

GLOBAL
EDITION



Starting Out With C++

From Control Structures Through Objects

BRIEF VERSION

EIGHTH EDITION

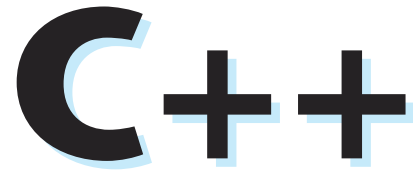
Tony Gaddis



ALWAYS LEARNING

PEARSON

STARTING OUT WITH

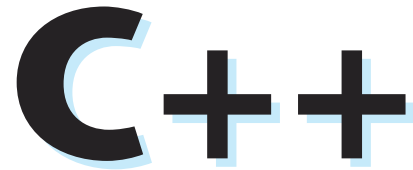


From Control Structures
through Objects

**BRIEF VERSION
EIGHTH EDITION
GLOBAL EDITION**

This page intentionally left blank

STARTING OUT WITH



From Control Structures
through Objects

**BRIEF VERSION
EIGHTH EDITION
GLOBAL EDITION**

Tony Gaddis

Haywood Community College

PEARSON

Boston Columbus Indianapolis New York San Francisco Hoboken Amsterdam
Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto Delhi
Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: Marcia Horton
Acquisitions Editor: Matt Goldstein
Assistant Acquisitions Editor, Global Editions: Aditee
Agarwal
Program Manager: Carole Snyder
Product Marketing Manager: Bram
van Kempen
Field Marketing Manager: Demetrius Hall
Marketing Assistant: Jon Bryant
Senior Managing Editor: Scott Disanno
Senior Project Manager: Camille Trentacoste
Project Editor, Global Editions: K.K. Neelakantan
Operations Supervisor: Vincent Scelta
Operations Specialist: Maura Zaldivar-Garcia

Senior Manufacturing Controller, Global Editions:
Trudy Kimber
Text Designer: Joyce Cosentino Wells
Cover Designer: Joyce Cosentino Wells
Manager, Visual Research: Karen Sanatar
Permissions Supervisor: Michael Joyce
Permission Administrator: Jenell Forschler
Cover Image: © Anthony Ricci/Shutterstock
Media Project Manager: Renata Butera
Media Production Manager, Global Editions:
Vikram Kumar
Full-Service Project Manager: Rashmi Tickyani
Aptara®, Inc.
Full-Service Vendor: Aptara®, Inc.

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:
www.pearsonglobaleditions.com

© Pearson Education Limited 2016

The right of Tony Gaddis to be identified as the authors of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

Authorized adaptation from the United States edition, entitled Starting Out With C++ : From Control Structures Through Objects, Brief Version, 8th Edition, ISBN 978-0-13-403732-5, by Tony Gaddis published by Pearson Education © 2016.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a license permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, Saffron House, 6–10 Kirby Street, London EC1N 8TS.

All trademarks used herein are the property of their respective owners. The use of any trademark in this text does not vest in the author or publisher any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of this book by such owners.

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

ISBN 10: 1-292-11942-X
ISBN 13: 978-1-292-11942-7

Typeset by Aptara®, Inc.

Printed in Malaysia.

Contents at a Glance

	Preface	15
CHAPTER 1	Introduction to Computers and Programming	31
CHAPTER 2	Introduction to C++	57
CHAPTER 3	Expressions and Interactivity	113
CHAPTER 4	Making Decisions	179
CHAPTER 5	Loops and Files	257
CHAPTER 6	Functions	329
CHAPTER 7	Arrays	405
CHAPTER 8	Searching and Sorting Arrays	487
CHAPTER 9	Pointers	525
CHAPTER 10	Characters, C-Strings, and More About the <code>string</code> Class	577
CHAPTER 11	Structured Data	629
CHAPTER 12	Advanced File Operations	687
CHAPTER 13	Introduction to Classes	741
CHAPTER 14	More About Classes	841
CHAPTER 15	Inheritance, Polymorphism, and Virtual Functions	921
	Appendix A: Getting Started with Alice	1001
	Appendix B: The ASCII Character Set	1027
	Appendix C: Operator Precedence and Associativity	1029
	Quick References	1031
	Index	1033
	Credit	1049
Online	The following appendices are available at www.pearsonglobaleditions.com/gaddis .	
	Appendix D: Introduction to Flowcharting	
	Appendix E: Using UML in Class Design	
	Appendix F: Namespaces	

