

PREFACE

Welcome to Introduction to Java Programming, Sixth Edition. This edition has substantially enhanced the previous edition in clarity, content, presentation, examples, and exercises, thanks to comments and suggestions by instructors and students. This edition is a great leap forward. We invite you to take a close look and be the judge.

Versions

The book is available in two versions:

- The Fundamentals First version (Chapters 1-19).
- The Comprehensive version (Chapters 1-36).

The following diagram summarizes the contents in the comprehensive version:

Introduction to Java Programming, 6E, Comprehensive Version	
Part I Fundamentals of Programming Chapter 1 Introduction to Computers, Programs, and Java Chapter 2 Primitive Data Types and Operations Chapter 3 Selection Statements Chapter 4 Loops Chapter 5 Methods Chapter 6 Arrays Part II Object-Oriented Programming Chapter 7 Objects and Classes Chapter 8 Strings and Text I/O Chapter 9 Inheritance and Polymorphism Chapter 10 Abstract Classes and Interfaces Chapter 11 Object-Oriented Design Part III GUI Programming Chapter 12 GUI Basics Chapter 13 Graphics Chapter 14 Event-Driven Programming Chapter 15 Creating User Interfaces Chapter 16 Applets and Multimedia Part IV Exception Handling, IO, and Recursion Chapter 17 Exceptions and Assertions Chapter 18 Binary I/O Chapter 19 Recursion Part V Data Structures Chapter 20 Lists, Stacks, Queues, Trees, and	Chapter 21 Generics Chapter 22 Java Collections Framework Chapter 23 Algorithm Efficiency and Sorting Part VI Concurrency, Networking, and Internationalization Chapter 24 Multithreading Chapter 25 Networking Chapter 26 Internationalization Part VII Advanced GUI Programming Chapter 27 JavaBeans and Bean Events Chapter 28 Containers, Layout Managers, and Borders Chapter 29 Menus, Toolbars, Dialogs, and Internal Frames Chapter 30 MVC and Swing Models Chapter 31 JTable and JTree Part VIII Web Programming Chapter 32 JavaDatabase Programming Chapter 33 Advanced Java Database Programming Chapter 34 Servlets Chapter 35 JavaServer Pages Chapter 36 Remote Method Invocation Appendixes

UNCORRECTED PAGE PROOFS

© 2007 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected under all copyright laws as they currently exist. No portion of this material may be reproduced, in any form or by any means, without permission in writing from the publisher.

Fundamentals First Version

The core version introduces fundamentals of programming, problem-solving, object-oriented programming, and GUI programming. This version is suitable for a course on introduction to problem solving and object-oriented programming.

Comprehensive Version

The comprehensive version contains all the chapters in the core version. Additionally, it covers data structures, networking, internationalization, advanced GUI programming and Web programming.

Custom Bundle

Instructors may order a custom bundle by adding JBuilder, NetBeans, or Eclipse supplements into any of the versions. Please contact your Prentice Hall sales representative to order custom versions and bundles.

Teaching Strategies

Fundamentals First

There are several strategies in teaching Java. This book adopts the fundamentals-first strategy, proceeding at a steady pace through all the necessary and important basic concepts, then moving to object-oriented programming, and then to the use of the object-oriented approach to build interesting GUI applications and applets with exception handling, and advanced features.

Fundamental Programming Techniques

From my own experience, confirmed by the experiences of many colleagues, we have found that learning basic logic and fundamental programming techniques like loops and step-wise refinement is essential for new programmers to succeed. Students who cannot write code in procedural programming are not able to learn object-oriented programming. A good introduction on primitive data types, control statements, methods, and arrays prepares students to learn object-oriented programming.

Using OOP Effectively

The fundamentals-first approach reinforces object-oriented programming by first presenting the procedural solutions and demonstrating how they can be improved using the object-oriented approach. Students can learn when and how to apply OOP effectively.

Object-Right, Not Object-Early

At every SIGCSE (Computer Science Education) conference prior to 2005, the object-early approach was trumpeted and the voice for the fundamentals-first approach was muted. This has been changed when some former proponents of object-early began to air their frustrations and declared that object-early failed. This book is fundamentals-first and object-right. OOP is introduced just right in time after fundamental programming techniques are covered. Many instructors of this book from research universities to community colleges have embraced the approach and have succeeded.

