academy.oracle.com

Java Foundations – Course Objectives

Overview

This course of study engages students with little programming experience. Students are introduced to object-oriented concepts, terminology, and syntax, and the steps required to create basic Java programs using hands-on, engaging activities. Students will learn the concepts of Java programming, design object-oriented applications with Java and create Java programs using hands-on, engaging activities.

Available Curriculum Languages:

- Arabic, Simplified Chinese, English, French, Indonesian, Japanese, Brazilian Portuguese, Russian, Spanish

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, information communications technology (ICT), or a related subject at a foundational level.
- Secondary and vocational school teachers who teach computer programming.

Students

- Students who wish learn Java programming and build their Object Oriented Programming experience using Java.
- This course is a suitable foundational class for computer science majors, and when taught in sequence with Java Programming may be used to prepare students for the AP Computer Science A exam.

Prerequisites

Required

- Oracle Academy Workshop - Getting Started with Java Using Alice
- Oracle Academy Workshop - Creating Java Programs with Greenfoot

Suggested

- Oracle Academy Course - Java Fundamentals

Suggested Next Courses

- Oracle Academy Course - Java Programming

Lesson-by-Lesson Topics and Objectives

Section 1 – Introduction

- 1-1 About the Course
    - Identify course goals and objectives
    - Understand the course environment
    - Describe the course learning strategy
- 1-2 A Brief History
    - Show examples of how people interact with Java in their daily lives
    - Summarize the history of Java
    - Understand Java technology product groups
- 1-3 Setting up Java
    - Understand the difference between the JDK and JRE
    - Understand the difference between .java and .class files
    - Describe the purpose of an integrated development environment (IDE)
    - Download and install the JDK, JRE, and NetBeans IDE
    - Import a project into NetBeans

Section 2 – Java Software Development

- 2-1 The Software Development Process
    - Understand the Spiral Model of development
    - Recognize tasks and subtasks of the Spiral Model
    - Recognize what happens when steps are ignored
    - Identify software features
    - Understand how features are gradually implemented
- 2-2 What is my Program Doing?
    - Understand how Java is read line by line
    - Set and use breakpoints
    - End statements with semicolons (;)
    - Organize code using whitespace and other conventions
    - Create comments
- 2-3 Introduction to Object-Oriented Programming Concepts
    - Differentiate between procedural and object-oriented programming
    - Understand a class as a blueprint for an object
    - Understand a class is used to create instances of an object
    - Model objects as a combination of
        - Properties (data fields)
        - Behaviors (methods)

Section 3 – Java Data Types

- 3-1 What is a Variable?
    - Understand the benefits of variables.
    - Identify four main types of variables:
        - (boolean, int, double, String)
    - Declare and assign values to variables
    - Name variables according to conventions
- 3-2 Numeric Data
    - Differentiate integer data types (byte, short, int, long)
    - Differentiate floating point data types (float, double)
    - Manipulate and do math with numeric data
    - Use parentheses and order of operations
- 3-3 Textual Data
    - Use the char data type
    - Use Strings
    - Concatenate Strings
    - Understand escape sequences
    - Understand print statements better

- 3-4 Converting Between Data Types
    - Take advantage of automatic promotion
        - And when to be cautious with promotions
    - Cast variables to other data types
        - And when to be cautious with casting
    - Parse Strings as numeric values
- 3-5 Keyboard Input
    - Understand user input
    - Create a JOptionPane to collect user input
    - Use a Scanner to collect input from the console
    - Use a Scanner to collect input from a file
    - Understand how a Scanner handles tokens and delimiters

Section 4 – Java Methods and Library Classes

- 4-1 What is a Method?
    - Structure code within a class
    - Instantiate an object
    - Understand the benefits of methods
    - Use the dot operator (.) to access an object's fields and methods
    - Supply arguments to a method
    - Return values from a method
- 4-2 The import Declaration and Packages
    - This lesson covers the following objectives:
    - Access a class by using its fully qualified name
    - Describe the function of the import statement
    - Use the import statement to access a class in a package
    - Understand the purpose of an asterisk in an import statement
    - Identify packages that are automatically imported
- 4-3 The String Class
    - Locate the String class in the Java API documentation
    - Understand the methods of the String class
    - Compare two String objects lexicographically
    - Find the location of a substring in a String object
    - Extract a substring from a String object
- 4-4 The Random Class
    - Describe the purpose and uses of random numbers in Java programming
    - Identify methods of the Random class that obtain random numbers
    - Obtain random numbers in a range of numbers
    - Understand the purpose of the random number seed
- 4-5 The Math Class
    - Understand the methods of the Math class
    - Use methods of the Math class to perform mathematical calculations
    - Use fields of the Math Class

