Java Programming – Course Objectives

Overview

This course of study builds on the skills gained by students in Java Fundamentals or Java Foundations to help advance Java programming skills. Students will design object-oriented applications with Java and will create Java programs using hands-on, engaging activities.

Available Curriculum Languages:

- English

Duration

- Recommended total course time: 90 hours*
- Professional education credit hours for educators who complete Oracle Academy training: 30

** Course time includes instruction, self-study/homework, practices, projects and assessment*

Target Audiences

Educators

- Technical, vocational, and 2- and 4-year college and university faculty members who teach computer programming or a related subject
- Secondary and vocational school teachers who teach computer programming

Students

- Students who wish to extend their programming experience in Java and develop more complex Java applications
- This course is a suitable foundational class for computer science majors and non-majors alike, and when taught in sequence with Java Fundamentals or Java Foundations, and may be used to prepare students for the AP Computer Science A exam

Prerequisites

Required:

- Fundamental knowledge of object-oriented concepts, terminology, and syntax, and the steps required to create basic Java programs

Suggested:

- Either:
  - Oracle Academy Course - Java Fundamentals
  - Oracle Academy Course – Java Foundations
- Previous experience with at least one programming language

Suggested Next Courses

- Advanced computer programming courses

Lesson-by-Lesson Topics and Objectives

Section 1 – Java Language – What I Should Know

- 1-1 Fundamentals of Java – What I Should Know
  - Review of Java Primitives
  - Review of Strings
  - Review of Logical and Relational Operators
  - Review of Conditional Statements
  - Review of Program Control
  - Review of Object Classes
  - Review of Constructor and Method Overloading
  - Review of Inheritance

Section 2 – Class Design and Exceptions

- 2-1 Working with Pre-Written Code
  - Read and understand a pre-written Java program consisting of classes and interacting objects
  - Apply the concept of inheritance in the solutions of problems
  - Test classes in isolation
  - Describe when it is more appropriate to use an ArrayList than an Array

- 2-2 Java Class Design – Interfaces
  - Model business problems using Java classes
  - Make classes immutable
  - Use Interfaces

- 2-3 Java Class Design – Abstract Classes
  - Use Abstract Classes
  - Use the instanceof operator to compare object types
  - Use virtual method invocation
  - Use upward and downward casts

- 2-4 Exceptions and Assertions
  - Use exception handling syntax to create reliable applications
  - Use try and throw statements
  - Use the catch, multi-catch, and finally statements
  - Recognize common exception classes and categories
  - Create custom exception and auto-closeable resources
  - Test invariants by using assertions

Section 3 – Data Structures: Generics and Collections

- 3-1 Generics
  - Create a custom generic class
  - Use the type interface diamond to create an object
  - Use generic methods
  - Use wildcards
  - Use enumerated types

- 3-2 Collections – Part 1
  - Create a collection without using generics
  - Create a collection using generics
  - Implement an ArrayList
  - Implement a Set

- 3-3 Collections – Part 2
  - Implement a HashMap
  - Implement a stack by using a deque
  - Define a link list
  - Define a queue
  - Implement a comparable interface

- 3-4 Sorting and Searching
    - Recognize the sort order of primitive types and objects
    - Trace and write code to perform a simple Bubble Sort of integers
    - Trace and write code to perform a Selection Sort of integers
    - Trace and write code to perform a Binary Search of integers
    - Compare and contrast search and sort algorithms
    - Analyze the Big-O for various sort algorithms

Section 4 – Strings, Regular Expressions, and Recursion

- 4-1 String Processing
    - Read, search, and parse Strings
    - Use StringBuilder to create Strings

- 4-2 Use Regular Expressions
    - Use regular expressions
    - Use regular expressions to:
        - Search Strings
        - Parse Strings
        - Replace Strings

- 4-3 Recursion
    - Create linear recursive methods
    - Create non-linear recursive methods
    - Compare the pros and cons of recursion

Section 5 – Input and Output

- 5-1 Basics of Input and Output
    - Describe the basics of input and output in Java
    - Read data from and write data to the console

- 5-2 Input and Output Fundamentals
    - Use streams to read and write files
    - Read and write objects by using serialization

- 5-3 Deploying an Application
    - Describe the concept of packages
    - Describe how to deploy an application
    - Describe a complete Java application that includes a database back end

Section 6 – JDBC

- 6-1 JDBC Introduction
    - Describe the JDBC
    - Introduce the Oracle JDBC driver
    - Outline the steps in JDBC programming
    - Describe the JDBC statement

- 6-2 JDBC Basics
    - JDBC Data Types
    - Programming with JDBC PreparedStatement
    - Programming with Use JDBC CallableStatement
    - Reading MetaData from Database

Section 7 – Java Memory and the JVM

- 7-1 Introduction to JVM Architecture
    - What is Java Technology?
    - Primary goals of Java Technology
    - The Java Virtual Machine architecture
    - JVM runtime area

- 7-2 Java Memory Structure
    - Introduce Java Heap Memory
    - Garbage collection
    - Analyze the memory allocation in JVM

Section 8 – class File and the JDK

- 8-1 JDK Tools
    - Introduce the javac command
    - Introduce the java command
    - How to use the jps command
    - How to use the jstat command
    - Introduce and use the javap command
    - How to use the jdb command
    - Introduce the jvisualvm tool
    - Introduce the hsdis plugin

- 8-2 class File
    - Understand the class file structure
    - Identify the access field
    - Identify the method structure and bytecode
        - Method Info: Code_attribute
        - Code Attribute: LineNumberTable_attribute
        - Class Attribute: SourceFile_attribute

Section 9 – Bytecode and ClassLoader

- 9-1 Java Bytecode
    - Understanding Bytecode
        - How to obtain bytecode listings
        - How to read bytecode
        - How the language constructs are mirrored by the compiler: calculation, method calls

- 9-2 ClassLoader
    - The Class Loading Overview
    - ClassLoader loading procedure
    - JDK ClassLoader Class
    - ClassLoader Hierarchy
    - Custom ClassLoader
    - Class Linking