Problem Solving

Programming isn't just syntax, classes, or objects. It is really problem solving. Loops, methods, and arrays are fundamental techniques for problem solving. From fundamental programming techniques to object-oriented programming, there are many layers of abstraction. Classes are simply a layer of abstraction. Applying the concept of abstraction in the design and implementation of software projects is the key to developing software. The overriding objective of the book, therefore, is to teach students to use many layers of abstraction in solving problems and to see problems in small and in large. The examples and exercises throughout the book center on problem solving and foster the concept of developing reusable components and using them to create practical projects.

Learning Strategies

Practice

A programming course is quite different from other courses. In a programming course, you learn from examples, from practice, and from mistakes. You need to devote a lot of time to writing programs, testing them, and fixing errors.

Programmatic Solution

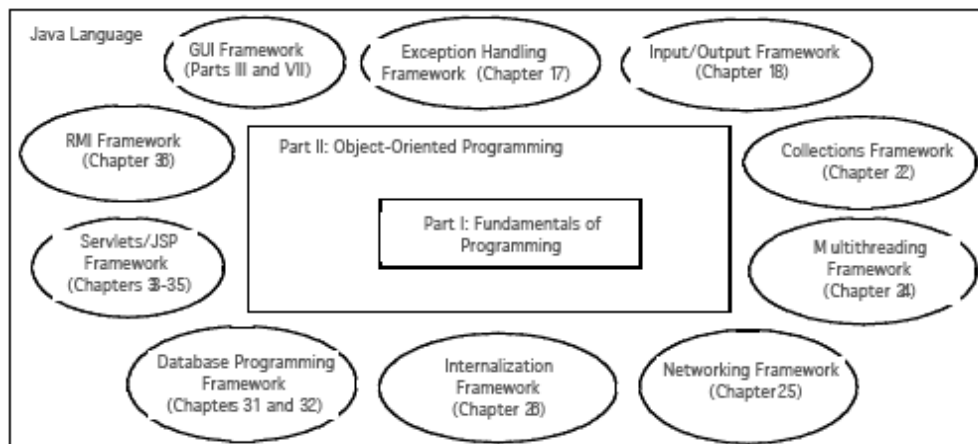
For first-time programmers, learning Java is like learning any high-level programming language. The fundamental point in learning programming is to develop the critical skills of formulating programmatic solutions for real problems and translating them into programs using selection statements, loops, and methods.

Object-Oriented Programming

Once you acquire the basic skills of writing programs using loops, methods, and arrays, you can begin to learn object-oriented programming. You will learn how to develop object-oriented software using class encapsulation and class inheritance.

Java API

Once you understand the concept of object-oriented programming, learning Java becomes a matter of learning the Java API. The Java API establishes a framework for programmers to develop applications using Java. You have to use these classes and interfaces in the API and follow their conventions and rules to create applications. The best way to learn the Java API is to imitate examples and do exercises. The following diagram highlights the API covered in the book.



What's New in This Edition?

This edition substantially enhances Introduction to Java Programming Fifth Edition. The major changes are as follows:

Full Java 5 Update

The Java 5 features are fully integrated in this edition. (Note: the fifth edition treats Java 5 features in separate optional sections.)

Complete Revision

The book is completely revised in every detail to enhance clarity, content, presentation, examples, and exercises.

GUI Options in Part II

Optional GUI sections are provided at the end of chapters 7-10 in Part II, while a complete introduction of GUI programming is presented in Part III. GUI components are excellent examples to demonstrate OOP. The optional GUI sections can be used as additional examples for OOP.

UNCORRECTED PAGE PROOFS

© 2007 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected under all copyright laws as they currently exist. No portion of this material may be reproduced, in any form or by any means, without permission in writing from the publisher.

Enhanced Examples

The book provides many new illustrations and uses small, simple, and stimulating examples to demonstrate concepts and techniques. Longer examples are presented in case studies with overall discussions and thorough line-by-line explanations.

Early Interesting Programs

Students can now write short, interesting, graphical game programs starting from Chapter 2.

JOptionPane or Scanner

Chapter 2 introduces to use `JOptionPane` and `Scanner` to receive input. Students can use either `JOptionPane` or `Scanner` for obtaining input.

New Recursion Chapter

Recursion is expanded to form a new chapter 19, “Recursion.” It can be covered after Chapter 5, “Methods.”

Early Text I/O

High-level text I/O using the `Formatter` and `Scanner` classes is introduced in Chapter 8 along with the strings. The text I/O supersedes the `Writer` and `Reader` classes and their subclasses. The binary I/O is covered in Chapter 18.

