GLOBAL EDITION

Absolute C++

SIXTH EDITION

Walter Savitch

ALWAYS LEARNING

PEARSON

ABSOLUTE C++

6TH EDITION

GLOBAL EDITION

Absolute C++, Global Edition

Table of Contents

Cover

Title Page

Copyright Page

Preface

Acknowledgments

Brief Contents

Contents

Chapter 1 C++ Basics

1.1 INTRODUCTION TO C++

Origins of the C++ Language

C++ and Object-Oriented Programming

The Character of C++

C++ Terminology

A Sample C++ Program

TIP: Compiling C++ Programs

1.2 VARIABLES, EXPRESSIONS, AND ASSIGNMENT STATEMENTS

Identifiers

Variables

Assignment Statements

Introduction to the string class

PITFALL: Uninitialized Variables

TIP: Use Meaningful Names

More Assignment Statements

Table of Contents

- Assignment Compatibility
- Literals
- Escape Sequences
- Raw String Literals
- Naming Constants
- Arithmetic Operators and Expressions
- Integer and Floating-Point Division
- PITFALL: Division with Whole Numbers
- Type Casting
- Increment and Decrement Operators
- PITFALL: Order of Evaluation

1.3 CONSOLE INPUT/OUTPUT

- Output Using cout
- New Lines in Output
- TIP: End Each Program with
  or endl
- Formatting for Numbers with a Decimal Point
- Output with cerr
- Input Using cin
- TIP: Line Breaks in I/O

1.4 PROGRAM STYLE

- Comments

1.5 LIBRARIES AND NAMESPACES

- Libraries and include Directives
- Namespaces
- PITFALL: Problems with Library Names
- Chapter Summary
- Answers to Self-Test Exercises
- Programming Projects

Table of Contents

Chapter 2 Flow of Control

2.1 BOOLEAN EXPRESSIONS

Building Boolean Expressions

PITFALL: Strings of Inequalities

Evaluating Boolean Expressions

Precedence Rules

PITFALL: Integer Values Can Be Used as Boolean Values

2.2 BRANCHING MECHANISMS

if-else Statements

Compound Statements

PITFALL: Using = in Place of = =

Omitting the else

Nested Statements

Multiway if-else Statement

The switch Statement

PITFALL: Forgetting a break in a switch Statement

TIP: Use switch Statements for Menus

Enumeration Types

The Conditional Operator

2.3 LOOPS

The while and do-while Statements

Increment and Decrement Operators Revisited

The Comma Operator

The for Statement

TIP: Repeat-N-Times Loops

PITFALL: Extra Semicolon in a for Statement

PITFALL: Infinite Loops

The break and continue Statements

Nested Loops

Table of Contents

2.4 INTRODUCTION TO FILE INPUT

Reading From a Text File Using ifstream

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 3 Function Basics

3.1 PREDEFINED FUNCTIONS

Predefined Functions That Return a Value

Predefined void Functions

A Random Number Generator

3.2 PROGRAMMER-DEFINED FUNCTIONS

Defining Functions That Return a Value

Alternate Form for Function Declarations

PITFALL: Arguments in the Wrong Order

PITFALL: Use of the Terms Parameter and Argument

Functions Calling Functions

EXAMPLE: A Rounding Function

Functions That Return a Boolean Value

Defining void Functions

return Statements in void Functions

Preconditions and Postconditions

main Is a Function

Recursive Functions

3.3 SCOPE RULES

Local Variables

Procedural Abstraction

Global Constants and Global Variables

Blocks

Nested Scopes

Table of Contents

TIP: Use Function Calls in Branching and Loop Statements

Variables Declared in a for Loop

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 4 Parameters and Overloading

