editorial assistant. Their guidance and encouragement made it a pleasure to write chapters and meet deadlines. I am also fortunate to have Demetrius Hall as my marketing manager. His hard work is truly inspiring, and he does a great job of getting this book out to the academic community. The production team, led by Sandra Rodriguez, worked tirelessly to make this book a reality. Thanks to you all!

### **About the Author**

Tony Gaddis is the principal author of the *Starting Out with* series of textbooks. He has nearly two decades of experience teaching computer science courses, primarily at Haywood Community College. Tony is a highly acclaimed instructor who was previously selected as the North Carolina Community College Teacher of the Year and has received the Teaching Excellence award from the National Institute for Staff and Organizational Development. The *Starting Out with* series includes introductory textbooks covering Programming Logic and Design, Alice, C++, Java™, Microsoft® Visual Basic®, Microsoft® Visual C#, Python, and App Inventor, all published by Pearson.

# MyProgrammingLab™

Through the power of practice and immediate personalized feedback, MyProgrammingLab helps improve your students' performance.

## PROGRAMMING PRACTICE

With MyProgrammingLab, your students will gain first-hand programming experience in an interactive online environment.

## IMMEDIATE, PERSONALIZED FEEDBACK

MyProgrammingLab automatically detects errors in the logic and syntax of their code submission and offers targeted hints that enables students to figure out what went wrong and why.

## GRADUATED COMPLEXITY

MyProgrammingLab breaks down programming concepts into short, understandable sequences of exercises. Within each sequence the level and sophistication of the exercises increase gradually but steadily.

## DYNAMIC ROSTER

Students' submissions are stored in a roster that indicates whether the submission is correct, how many attempts were made, and the actual code submissions from each attempt.

## PEARSON eTEXT

The Pearson eText gives students access to their textbook anytime, anywhere.

## STEP-BY-STEP VIDEONOTE TUTORIALS

These step-by-step video tutorials enhance the programming concepts presented in select Pearson textbooks.

For more information and titles available with **MyProgrammingLab**, please visit [www.myprogramminglab.com](http://www.myprogramminglab.com).

Copyright © 2018 Pearson Education, Inc. or its affiliate(s). All rights reserved. HELO88173 • 11/15