Early ArrayList

`ArrayList` is introduced early in Chapter 9.

UML Exercises

New exercises are provided in Part II for achieving three objectives: (1) design and draw UML for classes; (2) implement classes from the UML; (3) use classes to develop applications.

Extensive Supplements

Extensive supplements (e.g., installing and configuring JDK, IDE tutorials, design patterns, rapid GUI development, database design, SQL, etc.) are provided for instructors to customize a course.

Part I Enhancement

Part I, “Fundamentals of Programming,” is expanded into six chapters to focus on problem-solving and basic programming techniques with many new illustrations and practical examples such as Math learning games. New organization reinforces the teaching of fundamental problem solving techniques.

Part II Enhancement

Part II, “Object-Oriented Programming,” is expanded into five chapters to give a comprehensive introduction on OOP and how to use it to design programs. New organization enhances the presentation of object-oriented programming and enables GUI programming to be covered earlier.

Part III Enhancement

Part III, “GUI Programming,” is expanded into five chapters to introduce GUI programming, graphics painting, event-driven programming, creating user interfaces, and applets.

Part V Enhancement

Part V, “Data Structures and Collections Framework,” is expanded to cover data structure design and implementation (array list, linked list, stack, queue, heap, priority queue, and binary tree), generics, Java Collections Framework, and algorithm efficiencies and sorting.

Part VI Enhancement

Part VI, “Multithreading, Networking, and Internationalization,” is updated to cover Java 5 thread pooling and semaphores.

Part VII Enhancement

Part VII, “Advanced GUI Programming,” is expanded into five chapters with new short and simple examples to teach complex subjects.

Part VIII Enhancement

Part VIII, “Web Programming,” is expanded into five chapters. The advanced database programming is in a separate chapter and may be skipped.

Pedagogical Features

The philosophy of the Liang Java Series is teaching by example and learning by doing. Basic features are explained by example so that you can learn by doing. The book uses the following elements to get the most from the material:

Objectives list what students should have learned from the chapter. This will help them to determine whether they have met the objectives after completing the chapter.

Introduction opens the discussion with a brief overview of what to expect from the chapter.

Examples, carefully chosen and presented in an easy-to-follow style, teach programming concepts. The book uses many small, simple, and stimulating examples to demonstrate important ideas.

Chapter Summary reviews the important subjects that students should understand and remember. It helps them to reinforce the key concepts they have learned in the chapter.

Optional Sections cover nonessential but valuable features. Instructors may choose to include or skip an optional section, or cover it later. The section headers of optional sections are marked by a graphical icon.

Review Questions are grouped by sections to help students track their progress and evaluate their learning.

Programming Exercises are grouped by sections to provide students with opportunities to apply on their own the new skills they have learned. The level of difficulty is rated easy (no asterisk), moderate (*), hard (**), or challenging (***). The trick of learning programming is practice, practice, and practice. To that end, the book provides a great many exercises.

Interactive Self-Test lets students to test their knowledge interactively online. The Self-Test is accessible from the Companion Website. It provides more than one thousand multiple-choice questions organized by sections in each chapter. The Instructor Resource Website contains the quiz generator with additional multiple-choice questions.

Notes, Tips, and Cautions are inserted throughout the text to offer valuable advice and insight on important aspects of program development.

NOTE: Provides additional information on the subject and reinforces important concepts.

TIP: Teaches good programming style and practice.

CAUTION: Helps students steer away from the pitfalls of programming errors.

Flexible Chapter Orderings

The book provides flexible chapter orderings to enable GUI, Exception Handling, generics, and the Java Collections Framework to be covered earlier. Three common alternative orderings are shown as follows:

