

## Java Programming Language (SL-275-SE6)

<b>Course Name</b>	Java Programming Language (SL-275-SE6)
<b>Course Description</b>	The Java Programming Language course provides students with information about the syntax of the Java programming language; object-oriented programming with the Java programming language; creating graphical user interfaces (GUIs), exceptions, file input/output (I/O), and threads; and networking. Programmers familiar with object-oriented concepts can learn how to develop Java technology applications. The 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) 5.5.
<b>Target Population</b>	<ul style="list-style-type: none"> <li>• Programmers who are interested in adding the Java programming language to their list of skills</li> <li>• Students who are preparing for the Sun Certified Programmer for Java examination.</li> </ul>
<b>Pre-requisites</b>	<ul style="list-style-type: none"> <li>• Practical experience in any procedural programming language.</li> </ul>
<b>Course Objective</b>	<p>Upon completion of this course, students should be able to</p> <ul style="list-style-type: none"> <li>• Create Java technology applications that leverage the object-oriented features of the Java language, such as encapsulation, inheritance, and polymorphism</li> <li>• Execute a Java technology application from the command line</li> <li>• Use Java technology data types and expressions</li> <li>• Use Java technology flow control constructs</li> <li>• Use arrays and other data collections</li> <li>• Implement error-handling techniques using exception handling</li> <li>• Create an event-driven graphical user interface (GUI) using Swing components: panels, buttons, labels, text fields, and text areas</li> <li>• Implement input/output (I/O) functionality to read from and write to data and text files and understand advanced I/O streams</li> <li>• Create a simple Transmission Control Protocol/Internet Protocol (TCP/IP) networked client that communicates with a server through sockets</li> <li>• Create multithreaded programs</li> </ul>
<b>Course Topics</b>	<p>Module 1 - Getting Started</p> <p>Examine Java technology Analyze a simple Java technology application Execute a Java technology application</p>

## Module 2 - 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

## Module 3 - Identifiers, Keywords, and Types

- Use comments in a source program

- Distinguish between valid and invalid identifiers

- Recognize Java technology keywords

- List 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

- Describe default initialization

- Describe the significance of a reference variable

- State the consequence of assigning variables of class type

## Module 4 - 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 program

## Module 5 - 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

#### Module 6 - 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

#### Module 7 - 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

#### Module 8 - 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

#### Module 9 - 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

Use type parameters in generic classes

Refactor existing non-generic code

Write a program to iterate over a collection

Examine the enhanced for loop

#### Module 10 - 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

#### Module 11 - Console I/ O and File I/O

Read data from the console

Write data to the console

Describe files and file I/O

#### Module 12 - Building Java GUIs Using the Swing API

Describe the JFC Swing technology

Define Swing

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

#### Module 13 - 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

#### Module 14 - 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

	<p>Module 15 - Threads</p> <ul style="list-style-type: none"><li>Define a thread</li><li>Create separate threads in a Java technology program, controlling the code and data that are used by that thread</li><li>Control the execution of a thread and write platform-independent code with threads</li><li>Describe the difficulties that might arise when multiple threads share data</li><li>Use wait and notify to communicate between threads</li><li>Use synchronized to protect data from corruption</li></ul> <p>Module 16 - Networking</p> <ul style="list-style-type: none"><li>Develop code to set up the network connection</li><li>Understand TCP/IP</li><li>Use ServerSocket and Socket classes to implement TCP/IP clients and servers</li></ul>
<b>Course Duration</b>	5 Days (40 hours)