

# Java Programming Language (WJB-275A)

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 uses the Java 2 Software Development Kit, Standard Edition (J2SE SDK), version 5.0.

## Who Can Benefit

Students who can benefit from this course are programmers who are interested in adding the Java programming language to their skill set, and students who are preparing for the Sun Certified Programmer Exam for the Java Platform.



## Prerequisites

**To succeed fully in this course, students should be able to:**

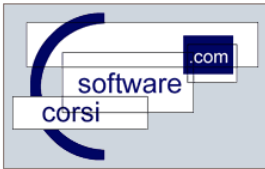
- Understand the principles of object-oriented programming.
- Create or compile simple programs in a programming language, such as C or C++, or have completed the SL-110: Fundamentals of the Java Programming Language course and have created and compiled simple Java programs.
- Use a text editor to create and edit text files.



## Skills Gained

**Upon completion of this course, students should be able to:**

- 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 Java technology GUI 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.
- Create multithreaded programs.
- Create a simple Transmission Control Protocol/Internet Protocol (TCP/IP) networked client that communicates with a server through sockets.



---

## Related Courses

### Before:

[Fundamentals of the Java Programming Language \(SL-110\)](#)

### After:

[Object-Oriented Analysis and Design Using UML \(OO-226\)](#)

[Java Programming Language Workshop \(SL-285\)](#)

Distributed Programming With Java Technology (SL-301)

GUI Construction with Java Foundation Classes (SL-320)



---

## Course Content

### Module 1 - Getting Started

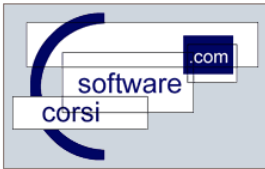
- Describe the key features of Java technology
- Write, compile, and run a simple Java technology application
- Describe the function of the Java Virtual Machine (JVM) Note: The terms "Java Virtual Machine" and "JVM" mean a Virtual Machine for the Java platform.
- Define garbage collection
- List the three tasks performed by the Java platform that handle code security

### Module 2 - Object-Oriented Programming

- Define modeling concepts: abstraction, encapsulation, and packages
- Discuss why you can reuse Java technology application code
- 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 application programming interface (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  
Identify and correct a Possible reference before assignment compiler error  
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**

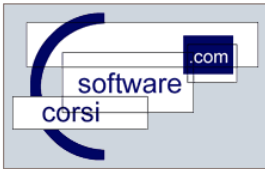
Define inheritance, polymorphism, overloading, overriding, and virtual method invocation  
Use the access modifiers protected and the default (package-friendly)  
Describe the concepts of constructor and method overloading  
Describe the complete object construction and initialization operation

#### **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 - Text-Based Applications**

- Write a program that uses command-line arguments and system properties
- Write a program that reads from standard input
- Describe the C-type formatted input and output
- Write a program that can create, read, and write files
- Describe the basic hierarchy of collections in the Java 2 Software Development Kit (Java 2 SDK)
- Write a program to iterate over a collection
- Write a program that uses generic collections

## **Module 10 - Building Java GUI's**

- Describe the Abstract Window Toolkit (AWT) package and its components
- Define the terms containers, components, and layout managers, and describe how they work together to build a GUI
- Use layout managers
- Use the FlowLayout, BorderLayout, and GridLayout managers to achieve a dynamic layout
- Add components to a container
- Use the Frame and Panel containers appropriately
- Describe how complex layouts with nested containers work

## **Module 11 - GUI Event handling**

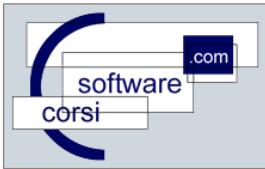
- Define events and event handling
- Write code to handle events that occur in a GUI
- Describe the concept of adapter classes, including how and when to use them
- Determine the user action that originated the event from the event object details
- Identify the appropriate listener interface for a variety of event types
- Create the appropriate event handler methods for a variety of event types
- Understand the use of inner classes and anonymous classes in event handling

## **Module 12 - GUI-Based Applications**

- Identify the key AWT components and the events that they trigger
- 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

## **Module 13 - 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

### **Module 14 - Advanced I/O Streams**

Describe the main features of the java.io package  
Construct node and processing streams, and use them appropriately  
Distinguish readers and writers from streams, and select appropriately between them

### **Module 15 - Networking**

Develop code to set up the network connection  
Understand the Transmission Control Protocol/Internet Protocol (TCP/IP)  
Use ServerSocket and Socket classes for implementation of TCP/IP clients and servers