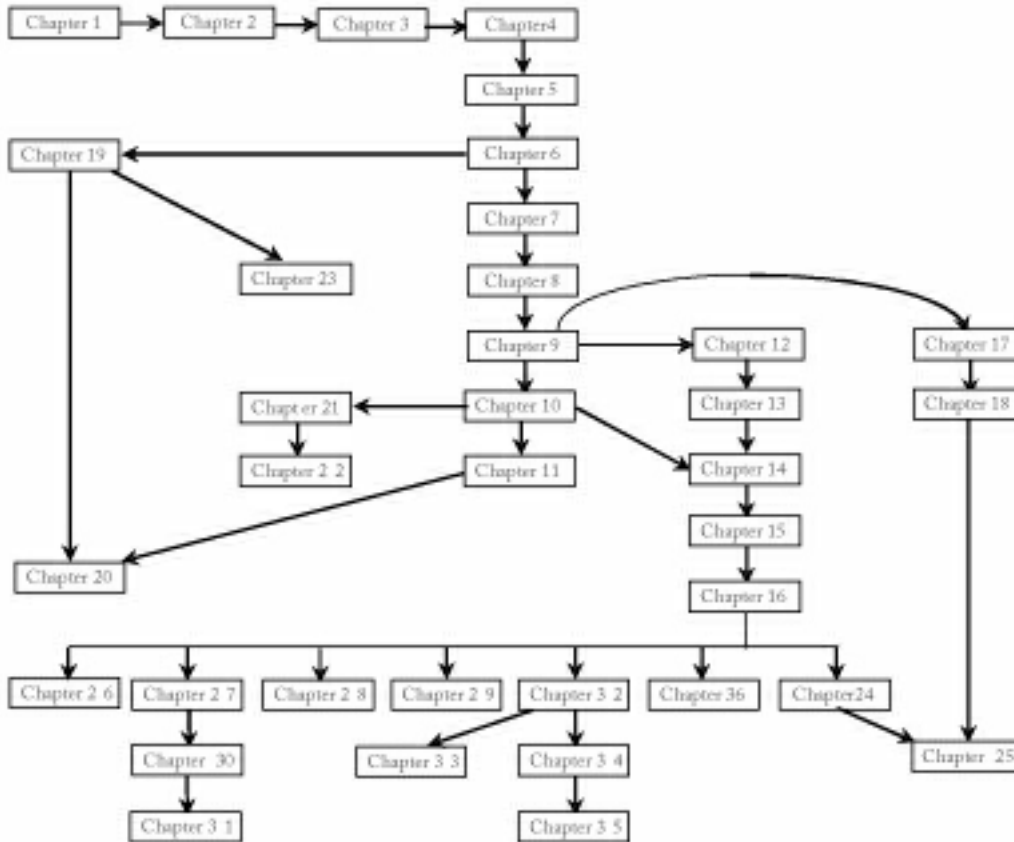
GUI Early	Exception Early	Data Structures Early
Chapter 9 Inheritance and Polymorphism	Chapter 9 Inheritance and Polymorphism	Chapter 9 Inheritance and Polymorphism
Chapter 12 GUI Basics	Chapter 17 Exceptions and Assertions	Chapter 10 Abstract Classes and Interfaces
Chapter 13 Graphics		Chapter 20 Lists, Stacks, Queues, Trees, and Heaps
Chapter 10 Abstract Classes and Interfaces		Chapter 21 Generics
Chapter 14 Event-Driven Programming		Chapter 22 Java Collections Framework
Chapter 15 Creating User Interfaces		
Chapter 16 Applets and Multimedia		

UNCORRECTED PAGE PROOFS

© 2007 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected under all copyright laws as they currently exist. No portion of this material may be reproduced, in any form or by any means, without permission in writing from the publisher.

Many of the chapters after Chapter 16 can be covered in any order. The following diagram shows the chapter dependencies.



NOTE: Some of the optional examples and exercises in later chapters may be dependent on earlier chapters. In such cases the examples and exercises can be omitted. For example, Chapter 17 has an example that uses GUI components in Chapter 15. If you have not covered Chapter 15, this type of examples and exercises can be skipped.

Java Development Tools

You can use a text editor, such as the Windows Notepad or WordPad, to create Java programs, and compile and run the programs from the command window. You can also use a Java development tool, such as TextPad, JBuilder, NetBeans, or Eclipse. These tools support an integrated development environment (IDE) for rapidly developing Java programs. Editing, compiling, building, executing, and debugging programs are integrated in one graphical user interface. Using these tools effectively will greatly increase your programming productivity. TextPad is a primitive IDE tool. JBuilder, NetBeans, and Eclipse are more sophisticated, but they are easy to use if you follow the tutorials. Tutorials on TextPad, JBuilder, NetBeans, and Eclipse are in the supplements on the Companion Website.

Companion Website

The Companion Website at www.prenhall.com/liang contains the following resources:

- Answers to review questions
- Solutions to even-numbered programming exercises
- Source code for the examples in the book
- Interactive Self-Test (organized by sections for each chapter)
- Supplements
- Resource links
- Errata

UNCORRECTED PAGE PROOFS

© 2007 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected under all copyright laws as they currently exist. No portion of this material may be reproduced, in any form or by any means, without permission in writing from the publisher.