Appendix G: Passing Command Line Arguments

Appendix H: Header File and Library Function Reference

Appendix I: Binary Numbers and Bitwise Operations

Appendix J: Multi-Source File Programs

Appendix K: Stream Member Functions for Formatting

Appendix L: Answers to Checkpoints

Appendix M: Solutions to Odd-Numbered Review Questions

Contents

Preface 15

CHAPTER 1 Introduction to Computers and Programming 31

- 1.1 Why Program? 31
- 1.2 Computer Systems: Hardware and Software 32
- 1.3 Programs and Programming Languages 38
- 1.4 What Is a Program Made of? 44
- 1.5 Input, Processing, and Output 47
- 1.6 The Programming Process 48
- 1.7 Procedural and Object-Oriented Programming 52

CHAPTER 2 Introduction to C++ 57

- 2.1 The Parts of a C++ Program 57
- 2.2 The `cout` Object 61
- 2.3 The `#include` Directive 66
- 2.4 Variables and Literals 67
- 2.5 Identifiers 71
- 2.6 Integer Data Types 72
- 2.7 The `char` Data Type 78
- 2.8 The C++ `string` Class 82
- 2.9 Floating-Point Data Types 84
- 2.10 The `bool` Data Type 87
- 2.11 Determining the Size of a Data Type 88
- 2.12 Variable Assignments and Initialization 89
- 2.13 Scope 91
- 2.14 Arithmetic Operators 91
- 2.15 Comments 99
- 2.16 Named Constants 101
- 2.17 Programming Style 103

CHAPTER 3 Expressions and Interactivity 113

- 3.1 The `cin` Object 113
- 3.2 Mathematical Expressions 119
- 3.3 When You Mix Apples and Oranges: Type Conversion 128
- 3.4 Overflow and Underflow 130
- 3.5 Type Casting 131
- 3.6 Multiple Assignment and Combined Assignment 134
- 3.7 Formatting Output 138
- 3.8 Working with Characters and `string` Objects 148
- 3.9 More Mathematical Library Functions 154
- 3.10 Focus on Debugging: Hand Tracing a Program 160
- 3.11 Focus on Problem Solving: A Case Study 162

CHAPTER 4 Making Decisions 179

- 4.1 Relational Operators 179
- 4.2 The `if` Statement 184
- 4.3 Expanding the `if` Statement 192
- 4.4 The `if/else` Statement 196
- 4.5 Nested `if` Statements 199
- 4.6 The `if/else if` Statement 206
- 4.7 Flags 211
- 4.8 Logical Operators 212
- 4.9 Checking Numeric Ranges with Logical Operators 219
- 4.10 Menus 220
- 4.11 Focus on Software Engineering: Validating User Input 223
- 4.12 Comparing Characters and Strings 225
- 4.13 The Conditional Operator 229
- 4.14 The `switch` Statement 232
- 4.15 More About Blocks and Variable Scope 241

CHAPTER 5 Loops and Files 257

- 5.1 The Increment and Decrement Operators 257
- 5.2 Introduction to Loops: The `while` Loop 262
- 5.3 Using the `while` Loop for Input Validation 269
- 5.4 Counters 271
- 5.5 The `do-while` Loop 272
- 5.6 The `for` Loop 277
- 5.7 Keeping a Running Total 287
- 5.8 Sentinels 290
- 5.9 Focus on Software Engineering: Deciding Which Loop to Use 291
- 5.10 Nested Loops 292
- 5.11 Using Files for Data Storage 295
- 5.12 Optional Topics: Breaking and Continuing a Loop 314

CHAPTER 6 Functions 329

- 6.1 Focus on Software Engineering: Modular Programming 329
- 6.2 Defining and Calling Functions 330
- 6.3 Function Prototypes 339
- 6.4 Sending Data into a Function 341

