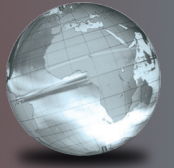


GLOBAL
EDITION



Introduction to Java™ Programming and Data Structures

Comprehensive Version

Y. Daniel Liang

TWELFTH EDITION



INTRODUCTION TO
JAVATM
PROGRAMMING AND
DATA STRUCTURES
COMPREHENSIVE VERSION

Twelfth Edition
Global Edition

Y. Daniel Liang

Georgia Southern University



Introduction to Java Programming and Data Structures, Comprehensive Version, Global Edition

Table of Contents

Cover

Title Page

Copyright

Preface

Brief Contents

Contents

VideoNotes

Animations

Chapter 1. Introduction to Computers, Programs, and Java

1.1 Introduction

1.2 What Is a Computer?

1.3 Programming Languages

1.4 Operating Systems

1.5 Java, the World Wide Web, and Beyond

1.6 The Java Language Specification, API, JDK, JRE, and IDE

1.7 A Simple Java Program

1.8 Creating, Compiling, and Executing a Java Program

1.9 Programming Style and Documentation

1.10 Programming Errors

1.11 Developing Java Programs Using NetBeans

1.12 Developing Java Programs Using Eclipse

Chapter 2. Elementary Programming

2.1 Introduction

Table of Contents

- 2.2 Writing a Simple Program
- 2.3 Reading Input from the Console
- 2.4 Identifiers
- 2.5 Variables
- 2.6 Assignment Statements and Assignment Expressions
- 2.7 Named Constants
- 2.8 Naming Conventions
- 2.9 Numeric Data Types and Operations
- 2.10 Numeric Literals
- 2.11 JShell
- 2.12 Evaluating Expressions and Operator Precedence
- 2.13 Case Study: Displaying the Current Time
- 2.14 Augmented Assignment Operators
- 2.15 Increment and Decrement Operators
- 2.16 Numeric Type Conversions
- 2.17 Software Development Process
- 2.18 Case Study: Counting Monetary Units
- 2.19 Common Errors and Pitfalls

Chapter 3. Selections

- 3.1 Introduction
- 3.2 boolean Data Type, Values, and Expressions
- 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 Generating Random Numbers
- 3.8 Case Study: Computing Body Mass Index
- 3.9 Case Study: Computing Taxes
- 3.10 Logical Operators
- 3.11 Case Study: Determining Leap Year

Table of Contents

- 3.12 Case Study: Lottery
- 3.13 switch Statements
- 3.14 Conditional Operators
- 3.15 Operator Precedence and Associativity
- 3.16 Debugging

Chapter 4. Mathematical Functions, Characters, and Strings

- 4.1 Introduction
- 4.2 Common Mathematical Functions
- 4.3 Character Data Type and Operations
- 4.4 The String Type
- 4.5 Case Studies
- 4.6 Formatting Console Output

Chapter 5. Loops

- 5.1 Introduction
- 5.2 The while Loop
- 5.3 Case Study: Guessing Numbers
- 5.4 Loop Design Strategies
- 5.5 Controlling a Loop with User Confirmation or a Sentinel Value
- 5.6 The do-while Loop
- 5.7 The for Loop
- 5.8 Which Loop to Use?
- 5.9 Nested Loops
- 5.10 Minimizing Numeric Errors
- 5.11 Case Studies
- 5.12 Keywords break and continue
- 5.13 Case Study: Checking Palindromes
- 5.14 Case Study: Displaying Prime Numbers

Chapter 6. Methods

- 6.1 Introduction
- 6.2 Defining a Method

Table of Contents

- 6.3 Calling a Method
- 6.4 void vs. Value-Returning Methods
- 6.5 Passing Arguments by Values
- 6.6 Modularizing Code
- 6.7 Case Study: Converting Hexadecimals to Decimals
- 6.8 Overloading Methods
- 6.9 The Scope of Variables
- 6.10 Case Study: Generating Random Characters
- 6.11 Method Abstraction and Stepwise Refinement

Chapter 7. Single-Dimensional Arrays

- 7.1 Introduction
- 7.2 Array Basics
- 7.3 Case Study: Analyzing Numbers
- 7.4 Case Study: Deck of Cards
- 7.5 Copying Arrays
- 7.6 Passing Arrays to Methods
- 7.7 Returning an Array from a Method
- 7.8 Case Study: Counting the Occurrences of Each Letter
- 7.9 Variable-Length Argument Lists
- 7.10 Searching Arrays
- 7.11 Sorting Arrays
- 7.12 The Arrays Class
- 7.13 Command-Line Arguments

Chapter 8. Multidimensional Arrays

- 8.1 Introduction
- 8.2 Two-Dimensional Array Basics
- 8.3 Processing Two-Dimensional Arrays
- 8.4 Passing Two-Dimensional Arrays to Methods
- 8.5 Case Study: Grading a Multiple-Choice Test
- 8.6 Case Study: Finding the Closest Pair

