Java Programming

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.

Duration

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

* Total 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

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

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

Java Class Design
- Model business problems using Java classes
- Use the instance of operator to compare object types
- Make classes immutable
Use virtual method invocation
• Compare default and public access levels
• Use upward and downward casts

Generics and Collections
• Create a custom generic class
• Implement an ArrayList
• Use the type inference diamond to create an object
• Implement a Set
• Create a collection without using generics
• Implement a HashMap
• Use enumerated types
• Implement a stack by using a deque

String Processing
• Read, search and parse strings
• Use StringBuilder to create strings
• Use regular expressions to search, parse, and replace strings

Exceptions and Assertions
• Use exception handling syntax to create reliable applications
• Recognize common exception classes and categories
• Create custom exception and auto-closeable resources
• Test invariants by using assertions

Input / Output Fundamentals
• Describe the basics of input and output in Java
• Read data from and write data to the console
• Use streams to read and write files
• Read and write objects by using serialization

Final Project and Presentation
• Develop a final project
• Present the final project

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