Java Programming Language, Java SE 6

Duration: 0 Days

What you will learn

This Java Programming Language, Java SE 6 training is a foundational course that teaches you how to develop applications using the Java Programming Language. Expert Oracle University instructors will help you deep dive into the syntax of the Java language, objective-oriented programming, creating graphical user interfaces (GUIs), exceptions, file input/output (I/O), creating multi-threaded applications and networked applications.

Learn To:

Create an event-driven graphical user interface (GUI) using Swing components, including panels, buttons, labels, text fields and text areas.
Implement error-handling techniques using exception handling.
Use arrays and other data collections.
Use Java flow control constructs.
Use Java data types and expressions.

Benefits to You

By taking this course, programmers will get the chance to add the Java programming language to their list of skills. The knowledge you gain and the exercises you perform will also help prepare for the Oracle Certified Professional, Java SE 6 Programmer examination. Furthermore, if you're looking to widen your career opportunities, developing in demand Java programming expertise will make you more marketable to both current and future employers.

Participate in Hands-On Labs

You'll also get the chance to practice your skills with labs that range from simple to complex. You'll be led by experienced instructors who answer your questions and guide your learning experience.

Audience

Developer

Related Training

Required Prerequisites

Understand object-oriented principles

Be competent in creating programs in any programming language

Create and edit text files using a text editor

Suggested Prerequisites
Fundamentals of the Java Programming Language (CDJ-110-SE6)

Course Objectives

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 multi-threaded programs

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

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