JAVA FOR Students

sixth edition



DOUGLAS BELL & MIKE PARR

Java for Students

Visit the *Java for Students,* sixth edition Companion Website at www.pearsoned.co.uk/bell to find valuable **student** learning material including:

- How to download Java 6.0
- Programs from the book
- An extra chapter on Java network programming

Java for Students

Table of Contents

Front Cover
Java for Students
Contents
Detailed contents
Introduction
Guided tour
The background to Java
- The history of Java
- The main features of Java
- What is a program?
- Programming principles
- Programming pitfalls
- Summary
- Exercises
- Answers to self-test questions

First programs
- Introduction
- Integrated development environments
- Files and folders
- Creating a Java program
- The libraries
- Demystifying the program
- Objects, methods: an introduction

Pearson

Table of Contents

- Classes: an analogy
- Using a text field
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Using graphics methods

- Introduction
- Events
- The button-click event
- The graphics coordinate system
- Explanation of the program
- Methods for drawing
- Drawing with colours
- Creating a new program
- The sequence concept
- Adding meaning with comments
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Variables and calculations

Table of Contents

Introduction
The nature of int
The nature of double
Declaring variables
The assignment statement
Calculations and operators
The arithmetic operators
The % operator
Joining strings with the + operator
Converting between strings and numbers
Message dialogs and input dialogs
Formatting text in dialogs with

Converting between numbers
Constants: using final
The role of expressions
Programming principles
Programming pitfalls
Grammar spot
New language elements
Summary
Exercises
Answers to self-test questions

Methods and parameters

Introduction
Writing your own methods
A first method
Calling a method

Table of Contents

Passing parameters
Formal and actual parameters
A triangle method
Local variables
Name clashes
Event-handling methods and main
return and results
Building on methods: drawHouse
Building on methods: areaHouse
this and objects
Overloading
Programming principles
Programming pitfalls
Grammar spot
New language elements
Summary
Exercises
Answers to self-test questions

Using objects

Introduction
Instance variables
Instantiation: using constructors with new
The Random class
The main method and new
The Swing toolkit
Events
Creating a JButton
Guidelines for using objects

Table of Contents

- The JLabel class
- The JTextField class
- The JPanel class
- The Timer class
- The JSlider class
- The ImageIcon class moving an image
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Selection

- Introduction
- The if statement
- if...else
- Comparison operators
- Multiple events
- And, or, not
- Nested ifs
- switch
- Boolean variables
- Comparing strings
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements

Table of Contents

- Summary
- Exercises
- Answers to self-test questions

Repetition

- Introduction
- while
- for
- And, or, not
- do...while
- Nested loops
- Combining control structures
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Writing classes

- Introduction
- Designing a class
- Classes and files
- private variables
- public methods
- The get and set methods
- Constructors
- Multiple constructors
- private methods

Table of Contents

- Scope rules
- Operations on objects
- Object destruction
- static methods
- static variables
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Inheritance

- Introduction
- Using inheritance
- protected
- Scope rules
- Additional items
- Overriding
- Class diagrams
- Inheritance at work
- super
- Constructors
- final
- Abstract classes
- Programming principles
- Programming pitfalls
- New language elements

Table of Contents

- Summary
- Exercises
- Answers to self-test questions

Calculations

- Introduction
- Library mathematical functions and constants
- Formatting numbers
- Case study money
- Case study iteration
- Graphs
- Exceptions
- Programming principles
- Programming pitfalls
- Summary
- Exercises
- Answer to self-test question

Array lists

- Introduction
- Creating an array list and generics
- Adding items to a list
- The length of a list
- Indices
- Displaying an array list
- The enhanced for statement
- Using index values
- Removing items from an array list
- Inserting items within an array list
- Lookup

Table of Contents

- Arithmetic on an array list
- Searching
- Programming principles
- Programming pitfalls
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Arrays

- Introduction
- Creating an array
- Indices
- The length of an array
- Passing arrays as parameters
- The enhanced for statement
- Using constants with arrays
- Initializing an array
- A sample program
- Lookup
- Searching
- Arrays of objects
- Programming principles
- Programming pitfalls
- Grammar spot
- Summary
- Exercises
- Answers to self-test questions

