ORACLE
Academy

academy.oracle.com

Java Fundamentals – Course Objectives

Overview

This course engages students with little or no programming experience to create Java programs. Participants are introduced to object-oriented programming concepts, terminology, and syntax, and the steps required to create basic Java programs using the Alice, Greenfoot, and Eclipse interactive development environments. Hand-on practices figure prominently throughout this course so students can experience firsthand the power of computer programming.

Available Curriculum Languages:

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

- College/university faculty who teach computer programming, information communications technology (ICT), or a related subject
- Secondary school teachers who teach computer programming

Students

- Students with little programming experience who wish to learn Java programming and build their Object Oriented Programming experience using fun Java development environments
- This course is a suitable foundational class for computer science majors

Prerequisites

Required

- Basic understanding of at least one programming language
- The ability to follow software installation instructions and install Alice, Greenfoot, and Eclipse on a computer

Suggested

- Getting Started with Java Using Alice and Creating Java Programs with Greenfoot or previous experience with at least one programming language

Suggested Next Courses

- Java Programming

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Lesson-by-Lesson Topics and Objectives

Section 1 - Introduction

- 1-1 Introduction
  - Examine the course sections
  - State the goal of the course
  - Become familiar with Oracle Academy Member Hub
  - Explain the course map
  - Describe the software used in this course
  - Recognize the IDEs used in this course

Section 2 - Using Alice 3

- 2-1 Getting Started with Alice 3
  - Identify scene components
  - Create and save a new project
  - Add an object to a scene
  - Communicate the value of saving multiple versions of a scene
  - Code a simple programming instruction
  - Use the copy and undo command
  - Understand the value of testing and debugging
- 2-2 Add and Position Objects
  - Open a saved version of a project
  - Add multiple objects to a scene
  - Describe the difference between precise positioning and drag-and-drop (or imprecise) positioning
  - Use a one-shot procedure to precisely position an object in a scene
  - Edit properties of an object in the Scene editor
  - Describe three-dimensional positioning axes
  - Position the sub-parts of an object in the Scene editor
- 2-3 Procedures and Arguments
  - Toggle between, and describe the visual differences between, the Scene editor and the Code editor
  - Locate and describe the purpose of the methods panel and the procedures tab
  - Use procedures to move objects
  - Add Java programming procedures to the Code editor
  - Demonstrate how procedure values can be altered
  - Create programming comments
  - Reorder, edit, delete, copy, and disable programming statements
  - Test and debug an animation
- 2-4 Rotation and Randomization
  - Correlate storyboard statements with program execution tasks
  - Add a control statement to the Code editor
  - Use random numbers to randomize motion
- 2-5 Declare Procedures
  - Compare and define an animation and a scenario
  - Write a storyboard
  - Flowchart a storyboard
  - Describe inheritance and how traits are passed from superclasses to subclasses
  - Describe when to implement procedural abstraction
  - Demonstrate how to declare a procedure
  - Identify and use procedural abstraction techniques to simplify animation development
- 2-6 Control Statements
  - Define multiple control statements to control animation timing
  - Create an animation that uses a control statement to control animation timing
  - Recognize programming constructs to invoke simultaneous movement
- 2-7 Functions
  - Use functions to control movement based on a return value
- 2-8 IF and WHILE Control Structures
  - Use the IF control structure to effect execution of instructions
  - Use the WHILE control structure to create a conditional loop for repetitive behavior
- 2-9 Expressions
  - Create an expression to perform a math operation
  - Interpret a math expression

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

- 2-10 Variables
    - Understand variables
    - Understand how variables are used in programming
    - Viewing Alice code as Java Code on the side
- 2-11 Keyboard Controls
    - Create an opening sequence
    - Use keyboard controls to manipulate an animation
    - Save your Class file
    - Using the starter tab
    - Add an existing class file to an animation
- 2-12 Develop a Complete Animation
    - Use functional decomposition to write a scenario and storyboard
    - Complete an animation
    - Test an animation
    - Reposition objects at runtime
    - Upload your animation
    - Plan the presentation of a completed animation project
- 2-13 Java Variables and Data Types
    - Describe variables
    - Describe Java simple types
    - Define arithmetic operators
    - Describe relational and logical operators
    - Describe assignment operators
- 2-14 Java Methods and Classes
    - Describe a method, class, and instance
    - Describe a scenario where an IF control structure would be used
    - Describe a scenario where a WHILE control structure would be used
    - Recognize the syntax for a method, class, function, and procedure
    - Describe input and output

Section 3 - Using Greenfoot

- 3-1 Getting Started With Greenfoot
    - Download and install Greenfoot
    - Describe the components of the Greenfoot interactive development environment
    - Create an instance of a class
    - Describe classes and subclasses
    - Recognize Java syntax used to correctly create a subclass
- 3-2 Methods, Variables and Parameters
    - Define parameters and how they are used in methods
    - Understand inheritance
    - Describe properties of an object
    - Examine the purpose of a variable
    - Discuss programming concepts and define terminology
- 3-3 Source Code and Documentation
    - Demonstrate source code changes to invoke methods programmatically
    - Demonstrate source code changes to write an if decision statement
    - Describe a method to display object documentation
- 3-4 Developing and Testing an Application
    - Demonstrate program testing strategies
    - Recognize phases for developing a software application
- 3-5 Randomization and Understanding Dot Notation and Constructors
    - Create randomized behaviors
    - Define comparison operators
    - Create if-else control statements
    - Create an instance of a class
    - Recognize and describe dot notation
- 3-6 Defined Methods
    - Describe effective placement of methods in a super or subclass
    - Simplify programming by creating and calling defined methods
    - Handling collisions

