

# Java: Start Your Engines

## Course Outline

### **Course Length:**

5 days

### **Audience:**

This course is intended for IT professionals who want to become familiar with Java as a programming language but have no prior programming experience.

### **Objectives:**

- ❑ Understand the importance of Java and how it is being used today in software development.
- ❑ Understand object oriented programming (OOP) versus procedural programming.
- ❑ Become familiar with OOAD methodologies and how to analyze and design an application from an object-oriented viewpoint.
- ❑ Understand classes, objects, attributes, and behaviors.
- ❑ Use UML to visually depict the classes in a problem domain.
- ❑ Become familiar with the Java language and tools using Java 2 (JDK 1.2).
- ❑ Become familiar with Java's built-in data types and the java.lang.String class.
- ❑ Be able to write a class in Java, including determining the appropriate member variables and methods.
- ❑ Learn how to make decisions in Java using the if/else and switch statements.
- ❑ Learn how to perform repetition in Java using for and while loops.
- ❑ Understand arrays.
- ❑ Become familiar with inheritance.
- ❑ Understand Java applets and how to write and display one in a web page.

# Java: Start Your Engines

## Module Descriptions

### Module 1: An Introduction to Java

- ❑ **Java Overview:** What Java is and why it is so popular today.
- ❑ **The Lifecycle of Other Programs:** A look at how programs are developed in other languages.
- ❑ **The Lifecycle of a Java Program:** A look at how Java programs are developed.
- ❑ **Performance Issues:** Translating vs. Interpreting.
- ❑ **A Simple Java Program:** Writing a class in Java with a main() in it.
- ❑ **The Java Development Kit:** The tools used to create a Java program.
- ❑ **Writing a Java Program:** The details of creating and running a Java program.

### Module 2: OOAD (Object Oriented Analysis & Design)

- ❑ **OOP and Procedural Languages:** Today's common programming languages.
- ❑ **Writing a Program Procedurally:** An overview of how procedural programs are designed.
- ❑ **Writing a Program Using Objects:** An overview of OOP.
- ❑ **Classes and Object:** The fundamental components of an object-oriented program.
- ❑ **An Introduction to OOAD:** Object Oriented Analysis and Design.
- ❑ **UML:** The Unified Modeling Language.

### Module 3: The Java Programming Language

- ❑ **Java Keywords:** The Java language.
- ❑ **Identifiers:** Names used to identify the various parts of a program, like names of classes, methods and fields.
- ❑ **The Built-in Data Types:** The way Java stores data.
- ❑ **The String Class:** A useful class in Java.
- ❑ **Declaring Variables:** Allocating memory for data.
- ❑ **Arithmetic Operators:** The syntax of the Java operators and their order of operation.
- ❑ **Parameters and Return Values:** How to determine if a method needs parameters and/or a return value.

#### **Module 4: Classes and Objects**

- ❑ **Classes and Objects:** An object is an instance of a class.
- ❑ **Writing a Class in Java:** Determining fields and methods.
- ❑ **Instantiating Objects:** The “new” keyword.
- ❑ **Using Objects:** The dot operator.
- ❑ **Understanding References:** Understanding the difference between a reference and an object.

#### **Module 5: Methods**

- ❑ **Methods:** The signature of a method.
- ❑ **Invoking Methods:** Using the dot operator.
- ❑ **Passing References by Value:** Understanding call-by-value.
- ❑ **Invoking Methods:** The dot operator.
- ❑ **Local Variables vs. Member Variables:** Understanding the difference.
- ❑ **Constructors:** A special type of method that allows an object to be initialized when it is instantiated.

#### **Module 6: Control Structures**

- ❑ **Flow of Control:** Java’s control structures.
- ❑ **Comparison Operators**
- ❑ **Boolean Expressions:** Logic and the Boolean operators.
- ❑ **The if Statement:** The basic tool for making decisions.
- ❑ **The if/else Statement:** Extending an if statement.
- ❑ **The switch Statement:** Another decision maker.
- ❑ **The while Loop:** Repeating code.
- ❑ **The do/while Loop:** A variation of the while loop.
- ❑ **The for Loop:** Useful for repeating a specific number of tasks.

#### **Module 7: Arrays**

- ❑ **Arrays:** Contiguous memory for storing data.
- ❑ **Array References**
- ❑ **Array Objects:** Instantiating arrays.
- ❑ **Arrays of Primitive Data Types**
- ❑ **Arrays of Objects**

**Module 8: Inheritance**

- ❑ **Inheritance:** Creating new classes from existing classes.
- ❑ **The “is a” Relationship:** Determining when inheritance is a good design.
- ❑ **The extends Keyword:** Implementing inheritance in Java.
- ❑ **What Gets Inherited:** Understanding what a child inherits from its parent.
- ❑ **Single Inheritance:** A child can only have one parent.
- ❑ **Method Overriding:** A child class overriding a behavior of the parent class.

**Module 9: Applets**

- ❑ **An Overview of Applets:** A Java program that runs in a web browser.
- ❑ **The Applet Class:** The parent class of all applets.
- ❑ **The Methods of the Applet Class:** init(), start(), stop(), destroy() and paint().
- ❑ **Embedding an Applet in a Webpage:** The <applet> tag in HTML.
- ❑ **The Graphics Class:** Used for drawing in the applet.
- ❑ **Parameters:** Allows the HTML to pass data to the applet.

# Java: Start Your Engines Weekly Schedule

The following is a tentative schedule for the pacing of the course. The actual flow of the course may vary.

## **Day One**

Module 1: An Introduction to Java

Module 2: OOAD

Module 3: The Java Programming Language

## **Day Two**

Module 4: Classes and Objects

Module 5: Methods

## **Day Three**

Module 6: Control Structures

## **Day Four**

Module 7: Arrays

Module 8: Inheritance

## **Day Five**

Module 9: Applets