- 6.5 Passing Data by Value 346
- 6.6 Focus on Software Engineering: Using Functions in a Menu-Driven Program 348
- 6.7 The `return` Statement 352
- 6.8 Returning a Value from a Function 354
- 6.9 Returning a Boolean Value 362
- 6.10 Local and Global Variables 364
- 6.11 Static Local Variables 372
- 6.12 Default Arguments 375
- 6.13 Using Reference Variables as Parameters 378
- 6.14 Overloading Functions 384
- 6.15 The `exit()` Function 388
- 6.16 Stubs and Drivers 391

CHAPTER 7 Arrays 405

- 7.1 Arrays Hold Multiple Values 405
- 7.2 Accessing Array Elements 407
- 7.3 No Bounds Checking in C++ 414
- 7.4 Array Initialization 417
- 7.5 The Range-Based `for` Loop 422
- 7.6 Processing Array Contents 426
- 7.7 Focus on Software Engineering: Using Parallel Arrays 434
- 7.8 Arrays as Function Arguments 437
- 7.9 Two-Dimensional Arrays 448
- 7.10 Arrays with Three or More Dimensions 455
- 7.11 Focus on Problem Solving and Program Design: A Case Study 457
- 7.12 If You Plan to Continue in Computer Science: Introduction to the STL `vector` 459

CHAPTER 8 Searching and Sorting Arrays 487

- 8.1 Focus on Software Engineering: Introduction to Search Algorithms 487
- 8.2 Focus on Problem Solving and Program Design: A Case Study 493
- 8.3 Focus on Software Engineering: Introduction to Sorting Algorithms 500
- 8.4 Focus on Problem Solving and Program Design: A Case Study 507
- 8.5 If You Plan to Continue in Computer Science: Sorting and Searching `vectors` 515

CHAPTER 9 Pointers 525

- 9.1 Getting the Address of a Variable 525
- 9.2 Pointer Variables 527
- 9.3 The Relationship Between Arrays and Pointers 534
- 9.4 Pointer Arithmetic 538
- 9.5 Initializing Pointers 540
- 9.6 Comparing Pointers 541
- 9.7 Pointers as Function Parameters 543
- 9.8 Focus on Software Engineering: Dynamic Memory Allocation 552
- 9.9 Focus on Software Engineering: Returning Pointers from Functions 556
- 9.10 Using Smart Pointers to Avoid Memory Leaks 563
- 9.11 Focus on Problem Solving and Program Design: A Case Study 566

CHAPTER 10 Characters, C-Strings, and More About the string Class 577

- 10.1 Character Testing 577
- 10.2 Character Case Conversion 581
- 10.3 C-Strings 584
- 10.4 Library Functions for Working with C-Strings 588
- 10.5 C-String/Numeric Conversion Functions 599
- 10.6 Focus on Software Engineering: Writing Your Own C-String-Handling Functions 605
- 10.7 More About the C++ string Class 611
- 10.8 Focus on Problem Solving and Program Design: A Case Study 620

CHAPTER 11 Structured Data 629

- 11.1 Abstract Data Types 629
- 11.2 Focus on Software Engineering: Combining Data into Structures 631
- 11.3 Accessing Structure Members 634
- 11.4 Initializing a Structure 638
- 11.5 Arrays of Structures 641
- 11.6 Focus on Software Engineering: Nested Structures 643
- 11.7 Structures as Function Arguments 647
- 11.8 Returning a Structure from a Function 650
- 11.9 Pointers to Structures 653
- 11.10 Focus on Software Engineering: When to Use ., When to Use ->, and When to Use * 656
- 11.11 Unions 658
- 11.12 Enumerated Data Types 662

CHAPTER 12 Advanced File Operations 687

- 12.1 File Operations 687
- 12.2 File Output Formatting 693
- 12.3 Passing File Stream Objects to Functions 695
- 12.4 More Detailed Error Testing 697
- 12.5 Member Functions for Reading and Writing Files 700
- 12.6 Focus on Software Engineering: Working with Multiple Files 708
- 12.7 Binary Files 710
- 12.8 Creating Records with Structures 715
- 12.9 Random-Access Files 719
- 12.10 Opening a File for Both Input and Output 727

CHAPTER 13 Introduction to Classes 741

- 13.1 Procedural and Object-Oriented Programming 741
- 13.2 Introduction to Classes 748
- 13.3 Defining an Instance of a Class 753
- 13.4 Why Have Private Members? 766
- 13.5 Focus on Software Engineering: Separating Class Specification from Implementation 767
- 13.6 Inline Member Functions 773
- 13.7 Constructors 776
- 13.8 Passing Arguments to Constructors 780