Table of Contents

8.7 Case Study: 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 Constructing Objects Using Constructors

9.5 Accessing Objects via Reference Variables

9.6 Using Classes from the Java Library

9.7 Static Variables, Constants, and Methods

9.8 Visibility Modifiers

9.9 Data Field Encapsulation

9.10 Passing Objects to Methods

9.11 Array of Objects

9.12 Immutable Objects and Classes

9.13 The Scope of Variables

9.14 The this Reference

Chapter 10. Object-Oriented Thinking

10.1 Introduction

10.2 Class Abstraction and Encapsulation

10.3 Thinking in Objects

10.4 Class Relationships

10.5 Case Study: Designing the Course Class

10.6 Case Study: Designing a Class for Stacks

10.7 Processing Primitive Data Type Values as Objects

10.8 Automatic Conversion between Primitive Types and Wrapper Class Types

10.9 The BigInteger and BigDecimal Classes

10.10 The String Class

10.11 The StringBuilder and StringBuffer Classes

Chapter 11. Inheritance and Polymorphism

Table of Contents

- 11.1 Introduction
- 11.2 Superclasses and Subclasses
- 11.3 Using the super Keyword
- 11.4 Overriding Methods
- 11.5 Overriding vs. Overloading
- 11.6 The Object Class and Its toString () Method
- 11.7 Polymorphism
- 11.8 Dynamic Binding
- 11.9 Casting Objects and the instanceof Operator
- 11.10 The Objects equals Method
- 11.11 The ArrayList Class
- 11.12 Useful Methods for Lists
- 11.13 Case Study: A Custom Stack Class
- 11.14 The protected Data and Methods
- 11.15 Preventing Extending and Overriding

Chapter 12. Exception Handling and Text I/O

- 12.1 Introduction
- 12.2 Exception-Handling Overview
- 12.3 Exception Types
- 12.4 Declaring, Throwing, and Catching Exceptions
- 12.5 The finally Clause
- 12.6 When to Use Exceptions
- 12.7 Rethrowing Exceptions
- 12.8 Chained Exceptions
- 12.9 Defining Custom Exception Classes
- 12.10 The File Class
- 12.11 File Input and Output
- 12.12 Reading Data from the Web
- 12.13 Case Study: Web Crawler

Chapter 13. Abstract Classes and Interfaces

Table of Contents

- 13.1 Introduction
- 13.2 Abstract Classes
- 13.3 Case Study: The Abstract Number Class
- 13.4 Case Study: Calendar and GregorianCalendar
- 13.5 Interfaces
- 13.6 The Comparable Interface
- 13.7 The Cloneable Interface
- 13.8 Interfaces vs. Abstract Classes
- 13.9 Case Study: The Rational Class
- 13.10 Class-Design Guidelines

Chapter 14. JavaFX Basics

- 14.1 Introduction
- 14.2 JavaFX vs. Swing and AWT
- 14.3 The Basic Structure of a JavaFX Program
- 14.4 Panes, Groups, UI Controls, and Shapes
- 14.5 Property Binding
- 14.6 Common Properties and Methods for Nodes
- 14.7 The Color Class
- 14.8 The Font Class
- 14.9 The Image and ImageView Classes
- 14.10 Layout Panes and Groups
- 14.11 Shapes
- 14.12 Case Study: The ClockPane Class

Chapter 15. Event-Driven Programming and Animations

- 15.1 Introduction
- 15.2 Events and Event Sources
- 15.3 Registering Handlers and Handling Events
- 15.4 Inner Classes
- 15.5 Anonymous Inner-Class Handlers
- 15.6 Simplifying Event Handling Using Lambda Expressions

Table of Contents

- 15.7 Case Study: Loan Calculator
- 15.8 Mouse Events
- 15.9 Key Events
- 15.10 Listeners for Observable Objects
- 15.11 Animation
- 15.12 Case Study: Bouncing Ball
- 15.13 Case Study: US Map

Chapter 16. JavaFX UI Controls and Multimedia

- 16.1 Introduction
- 16.2 Labeled and Label
- 16.3 Button
- 16.4 CheckBox
- 16.5 RadioButton
- 16.6 TextField
- 16.7 TextArea
- 16.8 ComboBox
- 16.9 ListView
- 16.10 ScrollBar
- 16.11 Slider
- 16.12 Case Study: Developing a Tic-Tac-Toe Game
- 16.13 Video and Audio
- 16.14 Case Study: National Flags and Anthems

Chapter 17. Binary I/O

- 17.1 Introduction
- 17.2 How Is Text I/O Handled in Java?
- 17.3 Text I/O vs. Binary I/O
- 17.4 Binary I/O Classes
- 17.5 Case Study: Copying Files
- 17.6 Object I/O
- 17.7 Random-Access Files

Table of Contents

Chapter 18. Recursion

- 18.1 Introduction
- 18.2 Case Study: Computing Factorials
- 18.3 Case Study: Computing Fibonacci Numbers
- 18.4 Problem Solving Using Recursion
- 18.5 Recursive Helper Methods
- 18.6 Case Study: Finding the Directory Size
- 18.7 Case Study: Tower of Hanoi
- 18.8 Case Study: Fractals
- 18.9 Recursion vs. Iteration
- 18.10 Tail Recursion

