Java Programming – Course Objectives

Overview

This course of study builds on the skills gained by students in Java Fundamentals and helps to 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 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:

- Oracle Academy Course - Java Fundamentals
- Previous experience with at least one programming language

Suggested Next Courses

- Advanced computer programming courses

Lesson-by-Lesson Topics and Objectives

Section1 - Introduction

- 1-1 Fundamentals of Java – What I should know
    - Review Java Primitives
    - Review Strings
    - Review Logical and Relational Operators
    - Review Conditional Statements
    - Review Program Control
    - Review Object Classes
    - Review Constructor and Method Overloading
    - Review Inheritance
- 1-2 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

Section2 - Classes and Collections

- 2-1 Java class Design – Interfaces
    - Model business problems using Java classes
    - Make classes immutable
    - Use Interfaces
- 2-2 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
- 2-3 Generics
    - Create a custom generic class
    - Use the type interface diamond to create an object
    - Use generic methods
    - Use wildcards
    - Use enumerated types
- 2-4 Collections – Part I
    - Create a collection without using generics
    - Create a collection using generics
    - Implement an ArrayList
    - Implement a Set
- 2-5 Collections – Part II
    - Implement a HashMap
    - Implement a stack by using a deque
    - Define a link list
    - Define a queue
    - Implement a comparable interface
- 2-6 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 3 - Strings and Recursion

- 3-1 String Processing
    - Read, search, and parse Strings
    - Use StringBuilder to create Strings
- 3-2 Use regular expressions
    - Use regular expressions
    - Use regular expressions to:
        - Search Strings
        - Parse Strings
        - Replace Strings

- 3-3 Recursion
    - Create linear recursive methods
    - Create non-linear recursive methods
    - Compare the advantages and disadvantages of recursion
- 3-4 Basics of Input and Output
    - Describe the basics of input and output in Java
    - Read data from and write data to the console
- 3-5 Input and Output Fundamentals
    - Read data from and write data to the console
    - Use streams to read and write files
    - Read and write objects by using serialization
- 3-6 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 4 - Deploying an Application

- 4-1 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

To search and register for events scheduled in your area, visit the Academy events calendar.