- 13.9 Destructors 788
- 13.10 Overloading Constructors 792
- 13.11 Private Member Functions 795
- 13.12 Arrays of Objects 797
- 13.13 Focus on Problem Solving and Program Design: An OOP Case Study 801
- 13.14 Focus on Object-Oriented Programming: Simulating Dice with Objects 808
- 13.15 Focus on Object-Oriented Programming: Creating an Abstract Array
Data Type 812
- 13.16 Focus on Object-Oriented Design: The Unified Modeling Language (UML) 815
- 13.17 Focus on Object-Oriented Design: Finding the Classes and Their
Responsibilities 818

CHAPTER 14 More About Classes 841

- 14.1 Instance and Static Members 841
- 14.2 Friends of Classes 849
- 14.3 Memberwise Assignment 854
- 14.4 Copy Constructors 855
- 14.5 Operator Overloading 861
- 14.6 Object Conversion 888
- 14.7 Aggregation 890
- 14.8 Focus on Object-Oriented Design: Class Collaborations 895
- 14.9 Focus on Object-Oriented Programming: Simulating the Game
of Cho-Han 899

CHAPTER 15 Inheritance, Polymorphism, and Virtual Functions 921

- 15.1 What Is Inheritance? 921
- 15.2 Protected Members and Class Access 930
- 15.3 Constructors and Destructors in Base and Derived Classes 936
- 15.4 Redefining Base Class Functions 948
- 15.5 Class Hierarchies 953
- 15.6 Polymorphism and Virtual Member Functions 959
- 15.7 Abstract Base Classes and Pure Virtual Functions 975
- 15.8 Multiple Inheritance 982

Appendix A: Getting Started with Alice 1001

Appendix B: The ASCII Character Set 1027

Appendix C: Operator Precedence and Associativity 1029

Quick References 1031

Index 1033

Credit 1049

Online The following appendices are available at www.pearsonglobaleditions.com/gaddis.

Appendix D: Introduction to Flowcharting

Appendix E: Using UML in Class Design

Appendix F: Namespaces

Appendix G: Passing Command Line Arguments

Appendix H: Header File and Library Function Reference

Appendix I: Binary Numbers and Bitwise Operations

Appendix J: Multi-Source File Programs

Appendix K: Stream Member Functions for Formatting

Appendix L: Answers to Checkpoints

Appendix M: Solutions to Odd-Numbered Review Questions

LOCATION OF VIDEONOTES IN THE TEXT



Chapter 1	Introduction to Flowcharting, p. 50 Designing a Program with Pseudocode, p. 50 Designing the Account Balance Program, p. 55 Predicting the Result of Problem 33, p. 56
Chapter 2	Using <code>cout</code> , p. 61 Variable Definitions, p. 67 Assignment Statements and Simple Math Expressions, p. 92 Solving the Restaurant Bill Problem, p. 110
Chapter 3	Reading Input with <code>cin</code> , p. 113 Formatting Numbers with <code>setprecision</code> , p. 141 Solving the Stadium Seating Problem, p. 172
Chapter 4	The <code>if</code> Statement, p. 184 The <code>if/else</code> statement, p. 196 The <code>if/else if</code> Statement, p. 206 Solving the Time Calculator Problem, p. 251
Chapter 5	The <code>while</code> Loop, p. 262 The <code>for</code> Loop, p. 277 Reading Data from a File, p. 304 Solving the Calories Burned Problem, p. 323
Chapter 6	Functions and Arguments, p. 341 Value-Returning Functions, p. 354 Solving the Markup Problem, p. 396
Chapter 7	Accessing Array Elements With a Loop, p. 410 Passing an Array to a Function, p. 437 Solving the Chips and Salsa Problem, p. 478
Chapter 8	The Binary Search, p. 490 The Selection Sort, p. 504 Solving the Charge Account Validation Modification Problem, p. 522
Chapter 9	Dynamically Allocating an Array, p. 553 Solving the Pointer Rewrite Problem, p. 575
Chapter 10	Writing a C-String-Handling Function, p. 605 More About the <code>string</code> Class, p. 611 Solving the Backward String Problem, p. 624

