Java Programming Language, Java SE 6

Duration: 5 Days

What you will learn

The Java Programming Language course give you a solid foundation for programming with Java. This course is ideal for programmers interested in adding the Java programming language to their list of skills, as well as those preparing for the Oracle Certified Professional, Java SE 6 Programmer examination.

Learn To:

Understand the syntax of the Java programming language.
Use object-oriented programming with the Java programming language.
Create graphical user interfaces (GUIs), exceptions, file input/output (I/O), threads and networking.
Develop Java technology applications.

Benefits to You
Boost the productivity, communication and collaboration of your organization. At the same time, reduce the cost of application ownership through more efficient development and deployment techniques. Maintain your edge by staying current with the global standard for developing networked applications.

Java SE 6
This course features the Java Platform, Standard Edition 6 (Java SE 6) platform, and utilizes the Java SE Development Kit 6 (JDK 6) product. The students perform the course lab exercises using the NetBeans Integrated Development Environment (IDE).

Training Requirements
This course counts towards the Hands-on course requirement for the Java SE 6 Developer Certification. Only instructor-led inclass or instructor-led online formats of this course will meet the Certification Hands-on Requirement. Self Study and Knowledge Center courses do not meet the Hands-on Requirement.

Related Training

Required Prerequisites

Create and edit text files using a text editor

Understand object-oriented principles

Be competent in creating programs in any programming language or have completed the SL-110-SE6

Fundamentals of the Java Programming Language course.

Suggested Prerequisites
Fundamentals of the Java Programming Language, Java SE 6

Fundamentos da Linguagem de Programação Java

Course Objectives

Create Java technology applications that leverage the object-oriented features of the Java language, such as encapsulation, inheritance, and polymorphism

Execute a Java technology application from the command line

Use Java technology data types and expressions

Use Java technology flow control constructs

Use arrays and other data collections

Implement error-handling techniques using exception handling

Create an event-driven graphical user interface (GUI) using Swing components: panels, buttons, labels, text fields, and text areas

Implement input/output (I/O) functionality to read from and write to data and text files and understand advanced I/O streams

Create a simple Transmission Control Protocol/Internet Protocol (TCP/IP) networked client that communicates with a server through sockets

Create multithreaded programs

Course Topics

Getting Started
Examine Java technology
Analyze a simple Java technology application
Execute a Java technology application

Object-Oriented Programming
Define modeling concepts: abstraction, encapsulation, and packages
Discuss Java technology application code reuse
Define class, member, attribute, method, constructor, and package
Use the access modifiers private and public as appropriate for the guidelines of encapsulation
Invoke a method on a particular object
Use the Java technology API online documentation

Identifiers, Keywords, and Types
Use comments in a source program

Distinguish between valid and invalid identifiers
Use the eight primitive types
Define literal values for numeric and textual types
Define the terms primitive variable and reference variable
Declare variables of class type
Construct an object using new and describe default initialization
Describe the significance of a reference variable

Expressions and Flow Control
Distinguish between instance and local variables
Describe how to initialize instance variables
Recognize, describe, and use Java software operators
Distinguish between legal and illegal assignments of primitive types
Identify boolean expressions and their requirements in control constructs
Recognize assignment compatibility and required casts in fundamental types
Use if, switch, for, while, and do constructions and the labeled forms of break and continue as flow control structures in a

Arrays
Declare and create arrays of primitive, class, or array types
Explain why elements of an array are initialized
Explain how to initialize the elements of an array
Determine the number of elements in an array
Create a multidimensional array
Write code to copy array values from one array to another

Class Design
Define inheritance, polymorphism, overloading, overriding, and virtual method invocation
Use the access modifiers protected and the default (package-friendly)
Describe the concepts of constructor and method overloading
Describe the complete object construction and initialization operation

Advanced Class Features
Create static variables, methods, and initializers
Create final classes, methods, and variables
Create and use enumerated types
Use the static import statement
Create abstract classes and methods
Create and use an interface

Exceptions and Assertions
Define exceptions
Use try, catch, and finally statements
Describe exception categories
Identify common exceptions
Develop programs to handle your own exceptions
Use assertions
Distinguish appropriate and inappropriate uses of assertions
Enable assertions at runtime

Collections and Generics Framework
Describe the general purpose implementations of the core interfaces in the Collections framework
Examine the Map interface

Examine the legacy collection classes
Create natural and custom ordering by implementing the Comparable and Comparator interfaces
Use generic collections and type parameters in generic classes
Refactor existing non-generic code
Write a program to iterate over a collection
Examine the enhanced for loop

I/O Fundamentals
Write a program that uses command-line arguments and system properties
Examine the Properties class
Construct node and processing streams, and use them appropriately
Serialize and deserialize objects
Distinguish readers and writers from streams, and select appropriately between them

Console I/ O and File I/O
Read data from the console
Write data to the console
Describe files and file I/O

Building Java GUIs Using the Swing API
Describe the JFC Swing technology
Identify the Swing packages
Describe the GUI building blocks: containers, components, and layout managers
Examine top-level, general-purpose, and special-purpose properties of container
Examine components
Examine layout managers
Describe the Swing single-threaded model
Build a GUI using Swing components

Handling GUI-Generated Events
Define events and event handling
Examine the Java SE event model
Describe GUI behavior
Determine the user action that originated an event
Develop event listeners
Describe concurrency in Swing-based GUIs and describe the features of the SwingWorker class

GUI-Based Applications
Describe how to construct a menu bar, menu, and menu items in a Java GUI
Understand how to change the color and font of a component

Threads
Define a thread
Create separate threads in a Java technology program, controlling the code and data that are used by that thread
Control the execution of a thread and write platform-independent code with threads
Describe the difficulties that might arise when multiple threads share data
Use wait and notify to communicate between threads
Use synchronized to protect data from corruption

Networking
Develop code to set up the network connection
Understand TCP/IP

Use ServerSocket and Socket classes to implement TCP/IP clients and servers