# JAVA INTRODUCTION

***This course provides students who have little or no programming experience with the basics of programming using the Java programming language.***

**Overall Objective**

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 programs.

**Audience**

The course is intended for technical writers, web developers, technical managers, and individuals with a technical, non-programming background, such as system administrators. It is also appropriate for novice programmers and for those programmers who prefer to start learning the Java programming language at an introductory level.

**Prerequisites**

Prior to attending this course, individuals are encouraged to have had some programming experience, whether with a scripting language, such as Perl, or a third-generation language, such as Basic or 'C'. Knowledge of object-oriented programming or design is not necessary.

**Duration**

Four days. The course uses desk quizzes and hands-on practical exercises to enable the delegate to gain experience in developing Java applications.

**Course objectives**

On completion of this course the delegate will be able to:

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

**Course contents**

*Introduction to Java Programming*

What is Java?; Applications and applets; Development Environments; JDK; The Java Virtual Machine (JVM); Producing, compiling and executing; Just-In-Time compilation; CLASSPATH.

*Introduction to Object-Oriented Programming*

What is OOP?; Objects and Classes; using Objects; Object references; Constructors; Access modifiers; Packages; Inheritance; Polymorphism; Encapsulation; OOD; the Java API.

*Data types and Operators*

Coding a program; Reserved words; Data types, variables and operators; Statements and expressions; Strings; StringBuffer; arrays; arguments to main(); garbage collection.

*Flow Control*

Decision-making; program flow control; loops.

*Classes and Methods*

More on objects; passing arguments; Polymorphism and Abstract classes; constructors and initialization; accessors and mutators; the *this* keyword; interfaces; method signatures; casting and instanceof; static and final modifiers; overriding and overloading.

*Collections*

Sets; Lists; Vectors; Maps.