4.1 PARAMETERS

Call-by-Value Parameters

A First Look at Call-by-Reference Parameters

Call-by-Reference Mechanism in Detail

Constant Reference Parameters

EXAMPLE: The swapValues Function

TIP: Think of Actions, Not Code

Mixed Parameter Lists

TIP: What Kind of Parameter to Use

PITFALL: Inadvertent Local Variables

TIP: Choosing Formal Parameter Names

EXAMPLE: Buying Pizza

4.2 OVERLOADING AND DEFAULT ARGUMENTS

Introduction to Overloading

PITFALL: Automatic Type Conversion and Overloading

Rules for Resolving Overloading

EXAMPLE: Revised Pizza-Buying Program

Default Arguments

4.3 TESTING AND DEBUGGING FUNCTIONS

The assert Macro

Stubs and Drivers

Chapter Summary

Answers to Self-Test Exercises

Table of Contents

Programming Projects

Chapter 5 Arrays

5.1 INTRODUCTION TO ARRAYS

Declaring and Referencing Arrays

TIP: Use for Loops with Arrays

PITFALL: Array Indexes Always Start with Zero

TIP: Use a Defined Constant for the Size of an Array

Arrays in Memory

PITFALL: Array Index out of Range

The Range-Based for Loop

Initializing Arrays

5.2 ARRAYS IN FUNCTIONS

Indexed Variables as Function Arguments

Entire Arrays as Function Arguments

The const Parameter Modifier

PITFALL: Inconsistent Use of const Parameters

Functions That Return an Array

EXAMPLE: Production Graph

5.3 PROGRAMMING WITH ARRAYS

Partially Filled Arrays

TIP: Do Not Skimp on Formal Parameters

EXAMPLE: Searching an Array

EXAMPLE: Sorting an Array

EXAMPLE: Bubble Sort

5.4 MULTIDIMENSIONAL ARRAYS

Multidimensional Array Basics

Multidimensional Array Parameters

EXAMPLE: Two-Dimensional Grading Program

Chapter Summary

Table of Contents

Answers to Self-Test Exercises

Programming Projects

Chapter 6 Structures and Classes

6.1 STRUCTURES

Structure Types

PITFALL: Forgetting a Semicolon in a Structure Definition

Structures as Function Arguments

TIP: Use Hierarchical Structures

Initializing Structures

6.2 CLASSES

Defining Classes and Member Functions

Encapsulation

Public and Private Members

Accessor and Mutator Functions

TIP: Separate Interface and Implementation

TIP: A Test for Encapsulation

Structures versus Classes

TIP: Thinking Objects

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 7 Constructors and Other Tools

7.1 CONSTRUCTORS

Constructor Definitions

PITFALL: Constructors with No Arguments

Explicit Constructor Calls

TIP: Always Include a Default Constructor

EXAMPLE: BankAccount Class

Class Type Member Variables

Table of Contents

Member Initializers and Constructor Delegation in C++11

7.2 MORE TOOLS

The const Parameter Modifier

PITFALL: Inconsistent Use of const

Inline Functions

Static Members

Nested and Local Class Definitions

7.3 VECTORSA PREVIEW OF THE STANDARD TEMPLATE LIBRARY

Vector Basics

PITFALL: Using Square Brackets beyond the Vector Size

TIP: Vector Assignment Is Well Behaved

Efficiency Issues

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 8 Operator Overloading, Friends, and References

8.1 BASIC OPERATOR OVERLOADING

Overloading Basics

TIP: A Constructor Can Return an Object

Returning by const Value

Overloading Unary Operators

Overloading as Member Functions

TIP: A Class Has Access to All Its Objects

Overloading Function Application ( )

PITFALL: Overloading &&,||, and the Comma Operator

8.2 FRIEND FUNCTIONS AND AUTOMATIC TYPE CONVERSION

Constructors for Automatic Type Conversion

PITFALL: Member Operators and Automatic Type Conversion

Friend Functions

Table of Contents

Friend Classes

PITFALL: Compilers without Friends

8.3 REFERENCES AND MORE OVERLOADED OPERATORS

References

TIP: Returning Member Variables of a Class Type

Overloading >> and <<

TIP: What Mode of Returned Value to Use

The Assignment Operator

Overloading the Increment and Decrement Operators

Overloading the Array Operator [ ]

Overloading Based on L-Value versus R-Value

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 9 Strings

9.1 AN ARRAY TYPE FOR STRINGS

C-String Values and C-String Variables

PITFALL: Using = and == with C-strings

Other Functions in <cstring>

EXAMPLE: Command-Line Arguments

C-String Input and Output

9.2 CHARACTER MANIPULATION TOOLS

Character I/O

The Member Functions get and put

EXAMPLE: Checking Input Using a Newline Function

PITFALL: Unexpected '
' in Input

The putback, peek, and ignore Member Functions

Character-Manipulating Functions

Table of Contents

PITFALL: toupper and tolower Return int Values

9.3 THE STANDARD CLASS String

Introduction to the Standard Class string

I/O with the Class string

TIP: More Versions of getline

PITFALL: Mixing cin >> variable

String Processing with the Class string

EXAMPLE: Palindrome Testing

Converting between string Objects and C-Strings

Converting between string Objects and Numbers

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 10 Pointers and Dynamic Arrays