(continued on the next page)

LOCATION OF VIDEONOTES IN THE TEXT *(continued)*



Chapter 11	Creating a Structure, p. 631 Passing a Structure to a Function, p. 647 Solving the Weather Statistics Problem, p. 682
Chapter 12	Passing File Stream Objects to Functions, p. 695 Working with Multiple Files, p. 708 Solving the File Encryption Filter Problem, p. 738
Chapter 13	Writing a Class, p. 748 Defining an Instance of a Class, p. 753 Solving the <code>Employee</code> Class Problem, p. 832
Chapter 14	Operator Overloading, p. 861 Class Aggregation, p. 890 Solving the <code>NumDays</code> Problem, p. 915
Chapter 15	Redefining a Base Class Function in a Derived Class, p. 948 Polymorphism, p. 959 Solving the <code>Employee</code> and <code>Production-Worker</code> Classes Problem, p. 993

Preface

Welcome to the Brief Version of *Starting Out with C++: From Control Structures through Objects, 8th edition*. This book is intended for use in a one or 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 Eighth Edition

C++11 is the latest standard version of the C++ language. In previous years, while the standard was being developed, it was known as C++0x. In August 2011, it was approved by the International Standards Organization (ISO), and the name of the standard was officially changed to C++11. Most of the popular compilers now support the C++11 standard.

The new C++11 standard was the primary motivation behind this edition. Although this edition introduces many of the new language features, a C++11 compiler is not strictly required to use the book. As you progress through the book, you will see C++11 icons in the margins, next to the new features that are introduced. Programs appearing in sections that are not marked with this icon will still compile using an older compiler.

Here is a summary of the new C++11 topics that are introduced in this edition:

- The `auto` key word is introduced as a way to simplify complex variable definitions. The `auto` key word causes the compiler to infer a variable's data type from its initialization value.
- The `long long int` and `unsigned long long int` data types, and the `LL` literal suffix are introduced.
- Chapter 5 shows how to pass a `string` object directly to a file stream object's `open` member function, without the need to call the `c_str()` member function. (A discussion of the `c_str()` function still exists for anyone using a legacy compiler.)

- The range-based `for` loop is introduced in Chapter 7. This new looping mechanism automatically iterates over each element of an array, `vector`, or other collection, without the need of a counter variable or a subscript.
- Chapter 7 shows how a `vector` can be initialized with an initialization list.
- The `nullptr` key word is introduced as the standard way of representing a null pointer.
- Smart pointers are introduced in Chapter 9, with an example of dynamic memory allocation using `unique_ptr`.
- Chapter 10 discusses the new, overloaded `to_string` functions for converting numeric values to `string` objects.
- The `string` class's new `back()` and `front()` member functions are included in Chapter 10's overview of the `string` class.
- Strongly typed enums are discussed in Chapter 11.
- Chapter 13 shows how to use the smart pointer `unique_ptr` to dynamically allocate an object.
- Chapter 15 discusses the `override` key word and demonstrates how it can help prevent subtle overriding errors. The `final` key word is discussed as a way of preventing a virtual member function from being overridden.

