

# Programming in JAVA

## MCS-508

### **Unit I Java Program Design and Development**

Introduction, Designing Good Programs, Designing a Riddle Program, Java Language Elements, Editing, Compiling, and Running a Java Program, From the Java Library: System and PrintStream

### **Unit 2 Objects: Using, Creating, and Defining**

Introduction, Using String Objects, Drawing Shapes with a Graphics Object (Optional), Class Definition, CASE STUDY: Simulating a Two-Person Game, From the Java Library: java.util.Scanner

### **Unit 3 Methods: Communicating with Objects**

Introduction, Passing Information to an Object, Constructor Methods, Retrieving Information from an Object, Passing a Value and Passing a Reference, Flow of Control: Control Structures, Testing an Improved OneRowNim, From the Java Library java.lang.Object, Object-Oriented Design: Inheritance and Polymorphism, Drawing Lines and Defining Graphical Methods

### **Unit 4 Input/Output: Designing the User Interface**

Introduction, The User Interface, A Command-Line Interface, A Graphical User Interface (GUI), Case Study: The One Row Nim Game, From the Java Library: java.io.File and File Input

### **Unit 5 Java Data and Operators**

Introduction, Boolean Data and Operators, Numeric Data and Operators, From the Java Library java.lang.Math, Numeric Processing Examples, From the Java Library java.text.NumberFormat, Character Data and Operators, Example: Character Conversions, Problem Solving = Representation + Action

### **Unit 6 Control Structures**

Introduction, Flow of Control: Repetition Structures, Counting Loops , Example: Car Loan, Graphics Example: Drawing a Checkerboard, Conditional Loops, Example: Computing Averages, Example: Data Validation, Principles of Loop Design, The switch Multiway Selection Structure, OBJECT-ORIENTED DESIGN: Structured Programming

### **Unit 7 Strings and String Processing**

Introduction, String Basics, Finding Things Within a String, Example: Keyword Search, From the Java Library: java.lang.StringBuffer, Retrieving Parts of Strings, Example: Processing Names and Passwords, Processing Each Character in a String, Comparing Strings, From the Java Library: java.util.StringTokenizer, Handling Text in a Graphics Context

### **Unit 8 Inheritance and Polymorphism**

Introduction, Java's Inheritance Mechanism, Abstract Classes, Interfaces and Polymorphism, Example: A Toggle Button, Example: The Cipher Class Hierarchy, Case Study: A Two Player Game Hierarchy, Principles of Object-Oriented Design

## **Unit 9 Arrays and Array Processing**

Introduction, One-Dimensional Arrays, Simple Array Examples, Example: Counting Frequencies of Letters, Array Algorithms: Sorting, Array Algorithms: Searching, Two-Dimensional Arrays, Multidimensional Arrays, OBJECT-ORIENTED DESIGN: Polymorphic Sorting, From the Java Library: `java.util.Vector`, Case Study: An N-Player Computer Game, A GUI-Based Game,

## **Unit 10 Exceptions: When Things Go Wrong**

Introduction, Handling Exceptional Conditions, Java's Exception Hierarchy, Handling Exceptions Within a Program, Error Handling and Robust Program Design, Creating and Throwing Your Own Exceptions, From the Java Library: `JOptionPane`

## **Unit 11 Files and Streams: Input/Output Techniques**

Introduction, Streams and Files, CASE STUDY: Reading and Writing Text Files, The File Class, Example: Reading and Writing Binary Files, Object Serialization: Reading and Writing Objects, From the Java Library `javax.swing.JFileChooser`, Using File Data in Programs

## **Unit 12 Recursive Problem Solving**

Introduction, Recursive Definition, Recursive String Methods, Recursive Array Processing, Example: Drawing (Recursive) Fractals, OBJECT-ORIENTED DESIGN: Tail Recursion, OBJECT-ORIENTED DESIGN:

Recursion or Iteration? From the Java Library: `javax.swing.JComboBox`

## **Unit 13 Graphical User Interfaces**

Introduction, Java GUIs: From AWT to Swing, The Swing Component Set, OBJECT-ORIENTED DESIGN:

Model-View-Controller Architecture, The Java Event Model, CASE STUDY: Designing a Basic GUI, Containers and Layout Managers, Checkboxes, Radio Buttons, and Borders, Menus and Scroll Panes

## **Unit 14 Threads and Concurrent Programming**

Introduction, What Is a Thread?, From the Java Library: `java.lang.Thread`, Thread States and Life Cycle, Using Threads to Improve Interface Responsiveness, CASE STUDY: Cooperating Threads, CASE STUDY: The Game of Pong

## **Unit 15 Sockets and Networking**

Introduction, An Overview of Networks, Using Multimedia Network Resources for a Graphical Program, From the Java Library: `java.net.URL`, The Slide Show Program, Adding Text Network Resources for an Application, Client/Server Communication via Sockets, CASE STUDY: Generic Client/Server Classes, Playing One Row Nim Over the Network, Java Network Security Restrictions, Java Servlets and Java Server Pages

## **Unit 16 Data Structures: Lists, Stacks, and Queues**

Introduction, The Linked List Data Structure, OBJECT-ORIENTED DESIGN: The List Abstract Data Type (ADT), The Stack ADT, The Queue ADT, From the Java Library: The Java

Collections Framework and Generic Types, Using the Set and Map Interfaces, The Binary Search Tree Data Structure