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

- 3-7 Sound and Keyboard Control
    - Write programming statements to include sound in a program
    - Write programming statements to include keyboard movements in a program
    - Write programming statements to include mouse interaction in a program
    - Write programming statements to retrieve information from the user
- 3-8 World Animation and Game End
    - Construct a world object using a constructor method
    - Create an object using a constructor
    - Write programming statements to use the new keyword
    - Define the purpose and syntax of a variable
    - Recognize the syntax to define and test variables
    - Write programming statements to switch between two images
    - Write programming statements to end a game
- 3-9 Abstraction
    - Define abstraction and provide an example of when it is used
    - Define casting
- 3-10 Loops, Variables, and Arrays
    - Create a while loop in a constructor to build a world
    - Describe an infinite loop and how to prevent one from occurring
    - Use an array to store multiple variables used to create a world
    - Create an expression using logic operators
    - Describe the scope of a local variable in a method

Section 4 – Java Basics

- 4-1 Getting Started with Eclipse
    - Identify components of Eclipse
    - Identify components of a Java application
    - Compile an application
    - Test to ensure application is complete
    - Write the code for GalToLit.java
    - Modify a program to execute error free
    - Modify a program to use a formula to convert units of measure
- 4-2 Object and Driver Classes
    - Describe the general form of a Java program
    - Describe the difference between an Object class and a Driver class
    - Access a minimum of two Java class APIs
    - Explain and give examples of Java keywords
    - Create an Object class
    - Create a Driver class
- 4-3 Data Types and Operators
    - Use primitive data types in Java code
    - Specify literals for the primitive types and for Strings
    - Demonstrate how to initialize variables
    - Describe the scope rules of a method
    - Recognize when an expression requires a type conversion
    - Apply casting in Java code
    - Use arithmetic operators
    - Use the assignment operator
    - Use a method from the Math class
    - Access a Math class method from the Java API
- 4-4 Strings
    - Instantiate (create) a String
    - Describe what happens when a String is modified
    - Use the + and += operators for concatenating Strings
    - Interpret escape sequences in String literals
    - Recognize the difference between a String and a primitive char data type
    - Test Strings with the compareTo() and equals() method
    - Describe why the == operator does not always work when testing String equality
    - Use String methods length(), substring(), indexOf(), and charAt()

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Section 5 - Program Structure

- 5-1 Scanner and Conditional Statements
  - Use Scanner for user input during program execution
  - Use if-else logic and statements
  - Apply switch logic and statements in Java code
  - Use break and default effectively in a switch statement
  - Use the ternary operator
- 5-2 Control Statements
  - Create a while loop
  - Create a do-while loop
  - Create a for loop

Section 6 - Arrays and Exceptions

- 6-1 Arrays
  - Write a single-dimensional array in a Java program using primitive data types
  - Write a single-dimensional array in a Java program using reference (Object) types
  - Write a 2-dimensional array in a Java program using primitive data types
  - Write a 2-dimensional array in a Java program using reference (Object) types
  - Declare an array, initialize an array, and traverse the array
  - Describe array initialization
  - Distinguish between the String method length() and an array's length value
  - Rewrite a Java program to store integers into an array, perform a mathematical calculation, and display the result
  - Use alternative array declaration syntax
- 6-2 Handling Errors
  - Describe the different kinds of errors that can occur and how they are handled in Java
  - Describe what exceptions are used for in Java
  - Determine what exceptions are thrown for any foundation class
  - Write code to handle an exception thrown by the method of a foundation class

Section 7 – Java Classes

- 7-1 Classes, Objects, and Methods
  - Recognize the correct general form of a class
  - Create an object of a class
  - Create methods that compile with no errors
  - Return a value from a method
  - Use parameters in a method
  - Create a driver class and add instances of Object classes
  - Add a constructor to a class
  - Apply the new operator
  - Describe garbage collection and finalizers
  - Apply the this reference
  - Add a constructor to initialize a value
- 7-2 Parameters and Overloading Methods
  - Use access modifiers
  - Pass objects to methods
  - Return objects from methods
  - Use variable argument methods
  - Overload constructors
  - Overload methods
  - Write a class with specified arrays, constructors, and methods
- 7-3 The Static Modifier and Nested Classes
  - Create static variables
  - Use static variables
  - Create static methods
  - Use static methods
  - Create static classes
  - Use static classes

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

- 7-4 Inheritance
    - Demonstrate and explain UML (Unified Modeling Language) class diagrams
    - Use the extends keyword to inherit a class
    - Compare and contrast superclasses and subclasses
    - Describe how inheritance affects member access
    - Use super to call a superclass constructor
    - Use super to access superclass members
    - Create a multilevel class hierarchy
    - Recognize when constructors are called in a class hierarchy
    - Demonstrate understanding of inheritance through the use of applets
    - Recognize correct parameter changes in an existing applet
- 7-5 Polymorphism
    - Apply superclass references to subclass objects
    - Write code to override methods
    - Use dynamic method dispatch to support polymorphism
    - Create abstract methods and classes
    - Recognize a correct method override
    - Use the final modifier
    - Explain the purpose and importance of the Object class
    - Write code for an applet that displays two triangles of different colors
    - Describe object references

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

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.