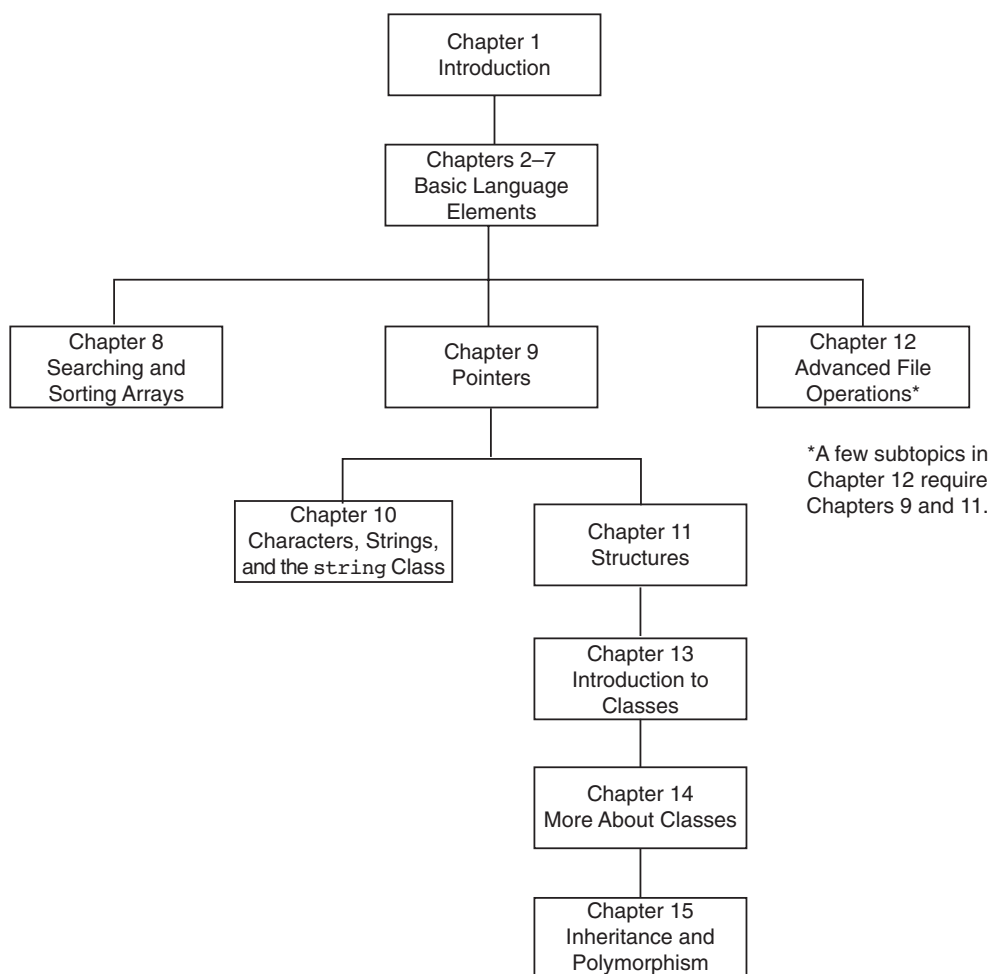
In addition to the C++11 topics, the following general improvements were made:

- Several new programming problems have been added to the text, and many of the existing programming problems have been modified to make them unique from previous editions.
- The discussion of early, historic computers in Chapter 1 is expanded.
- The discussion of literal values in Chapter 2 is improved.
- The introduction of the `char` data type in Chapter 2 is reorganized to use character literals in variable assignments before using ASCII values in variable assignments.
- The discussion of random numbers in Chapter 3 is expanded and improved, with the addition of a new *In the Spotlight* section.
- A new *Focus on Object-Oriented Programming* section has been added to Chapter 13, showing how to write a class that simulates dice.
- A new *Focus on Object-Oriented Programming* section has been added to Chapter 14, showing an object-oriented program that simulates the game of Cho-Han. The program uses objects for the dealer, two players, and a pair of dice.

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.

Figure P-1



Chapter 1 covers fundamental hardware, software, and programming concepts. You may choose to skip this chapter if the class has already mastered 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 either Chapter 9 or Chapter 12. (If you jump to Chapter 12 at this point, you will need to postpone sections 12.7, 12.8, and 12.10 until Chapters 9 and 11 have been covered.)

After Chapter 9 has been covered, either of Chapters 10 or 11 may be covered. After Chapter 11, you may cover Chapters 13 through 15 in sequence.

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

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. An optional section explains the difference between ANSI standard and pre-standard C++ programs.

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

In this chapter the student learns to create and work with single and multidimensional 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: Sorting and Searching 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: Getting Started with Alice

This appendix gives a quick introduction to Alice. Alice is free software that can be used to teach fundamental programming concepts using 3D graphics.

Appendix B: ASCII Character Set

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

Appendix C: Operator Precedence and Associativity

A chart showing the C++ operators and their precedence.

The following appendices are available online at www.pearsonglobaleditions.com/gaddis.

Appendix D: Introduction to Flowcharting

A brief introduction to flowcharting. This tutorial discusses sequence, selection, case, repetition, and module structures.

