

INTERNATIONAL
EDITION



Introduction to Programming with C++

THIRD EDITION

Y. Daniel Liang



ALWAYS LEARNING

PEARSON

ONLINE ACCESS

Thank you for purchasing a new copy of *Introduction to Programming with C++, Third Edition*. Your textbook includes six months of prepaid access to the book's Companion Website. This prepaid subscription provides you with full access to the following student support areas:

- VideoNotes
- Web Chapters
- Student supplements

**Use a coin to scratch off the coating and reveal your student access code.
Do not use a knife or other sharp object as it may damage the code.**

To access the *Introduction to Programming with C++, Third Edition*, Companion Website for the first time, you will need to register online using a computer with an Internet connection and a web browser. The process takes just a couple of minutes and only needs to be completed once.

- 1.** Go to **www.pearsoninternationaleditions.com/liang**
- 2.** Click on **Companion Website**.
- 3.** Click on the **Register** button.
- 4.** On the registration page, enter your student access code* found beneath the scratch-off panel. Do not type the dashes. You can use lower- or uppercase.
- 5.** Follow the on-screen instructions. If you need help at any time during the online registration process, simply click the **Need Help?** icon.
- 6.** Once your personal Login Name and Password are confirmed, you can begin using the *Introduction to Programming with C++* Companion Website!

To log in after you have registered:

You only need to register for this Companion Website once. After that, you can log in any time at **www.pearsoninternationaleditions.com/liang** by providing your Login Name and Password when prompted.

*Important: The access code can only be used once. This subscription is valid for six months upon activation and is not transferable. If this access code has already been revealed, it may no longer be valid.

Introduction to Programming with C++

Table of Contents

Cover

CONTENTS

Chapter 1 Introduction to Computers, Programs, and C++

- 1.1 Introduction
- 1.2 What Is a Computer?
- 1.3 Programming Languages
- 1.4 Operating Systems
- 1.5 History of C++
- 1.6 A Simple C++ Program
- 1.7 C++ Program-Development Cycle
- 1.8 Programming Style and Documentation
- 1.9 Programming Errors

Chapter 2 elementary Programming

- 2.1 Introduction
- 2.2 Writing a Simple Program
- 2.3 Reading Input from the Keyboard
- 2.4 Identifiers
- 2.5 Variables
- 2.6 Assignment Statements and Assignment Expressions
- 2.7 Named Constants
- 2.8 Numeric Data Types and Operations
- 2.9 Evaluating Expressions and Operator Precedence
- 2.10 Case Study: Displaying the Current Time
- 2.11 Augmented Assignment Operators
- 2.12 Increment and Decrement Operators
- 2.13 Numeric Type Conversions

Table of Contents

- 2.14 Software Development Process
- 2.15 Case Study: Counting Monetary Units
- 2.16 Common Errors

Chapter 3 Selections

- 3.1 Introduction
- 3.2 The bool Data Type
- 3.3 if Statements
- 3.4 Two-Way if-else Statements
- 3.5 Nested if and Multi-Way if-else Statements
- 3.6 Common Errors and Pitfalls
- 3.7 Case Study: Computing Body Mass Index
- 3.8 Case Study: Computing Taxes
- 3.9 Generating Random Numbers
- 3.10 Logical Operators
- 3.11 Case Study: Determining Leap Year
- 3.12 Case Study: Lottery
- 3.13 Switch Statements
- 3.14 Conditional Expressions
- 3.15 Operator Precedence and Associativity
- 3.16 Debugging

Chapter 4 Mathematical Functions, Characters, and Strings

- 4.1 Introduction
- 4.2 Mathematical Functions
- 4.3 Character Data Type and Operations
- 4.4 Case Study: Generating Random Characters
- 4.5 Case Study: Guessing Birthdays
- 4.6 Character Functions
- 4.7 Case Study: Converting a Hexadecimal Digit to a Decimal Value
- 4.8 The string Type
- 4.9 Case Study: Revising the Lottery Program Using Strings

Table of Contents

4.10 Formatting Console Output

4.11 Simple File Input and Output

Chapter 5 Loops

5.1 Introduction

5.2 The while Loop

5.3 The do-while Loop

5.4 The for Loop

5.5 Which Loop to Use?

5.6 Nested Loops

5.7 Minimizing Numeric Errors

5.8 Case Studies

5.9 Keywords break and continue

5.10 Case Study: Checking Palindromes

5.11 Case Study: Displaying Prime Numbers

Chapter 6 Functions

6.1 Introduction

6.2 Defining a Function

6.3 Calling a Function

6.4 void Functions

6.5 Passing Arguments by Value

6.6 Modularizing Code

6.7 Overloading Functions

6.8 Function Prototypes

6.9 Default Arguments

6.10 Inline Functions

6.11 Local, Global, and Static Local Variables

6.12 Passing Arguments by Reference

6.13 Constant Reference Parameters

6.14 Case Study: Converting Hexadecimals to Decimals

6.15 Function Abstraction and Stepwise Refinement

Table of Contents

Chapter 7 Single-Dimensional Arrays and C-Strings

- 7.1 Introduction
- 7.2 Array Basics
- 7.3 Problem: Lotto Numbers
- 7.4 Problem: Deck of Cards
- 7.5 Passing Arrays to Functions
- 7.6 Preventing Changes of Array Arguments in Functions
- 7.7 Returning Arrays from Functions
- 7.8 Problem: Counting the Occurrences of Each Letter
- 7.9 Searching Arrays
- 7.10 Sorting Arrays
- 7.11 C-Strings