Section 5 – Decision Statements

- 5-1 Boolean Expressions and if/else Constructs
    - Declare, initialize, and use boolean variables
    - Compare boolean expressions using relational operators
    - Create an if statement
    - Create if/else constructs
    - Compare Strings
- 5-2 Understanding Conditional Execution
    - Describe conditional execution
    - Describe logical operators
    - Understand "short circuit" evaluation of logical operators
    - Build chained if constructs

- 5-3 switch Statement
  - Create a switch control structure
  - Compare if/else constructs with switch control structures
  - Understand the purpose of the break keyword

Section 6 – Loop Constructs

- 6-1 for Loops
  - Understand the components of the standard for loop
  - Understand how to create and use a for loop
  - Understand variable scope
  - Understand debugging techniques
  - Explain how infinite loops occur in Java
- 6-2 while and do-while Loops
  - Use a while loop in a Java program (pre-test)
  - Use a do-while loop in a Java program (post-test)
  - Understand when one loop type may be more beneficial than another
- 6-3 Using break and continue Statements
  - Use a break statement to exit a loop
  - Use a continue statement to skip part of a loop
  - Explain the need for loop comments

Section 7 – Creating Classes

- 7-1 Creating a Class?
  - Create a Java test/main class
  - Create a Java class in NetBeans
  - Use conditionals in methods
  - Translate specifications or a description into fields and behaviors
- 7-2 Instantiating Objects
  - Understand the memory consequences of instantiating objects
  - Understand object references
  - Understand the difference between stack and heap memory
  - Understand how Strings are special objects
- 7-3 Constructors
  - Understand default values
  - Crash the program with a null reference
  - Understand the default constructor
  - Write a constructor that accepts arguments
  - Initialize fields with a constructor
  - Use this as an object reference
- 7-4 Overloading Methods
  - Understand the effects of multiple constructors in a class
  - Define overloading of a method
  - Explain the method signature
  - Understand when overloading is and isn’t possible
- 7-5 Object Interaction and Encapsulation
  - Understand object interaction in greater detail
  - Use the private modifier to define class variables
  - Understand the purpose of getter methods
  - Understand the purpose of setter methods
- 7-6 static Variables and Methods
  - Describe a static variable and demonstrate its use within a program
  - Describe a static method and demonstrate its use within a program
  - Understand how to use the final keyword with static variables

Section 8 – Arrays and Exceptions

- 8-1 One-dimensional Arrays
  - Create and initialize one-dimensional arrays
  - Modify an array element
  - Traverse a one-dimensional array by using a for loop
  - Identify the cause of an ArrayIndexOutOfBoundsException
- 8-2 ArrayLists
  - Create an ArrayList
  - Manipulate an ArrayList by using its methods
  - Traverse an ArrayList using iterators and for-each loops
  - Use wrapper classes and Autoboxing to add primitive data types to an ArrayList
- 8-3 Exception Handling
  - Explain the purpose of exception handling
  - Handle exceptions with a try/catch construct
  - Describe common exceptions thrown in Java
- 8-4 Debugging Concepts and Techniques
  - Test and debug a Java program
  - Identify the three types of errors
  - Apply debugging techniques
    - print statements
    - NetBeans debugger
  - Apply some debugging tips and techniques

Section 9 – JavaFX

- 9-1 Introduction to Java FX
  - Create a JavaFX project
  - Explain the components of the default JavaFX project
  - Describe different types of Nodes and Panes
  - Explain the Scene Graph, Root Node, Scenes, and Stages
- 9-2 Colors and Shapes
  - Create and use custom colors
  - Create shapes and explain their properties and behaviors
  - Reference the JavaFX Ensemble
- 9-3 Graphics, Audio and MouseEvents
  - Create and use a JavaFX image and ImageView
  - Create and use JavaFX audio
  - Create and use MouseEvents
  - Understand Lambda expressions in GUI applications

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