Appendix E: Using UML in Class Design

This appendix shows the student how to use the Unified Modeling Language to design classes. Notation for showing access specification, data types, parameters, return values, overloaded functions, composition, and inheritance are included.

Appendix F: Namespaces

This appendix explains namespaces and their purpose. Examples showing how to define a namespace and access its members are given.

Appendix G: Passing Command Line Arguments

Teaches the student how to write a C++ program that accepts arguments from the command line. This appendix will be useful to students working in a command line environment, such as Unix, Linux, or the Windows command prompt.

Appendix H: Header File and Library Function Reference

This appendix provides a reference for the C++ library functions and header files discussed in the book.

Appendix I: Binary Numbers and Bitwise Operations

A guide to the C++ bitwise operators, as well as a tutorial on the internal storage of integers.

Appendix J: Multi-Source File Programs

Provides a tutorial on creating programs that consist of multiple source files. Function header files, class specification files, and class implementation files are discussed.

Appendix K: Stream Member Functions for Formatting

Covers stream member functions for formatting such as `setf`.

Appendix L: Answers to Checkpoints

Students may test their own progress by comparing their answers to the checkpoint exercises against this appendix. The answers to all Checkpoints are included.

Appendix M: Solutions to Odd-Numbered Review Questions

Another tool that students can use to gauge their progress.

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.pearsonglobaleditions.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 Companion Web site at www.pearsonglobaleditions.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 Companion Web site at www.pearsonglobaleditions.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 Companion Web site at www.pearsonglobaleditions.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 last two pages of Appendix C in 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 Gaddis Series Companion Web site at www.pearsonglobaleditions.com/gaddis:

- The source code for each example program in the book
- Access to the book's companion 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.

Instructor Resources

The following supplements are available to qualified instructors only:

- 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.pearsonglobaleditions.com/gaddis) 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.pearsonglobaleditions.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.

■ FROM CONTROL STRUCTURES THROUGH OBJECTS ■ BRIEF VERSION	■ EARLY OBJECTS
<p>LATE INTRODUCTION OF OBJECTS 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>INTRODUCTION OF DATA STRUCTURES AND RECURSION 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>EARLIER INTRODUCTION OF OBJECTS 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>INTRODUCTION OF DATA STRUCTURES AND RECURSION 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 8th Edition

Robert Burn
Diablo Valley College

Michael Dixon
Sacramento City College

Qiang Duan
Penn State University—Abington

Daniel Edwards
Ohlone College

Xisheng Fang
Ohlone College

Ken Hang
Green River Community College

Kay Johnson
Community College of Rhode Island

Michelle Levine
Broward College

Cindy Lindstrom
Lakeland College

Susan Reeder
Seattle University

Sandra Roberts
Snead College

Lopa Roychoudhuri
Angelo State University

Richard Snyder
Lehigh Carbon Community College

Donald Southwell
Delta College

Chadd Williams
Pacific University

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
Pikes Peak Community 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

Jeanne Douglas
University of Vermont

Michael Dowell
Augusta State U

William E. Duncan
Louisiana State University

Judy Etchison
Southern Methodist University

Dennis Fairclough
Utah Valley State 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

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

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

Jennifer Li
Ohlone College

Norman H. Liebling
San Jacinto 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
- 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
- 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
Oblone 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
- Vida Winans
Illinois Institute of Technology

I would like to thank my family for their love and support in all of my many projects. I am extremely fortunate to have Matt Goldstein as my editor. He and Kelsey Loanes, Editorial Assistant, guided me through the delicate process of revising the book. I am also fortunate to have Demetrius Hall and Bram Van Kempen as marketing managers. They do a great job getting my books out to the academic community. I had a great production team led by Camille Trentacoste. Thanks to you all!

Pearson would like to thank and acknowledge the following people for their work on the Global Edition.

Contributor

Moumita Mitra Manna
Bangabasi College

Reviewers

Shaligram Prajapat
Devi Ahilya University

Vikas Saxena
Jaypee Institute of Information Technology

Andres Baravelle
University of East London

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.

BREAKTHROUGH
To improving results



get with the programming

Through the power of practice and immediate personalized feedback, MyProgrammingLab improves your performance.

MyProgrammingLab™

Learn more at www.myprogramminglab.com

This page intentionally left blank