Chapter 8 Multidimensional Arrays

- 8.1 Introduction
- 8.2 Declaring Two-Dimensional Arrays
- 8.3 Processing Two-Dimensional Arrays
- 8.4 Passing Two-Dimensional Arrays to Functions
- 8.5 Problem: Grading a Multiple-Choice Test
- 8.6 Problem: Finding a Closest Pair
- 8.7 Problem: Sudoku
- 8.8 Multidimensional Arrays

Chapter 9 Objects and Classes

- 9.1 Introduction
- 9.2 Defining Classes for Objects
- 9.3 Example: Defining Classes and Creating Objects
- 9.4 Constructors
- 9.5 Constructing and Using Objects
- 9.6 Separating Class Definition from Implementation
- 9.7 Preventing Multiple Inclusions
- 9.8 Inline Functions in Classes

Table of Contents

9.9 Data Field Encapsulation

9.10 The Scope of Variables

9.11 Class Abstraction and Encapsulation

Chapter 10 Object-Oriented Thinking

10.1 Introduction

10.2 The string Class

10.3 Passing Objects to Functions

10.4 Array of Objects

10.5 Instance and Static Members

10.6 Constant Member Functions

10.7 Thinking in Objects

10.8 Object Composition

10.9 Case Study: The StackOfIntegers Class

10.10 Class Design Guidelines

Chapter 11 Pointers and Dynamic Memory Management

11.1 Introduction

11.2 Pointer Basics

11.3 Defining Synonymous Types Using the typedef Keyword

11.4 Using const with Pointers

11.5 Arrays and Pointers

11.6 Passing Pointer Arguments in a Function Call

11.7 Returning a Pointer from Functions

11.8 Useful Array Functions

11.9 Dynamic Persistent Memory Allocation

11.10 Creating and Accessing Dynamic Objects

11.11 The this Pointer

11.12 Destructors

11.13 Case Study: The Course Class

11.14 Copy Constructors

11.15 Customizing Copy Constructors

Table of Contents

Chapter 12 Templates, Vectors, and Stacks

- 12.1 Introduction
- 12.2 Templates Basics
- 12.3 Example: A Generic Sort
- 12.4 Class Templates
- 12.5 Improving the Stack Class
- 12.6 The C++ vector Class
- 12.7 Replacing Arrays Using the vector Class
- 12.8 Case Study: Evaluating Expressions

Chapter 13 File Input and Output

- 13.1 Introduction
- 13.2 Text I/O
- 13.3 Formatting Output
- 13.4 Functions: getline, get, and put
- 13.5 fstream and File Open Modes
- 13.6 Testing Stream States
- 13.7 Binary I/O
- 13.8 Random Access File
- 13.9 Updating Files

Chapter 14 Operator Overloading

- 14.1 Introduction
- 14.2 The Rational Class
- 14.3 Operator Functions
- 14.4 Overloading the Subscript Operator []
- 14.5 Overloading Augmented Assignment Operators
- 14.6 Overloading the Unary Operators
- 14.7 Overloading the ++ and - Operators
- 14.8 friend Functions and friend Classes
- 14.9 Overloading the << and >> Operators
- 14.10 Automatic Type Conversions

Table of Contents

14.11 Defining Nonmember Functions for Overloading Operators

14.12 The Rational Class with Overloaded Function Operators

14.13 Overloading the = Operators

Chapter 15 Inheritance and Polymorphism

15.1 Introduction

15.2 Base Classes and Derived Classes

15.3 Generic Programming

15.4 Constructors and Destructors

15.5 Redefining Functions

15.6 Polymorphism

15.7 Virtual Functions and Dynamic Binding

15.8 The protected Keyword

15.9 Abstract Classes and Pure Virtual Functions

15.10 Casting: static_cast versus dynamic_cast

Chapter 16 Exception Handling

16.1 Introduction

16.2 Exception-Handling Overview

16.3 Exception-Handling Advantages

16.4 Exception Classes

16.5 Custom Exception Classes

16.6 Multiple Catches

16.7 Exception Propagation

16.8 Rethrowing Exceptions

16.9 Exception Specification

16.10 When to Use Exceptions

Chapter 17 Recursion

17.1 Introduction

17.2 Example: Factorials

17.3 Case Study: Fibonacci Numbers

17.4 Problem Solving Using Recursion

Table of Contents

17.5 Recursive Helper Functions

17.6 Towers of Hanoi

17.7 Eight Queens

17.8 Recursion versus Iteration

17.9 Tail Recursion

APPENDIXES

Appendix A: C++ Keywords

Appendix B: The ASCII Character Set

Appendix C: Operator Precedence Chart

Appendix D: Number Systems

Appendix E: Bitwise Operations

INDEX

CREDIT