

## Java Programming Language – Fundamentals

(5 Days)

### Course Description:

The Fundamentals of the Java Programming Language course provides students with an excellent choice for beginning to learn programming using the Java programming language. This course is intended for students with little or no programming experience. The course teaches the significance of object-oriented programming, the keywords and constructs of the Java programming language, and the steps required to create simple Java technology programs. Students taking this course can receive a solid basis in the Java programming language upon which to base continued work and training. The course features the Java Platform, Standard Edition 6 (Java SE 6) platform, and uses the Java SE Development Kit 6 (JDK 6) product.

### Prerequisites:

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

- Create programs using a procedural language, such as C, or a scripting language, such as Perl
- Create and edit text files using a text editor
- Use a World Wide Web (WWW) browser
- Solve logic problems
- Describe the concept of a variable
- Execute commands using a command-line interface

### Skills Gained

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

- Demonstrate knowledge of Java technology, the Java programming language, and the product life cycle
- Use various Java programming language constructs to create several Java technology applications
- Use decision and looping constructs and methods to dictate program flow
- Implement intermediate Java technology programming and object-oriented (OO) concepts in Java technology programs

### Course Content

#### Module 1 - Explaining Java Technology

- Describe key concepts of the Java programming language
- List the three Java technology product groups
- Summarize each of the seven stages of the product life cycle

## **Module 2 - Analyzing a Problem and Designing a Solution**

- Analyze a problem using object-oriented analysis
- Design classes from which objects will be created

## **Module 3 - Developing and Testing a Java Technology Program**

- Identify the four components of a class in the Java programming language
- Use the main method in a test class to run a Java technology program from the command line
- Compile and execute a Java technology program

## **Module 4 - Declaring, Initializing, and Using Variables**

- Identify the use the syntax for variables and define the syntax for a variable
- List the eight Java programming language primitive data types
- Declare, initialize, and use variables and constants according to Java programming language guidelines and coding standards
- Modify variable values using operators
- Use promotion and type casting

## **Module 5 - Creating and Using Objects**

- Declare, instantiate, and initialize object reference variables
- Compare how object reference variables are stored in relation to primitive variables
- Use a class (the String class) included in the Java Software Developer Kit (SDK)
- Use the Java 2 Platform, Standard Edition (J2SE[™]) class library specification to learn about other classes in this application programming interface (API)

## **Module 6 - Using Operators and Decision Constructs**

- Identify relational and conditional operators
- Create if and if/else constructs
- Use the switch construct

## **Module 7 - Using Loop Constructs**

- Create while loops
- Develop for loops
- Create do/while loops

## **Module 8 - Developing and Using Methods**

- Describe the advantages of methods and define worker and calling methods
- Declare and invoke a method
- Compare object and static methods
- Use overloaded methods

## **Module 9 - Implementing Encapsulation and Constructors**

- Use encapsulation to protect data
- Create constructors to initialize objects

## **Module 10 - Creating and Using Arrays**

- Code one-dimensional arrays
- Set array values using length attribute and a loop
- Pass arguments to the main method for use in a program
- Create two-dimensional arrays

## **Module 11 - Implementing Inheritance**

- Define and test your use of inheritance
- Explain abstraction
- Explicitly identify class libraries used in your code