Arrays two dimensional

Table of Contents

- Introduction
- Declaring an array
- Indices
- The size of an array
- Passing arrays as parameters
- Using constants with two-dimensional arrays
- Initializing an array
- A sample program
- Programming principles
- Programming pitfalls
- Summary
- Exercises
- Answers to self-test questions

String manipulation

- Introduction
- Using strings a recap
- The characters within strings
- A note on the char type
- The String class
- The String class methods
- Comparing strings
- Amending strings
- Examining strings
- String conversions
- String parameters
- An example of string processing
- String case study Frasier
- Programming principles

Table of Contents

- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answer to self-test question

Exceptions

- Introduction
- Exceptions and objects
- When to use exceptions
- The jargon of exceptions
- A trycatch example
- try and scopes
- The search for a catcher
- Throwing an introduction
- Exception classes
- Compilation and checked exceptions
- Catching the common cases
- Using the exception class structure
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Files and console applications

Table of Contents

- Introduction
- File access: stream or random?
- The essentials of streams
- The Java I/O classes
- The BufferedReader and PrintWriter classes
- File output
- File input
- File searching
- The File class
- The JFileChooser class
- Console I/O
- The System class
- Using JOptionPane
- A console example: Finder
- Reading from a remote site
- Command-line arguments
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Object-oriented design

- Introduction
- The design problem
- Identifying objects and methods
- Case study in design

Table of Contents

- Looking for reuse
- Composition or inheritance?
- Guidelines for class design
- Summary
- Exercises
- Answers to self-test questions

Program style

- Introduction
- Program layout
- Names
- Classes
- Comments
- Javadoc
- Constants
- Methods
- Nested ifs
- Nested loops
- Complex conditions
- Documentation
- Consistency
- Programming pitfalls
- Summary
- Exercises

Testing

- Introduction
- Program specifications
- Exhaustive testing
- Black box (functional) testing

Table of Contents

- White box (structural) testing
- Inspections and walkthroughs
- Stepping through code
- Incremental development
- Programming principles
- Summary
- Exercises
- Answers to self-test questions

Debugging

- Introduction
- Debugging without a debugger
- Using a debugger
- Common errors compilation errors
- Common errors run-time errors
- Common errors logic errors
- Common errors misunderstanding the language
- Summary
- Answer to self-test question

Threads

- Introduction
- Threads
- Starting a thread
- Thread dying
- join
- The state of a thread
- Scheduling, thread priorities and yield
- Programming principles
- Summary

Table of Contents

- Exercises
- Answers to self-test questions

Interfaces

- Introduction
- Interfaces for design
- Interfaces and interoperability
- Interfaces and the Java library
- Multiple interfaces
- Interfaces versus abstract classes
- Programming principles
- Programming pitfalls
- Grammar spot
- New language elements
- Summary
- Exercises
- Answers to self-test questions

Programming in the large packages

- Introduction
- Using classes and the import statement
- Creating packages using the package statement
- Packages, files and folders
- Scope rules
- The Java library packages
- Programming pitfalls
- New language elements
- Summary
- Exercise
- Answers to self-test questions

Table of Contents

Polymorphism
- Introduction
- Polymorphism in action
- Programming principles
- Programming pitfalls
- New language elements
- Summary
- Exercises

Java in context
- Introduction
- Simple
- Object oriented
- Platform independence (portability)
- Performance
- Security
- Open source
- The versions of Java
- Java capabilities
- Java libraries
- Java beans
- Databases JDBC
- Java and the Internet
- Java and the World Wide Web
- The opposition: Microsofts .NET platform
- JavaScript
- Conclusion
- Summary
- Exercises

Table of Contents

Appendices

- Java libraries
- The Abstract Window Toolkit
- Applets
- Glossary
- Rules for names
- Keywords
- Scope rules (visibility)
- Bibliography
- Installing and using Java

Index

Back Cover

Pearson