10.1 POINTERS

Pointer Variables

Basic Memory Management

nullptr

PITFALL: Dangling Pointers

Dynamic Variables and Automatic Variables

TIP: Define Pointer Types

PITFALL: Pointers as Call-by-Value Parameters

Uses for Pointers

10.2 DYNAMIC ARRAYS

Array Variables and Pointer Variables

Creating and Using Dynamic Arrays

EXAMPLE: A Function That Returns an Array

Pointer Arithmetic

Multidimensional Dynamic Arrays

Table of Contents

10.3 CLASSES, POINTERS, AND DYNAMIC ARRAYS

The -> Operator

The this Pointer

Overloading the Assignment Operator

EXAMPLE: A Class for Partially Filled Arrays

Destructors

Copy Constructors

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 11 Separate Compilation and Namespaces

11.1 SEPARATE COMPILATION

Encapsulation Reviewed

Header Files and Implementation Files

EXAMPLE: DigitalTime Class

TIP: Reusable Components

Using ifndef

TIP: Defining Other Libraries

11.2 NAMESPACES

Namespaces and using Directives

Creating a Namespace

using Declarations

Qualifying Names

TIP: Choosing a Name for a Namespace

EXAMPLE: A Class Definition in a Namespace

Unnamed Namespaces

PITFALL: Confusing the Global Namespace and the Unnamed Namespace

TIP: Unnamed Namespaces Replace the static Qualifier

TIP: Hiding Helping Functions

Table of Contents

Nested Namespaces

TIP: What Namespace Specification Should You Use?

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 12 Streams and File I/O

12.1 I/O STREAMS

File I/O

PITFALL: Restrictions on Stream Variables

Appending to a File

TIP: Another Syntax for Opening a File

TIP: Check That a File Was Opened Successfully

Character I/O

Checking for the End of a File

12.2 TOOLS FOR STREAM I/O

File Names as Input

Formatting Output with Stream Functions

Manipulators

Saving Flag Settings

More Output Stream Member Functions

EXAMPLE: Cleaning Up a File Format

EXAMPLE: Editing a Text File

12.3 STREAM HIERARCHIES: A PREVIEW OF INHERITANCE

Inheritance among Stream Classes

EXAMPLE: Another newLine Function

Parsing Strings with the stringstream Class

12.4 RANDOM ACCESS TO FILES

Chapter Summary

Answers to Self-Test Exercises

Table of Contents

Programming Projects

Chapter 13 Recursion

13.1 RECURSIVE Void FUNCTIONS

EXAMPLE: Vertical Numbers

Tracing a Recursive Call

A Closer Look at Recursion

PITFALL: Infinite Recursion

Stacks for Recursion

PITFALL: Stack Overflow

Recursion versus Iteration

13.2 RECURSIVE FUNCTIONS THAT RETURN A VALUE

General Form for a Recursive Function That Returns a Value

EXAMPLE: Another Powers Function

Mutual Recursion

13.3 THINKING RECURSIVELY

Recursive Design Techniques

Binary Search

Coding

Checking the Recursion

Efficiency

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 14 Inheritance

14.1 INHERITANCE BASICS

Derived Classes

Constructors in Derived Classes

PITFALL: Use of Private Member Variables from the Base Class

PITFALL: Private Member Functions Are Effectively Not Inherited

Table of Contents

The protected Qualifier

Redefinition of Member Functions

Redefining versus Overloading

Access to a Redefined Base Function

Functions That Are Not Inherited

14.2 PROGRAMMING WITH INHERITANCE

Assignment Operators and Copy Constructors in Derived Classes

Destructors in Derived Classes

EXAMPLE: Partially Filled Array with Backup

PITFALL: Same Object on Both Sides of the Assignment Operator

EXAMPLE: Alternate Implementation of PFArrayDBak

TIP: A Class Has Access to Private Members of All Objects of the Class

TIP: Is a versus Has a

Protected and Private Inheritance

Multiple Inheritance

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 15 Polymorphism and Virtual Functions