Supplements

The text covers the essential subjects. The supplements extend the text to introduce additional topics that might be of interest to readers. The following supplements are available from the Companion Website.

Supplements for Introduction to Java Programming, 6E	
Part I General Supplements	
A Glossary	D Security
B Installing and Configuring JDK	E Java Media Framework
C Compiling and Running Java from the Command Window	F Java Sound
D Java Coding Style Guidelines	G Java Mail
E Creating Desktop Shortcuts for Java Applications on Windows	H Design Patterns
Part II IDE Supplements	Part VI Obsolete Java Features
A TextPad Tutorial	A StringTokenizer
B JBuilder Tutorial	B Text I/O using Reader and Writer
C Learning Java Effectively with JBuilder	Part VII Web Programming Supplements
D NetBeans Tutorial	A HTML and XHTML Tutorial
E Learning Java Effectively with NetBeans	B CSS Tutorial
F Eclipse Tutorial	C XML
G Learning Java Effectively with Eclipse	D Java and XML
Part III Data Structures Supplements	E Tomcat Tutorial
A Hashing	Part VIII Visual GUI Development Using NetBeans
B B-Tree	A Getting Started with NetBeans
C Graph Algorithms	B Visual Design Using NetBeans
Part IV Database Supplements	C Rapid Component Development Using NetBeans
A SQL statements for creating and initializing tables used in the book	D Customizing Property Editors in NetBeans
B MySQL Tutorial	Part IX Visual GUI Development Using JBuilder
C Oracle Tutorial	A Getting Started with JBuilder
D Microsoft Access Tutorial	B Basic UI Design Using JBuilder
E Introduction to Database Systems	C Rapid Component Development Using JBuilder
F Relational Database Concept	D Customizing Property Editors in JBuilder
G Database Design	Part X Visual GUI Development Using Eclipse
H SQL Basics	A Getting Started with Eclipse
I Advanced SQL	B Basic UI Design Using Eclipse
Part V Java Supplements	C Rapid Component Development Using Eclipse
A More on Regular Expressions	D Customizing Property Editors in Eclipse
B Enumerated Types (Java 5)	
C Java 2D	

Instructor Resource Website

The Instructor Resource Website accessible from www.prenhall.com/liang contains the following resources:

Microsoft PowerPoint slides with interactive buttons to view full-color, syntax-highlighted source code and to run programs without leaving the slides.

Sample exams. In general, each exam has four parts:

- (1) Multiple-choice questions or short-answer questions (most of these are different from the questions in the self-test on the Companion Website)
- (2) Correct programming errors
- (3) Trace programs
- (4) Write programs

UNCORRECTED PAGE PROOFS

© 2007 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected under all copyright laws as they currently exist. No portion of this material may be reproduced, in any form or by any means, without permission in writing from the publisher.

Solutions to all the exercises. Students will have access to the solutions of even-numbered exercises in the book's companion CD-ROM.

Web-based quiz generator. (Instructors can choose chapters to generate quizzes from a large database of more than 2000 questions.)

Online quiz. (Students can take the online quiz for each chapter and a quiz report will be sent to the instructor.)

Some readers have requested the materials from the Instructor Resource Website. Please understand that these are for instructors only. Such requests will not be answered.

Acknowledgments

I would like to thank Ray Greenlaw and my colleagues at Armstrong Atlantic State University for enabling me to teach what I write and for supporting me in writing what I teach. Teaching is the source of inspiration for continuing to improve the book. I am grateful to the instructors and students who have offered comments, suggestions, bug reports, and praise.

This book was greatly enhanced thanks to outstanding reviews. Many thanks to the reviewers of this and previous editions.

It is a great pleasure, honor, and privilege to work with Prentice Hall. I would like to thank Marcia Horton, Tracy Dunkelburger, Robin O'Brien, Christianna Lee, Jennifer Cappello, Barrie Reinhold, Vince O'Brien, Camille Trentacoste, Xiaohong Zhu, and their colleagues for organizing, producing, and promoting this project, and Robert Milch for copy editing.

As always, I am indebted to my wife, Samantha, for her love, support, and encouragement.

Y. Daniel Liang

liang@armstrong.edu

www.cs.armstrong.edu/liang/intro6e