

**Course Description:** Are you already familiar with the basics of the Java language? Are you interested in moving toward certification in Java? Move beyond the fundamental skills of Java syntax and logic flow into concepts. Design flexible and maintainable applications with interfaces. Create Graphical User Interfaces (GUI) for stand-alone applications and applets using Swing. Learn to manipulate and manage data using Java's Collection Application Programming Interface (API). Create threads of execution for multithreaded applications, and create simple distributed object systems through Java's Networking API and Java Remote Method Invocation (RMI). This course includes hands-on labs.

**Who Should Attend:** This is a basic course for Information Technology (IT) architects, designers, and developers.

**Prerequisites:** Students should have experience with Java.

**Benefits of Attendance:** Upon completion of this course, students will be able to:

- Describe JVM memory allocation and garbage collection
- Understand how to override the finalize() method
- Write code to access system properties
- Use arrays to store data
- Create an abstract class and use inheritance to develop a class hierarchy
- Override inherited methods
- Utilize inherited methods and constructors
- Define and use interfaces
- Use the final keyword to define constants
- Use the Collection API to manage and store objects
- Create and use data streams

### Course Outline:

#### Java review

difference between stand-alone applications, applets, and servlets  
 role of the JVM  
 Java development cycle  
 components of a class in Java  
 encapsulation and class instantiation  
 instances  
 keywords, operators, and primitive data types  
 implicit and explicit casting  
 String class  
 block structure of Java  
 conditional statements  
 iteration statements  
 arrays  
 command line parameters  
 purpose and value of wrapper classes

#### Java development environment

basic Java 2 Platform, Standard Edition (J2SE) Software Development Kit (SDK) tools  
 additional SDK tools  
 JVM garbage collection process  
 finalize() method  
 using the System class to obtain platform information  
 using the Runtime class to issue operating system commands

#### Inheritance, interfaces, and abstract classes

concept of inheritance  
 new classes using inheritance  
 superclass and a subclass  
 keywords relevant to subclassing  
 override methods  
 principle of dynamic binding  
 dynamic binding in applications

#### Manipulating objects using Java utility classes

basic Java collection mechanisms  
 hierarchy of interfaces and classes for dealing with collections  
 use of the appropriate concrete class, or interface for the task  
 date and text management classes

#### Streams

difference between byte streams and character streams  
 abstraction of byte streams through the InputStream and OutputStream classes  
 abstraction character streams through the Reader and Writer classes  
 streams using wrapper classes to enhance basic stream behavior  
 file Input/Output (I/O)  
 object serialization  
 serialization to save an object to a file and deserialization of object  
 transient keyword

#### Exceptions

throwable hierarchy  
 errors and exceptions  
 runtime and checked exceptions  
 try/catch/finally constructs  
 throw and throws to pass propagate exceptions  
 user-defined exceptions  
 overriding with exceptions

#### User interfaces Abstract Window Toolkit (AWT) and

##### Swing components

AWT and Swing  
 general organization of the AWT and Swing class structures  
 Component class hierarchy  
 graphical widgets  
 Java layout managers  
 process of setting a container layout manager  
 layout managers used to achieve proper GUI layout  
 containment hierarchy

#### Applets

applets and applications  
 life cycle of an applet and its inherited methods  
 embedding an applet into an HTML document  
 passing parameters from an HTML document to its contained applet  
 applet security restrictions  
 converting an applet into an application  
 converting an application into an applet

#### Security

Java-related security concepts  
 Java security model  
 digital signatures and certificates

#### Threads and synchronization

creating and instantiating threads using two different techniques  
 controlling single-thread flow using many thread methods  
 four thread states and their relationships to thread methods  
 synchronization in relation to object monitors  
 synchronized methods and blocks  
 thread racing using thread synchronization  
 thread deadlock  
 system event queue

#### Networking and RMI

Socket APIs in java.net  
 URL-named network resources  
 RMI concept, methods, and interfaces  
 RMI lookup facilities  
 converting an application to a client/server application