15.1 VIRTUAL FUNCTION BASICS

Late Binding

Virtual Functions in C++

Provide Context with C++11s override Keyword

Preventing a Virtual Function from Being Overridden

TIP: The Virtual Property Is Inherited

TIP: When to Use a Virtual Function

PITFALL: Omitting the Definition of a Virtual Member Function

Abstract Classes and Pure Virtual Functions

EXAMPLE: An Abstract Class

Table of Contents

15.2 POINTERS AND VIRTUAL FUNCTIONS
Virtual Functions and Extended Type Compatibility
PITFALL: The Slicing Problem
TIP: Make Destructors Virtual
Downcasting and Upcasting
How C++ Implements Virtual Functions
Chapter Summary
Answers to Self-Test Exercises
Programming Projects
Chapter 16 Templates
16.1 FUNCTION TEMPLATES
Syntax for Function Templates
PITFALL: Compiler Complications
TIP: How to Define Templates
EXAMPLE: A Generic Sorting Function
PITFALL: Using a Template with an Inappropriate Type
16.2 CLASS TEMPLATES
Syntax for Class Templates
EXAMPLE: An Array Template Class
The vector and basic_string Templates
16.3 TEMPLATES AND INHERITANCE
EXAMPLE: Template Class For a Partially Filled Array with Backup
Chapter Summary
Answers to Self-Test Exercises
Programming Projects
Chapter 17 Linked Data Structures
17.1 NODES AND LINKED LISTS
Nodes
Linked Lists

Table of Contents

Inserting a Node at the Head of a List
PITFALL: Losing Nodes
Inserting and Removing Nodes Inside a List
PITFALL: Using the Assignment Operator with Dynamic Data Structures
Searching a Linked List
Doubly Linked Lists
Adding a Node to a Doubly Linked List
Deleting a Node from a Doubly Linked List
EXAMPLE: A Generic Sorting Template Version of Linked List Tools
17.2 LINKED LIST APPLICATIONS
EXAMPLE: A Stack Template Class
EXAMPLE: A Queue Template Class
TIP: A Comment on Namespaces
Friend Classes and Similar Alternatives
EXAMPLE: Hash Tables With Chaining
Efficiency of Hash Tables
EXAMPLE: A Set Template Class
Efficiency of Sets Using Linked Lists
17.3 ITERATORS
Pointers as Iterators
Iterator Classes
EXAMPLE: An Iterator Class
17.4 TREES
Tree Properties
EXAMPLE: A Tree Template Class
Chapter Summary
Answers to Self-Test Exercises
Programming Projects
Chapter 18 Exception Handling

Table of Contents

18.1 EXCEPTION HANDLING BASICS

A Toy Example of Exception Handling

Defining Your Own Exception Classes

Multiple Throws and Catches

PITFALL: Catch the More Specific Exception First

TIP: Exception Classes Can Be Trivial

Throwing an Exception in a Function

EXAMPLE: Returning the High Score

Exception Specification

PITFALL: Exception Specification in Derived Classes

18.2 PROGRAMMING TECHNIQUES FOR EXCEPTION HANDLING

When to Throw an Exception

PITFALL: Uncaught Exceptions

PITFALL: Nested try-catch Blocks

PITFALL: Overuse of Exceptions

Exception Class Hierarchies

Testing for Available Memory

Rethrowing an Exception

Chapter Summary

Answers to Self-Test Exercises

Programming Projects

Chapter 19 Standard Template Library

19.1 ITERATORS

Iterator Basics

PITFALL: Compiler Problems

TIP: Use auto to Simplify Variable Declarations

Kinds of Iterators

Constant and Mutable Iterators

Reverse Iterators

Table of Contents

Other Kinds of Iterators

19.2 CONTAINERS

Sequential Containers

PITFALL: Iterators and Removing Elements

TIP: Type Definitions in Containers

The Container Adapters stack and queue

PITFALL: Underlying Containers

The Associative Containers set and map

Efficiency

TIP: Use Initialization, Ranged for, and auto with Containers

19.3 GENERIC ALGORITHMS

Running Times and Big-O Notation

Container Access Running Times

Nonmodifying Sequence Algorithms

Modifying Sequence Algorithms

Set Algorithms

Sorting Algorithms

Chapter Summary

Answers to Self-Test Exercises

Appendix 1 C++ Keywords

Appendix 2 Precedence of Operators

Appendix 3 The ASCII Character Set

Appendix 4 Some Library Functions

Appendix 5 Old and New Header Files

Appendix 6 Additional C++11 Language Features

Index