Chapter 19. Generics

- 19.1 Introduction
- 19.2 Motivations and Benefits
- 19.3 Defining Generic Classes and Interfaces
- 19.4 Generic Methods
- 19.5 Case Study: Sorting an Array of Objects
- 19.6 Raw Types and Backward Compatibility
- 19.7 Wildcard Generic Types
- 19.8 Erasure and Restrictions on Generics
- 19.9 Case Study: Generic Matrix Class

Chapter 20. Lists, Stacks, Queues, and Priority Queues

- 20.1 Introduction
- 20.2 Collections
- 20.3 Iterators
- 20.4 Using the for Each Method
- 20.5 Lists
- 20.6 The Comparator Interface
- 20.7 Static Methods for Lists and Collections
- 20.8 Case Study: Bouncing Balls

Table of Contents

20.9 Vector and Stack Classes

20.10 Queues and Priority Queues

20.11 Case Study: Evaluating Expressions

Chapter 21. Sets and Maps

21.1 Introduction

21.2 Sets

21.3 Comparing the Performance of Sets and Lists

21.4 Case Study: Counting Keywords

21.5 Maps

21.6 Case Study: Occurrences of Words

21.7 Singleton and Unmodifiable Collections and Maps

Chapter 22. Developing Efficient Algorithms

22.1 Introduction

22.2 Measuring Algorithm Efficiency Using Big O Notation

22.3 Examples: Determining Big O

22.4 Analyzing Algorithm Time Complexity

22.5 Finding Fibonacci Numbers Using Dynamic Programming

22.6 Finding Greatest Common Divisors Using Euclids Algorithm

22.7 Efficient Algorithms for Finding Prime Numbers

22.8 Finding the Closest Pair of Points Using Divide-and-Conquer

22.9 Solving the Eight Queens Problem Using Backtracking

22.10 Computational Geometry: Finding a Convex Hull

22.11 String Matching

Chapter 23. Sorting

23.1 Introduction

23.2 Insertion Sort

23.3 Bubble Sort

23.4 Merge Sort

23.5 Quick Sort

23.6 Heap Sort

Table of Contents

23.7 Bucket and Radix Sorts

23.8 External Sort

Chapter 24. Implementing Lists, Stacks, Queues, and Priority Queues

24.1 Introduction

24.2 Common Operations for Lists

24.3 Array Lists

24.4 Linked Lists

24.5 Stacks and Queues

24.6 Priority Queues

Chapter 25. Binary Search Trees

25.1 Introduction

25.2 Binary Search Trees Basics

25.3 Representing Binary Search Trees

25.4 Searching for an Element

25.5 Inserting an Element into a BST

25.6 Tree Traversal

25.7 The BST Class

25.8 Deleting Elements from a BST

25.9 Tree Visualization and MVC

25.10 Iterators

25.11 Case Study: Data Compression

Chapter 26. AVL Trees

26.1 Introduction

26.2 Rebalancing Trees

26.3 Designing Classes for AVL Trees

26.4 Overriding the insert Method

26.5 Implementing Rotations

26.6 Implementing the delete Method

26.7 The AVLTree Class

26.8 Testing the AVLTree Class

Table of Contents

26.9 AVL Tree Time Complexity Analysis

Chapter 27. Hashing

27.1 Introduction

27.2 What Is Hashing?

27.3 Hash Functions and Hash Codes

27.4 Handling Collisions Using Open Addressing

27.5 Handling Collisions Using Separate Chaining

27.6 Load Factor and Rehashing

27.7 Implementing a Map Using Hashing

27.8 Implementing Set Using Hashing

Chapter 28. Graphs and Applications

28.1 Introduction

28.2 Basic Graph Terminologies

28.3 Representing Graphs

28.4 Modeling Graphs

28.5 Graph Visualization

28.6 Graph Traversals

28.7 Depth-First Search (DFS)

28.8 Case Study: The Connected Circles Problem

28.9 Breadth-First Search (BFS)

28.10 Case Study: The Nine Tails Problem

Chapter 29. Weighted Graphs and Applications

29.1 Introduction

29.2 Representing Weighted Graphs

29.3 The WeightedGraph Class

29.4 Minimum Spanning Trees

29.5 Finding Shortest Paths

29.6 Case Study: The Weighted Nine Tails Problem

Chapter 30. Aggregate Operations for Collection Streams

Table of Contents

30.1 Introduction

30.2 Stream Pipelines

30.3 IntStream, LongStream, and DoubleStream

30.4 Parallel Streams

30.5 Stream Reduction Using the reduce Method

30.6 Stream Reduction Using the collect Method

30.7 Grouping Elements Using the grouping By Collector

30.8 Case Studies

Appendixes

Appendix A. Java Keywords and Reserved Words

Appendix B. The ASCII Character Set

Appendix C. Operator Precedence Chart

Appendix D. Java Modifiers

Appendix E. Special Floating-Point Values

Appendix F. Number Systems

Appendix G. Bitwise Operations

Appendix H. Regular Expressions

Appendix I. Enumerated Types

Appendix J. The Big-O, Big-Omega, and Big-Theta Notations

Java Quick Reference

Index