Java Programming Language

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991 and first released in 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

INTRODUCTION TO JAVA

- History and Features of Java
- Comparison of C, C++, and Java
- Java Versions and its domain areas
- Life cycle of Java program
- Writing first Java program

DATA TYPES, IDENTIFIERS AND VARIABLES

CLASSES & OBJECTS

- The concepts of OOPS
- What is the class and object
- How to create a class and object

METHODS IN JAVA

RELATIONSHIP BETWEEN OBJECTS

INHERITANCE & POLYMORPHISM

- Concept of Inheritance
- The 'this' and 'super' keyword
- The introduction to Object class (the ultimate super-class) and its methods
- The garbage collection in java

EXCEPTION HANDLING

MULTI THREADING

WHAT IS A JAVA THREAD

- Need of thread in Java
- The constructor and methods of Thread class
- Life cycle of Java thread
- Writing Thread using Thread Class and Runnable Interface
- Daemon and Non-Daemon threads
- Synchronization in java

DATABASE CONNECTIVITY (THE JDBC)

ARRAYS

PACKAGES

- Concept of Packages
- Need of packages
- How to create packages using compiler
- How to use packages

STRING

- What is String
- Creating String literals
- The constructors of String
- The methods of String
- String immutability and its use

STRING BUFFER & STRING BUILDER

- What is StringBuffer class
- The constructors of StringBuffer
- The methods of StringBuffer
- The StringBuilder and its usage

REFLECTION API

- What is reflection API
- Need of reflection
- The Method, Field, Constructor, and Modifier class
- Implementation of the reflection concepts
- Access private members of a class using Reflection API

SERVLETS AND JSP

- Introduction of Servlet
- Difference b/w CGI, PHP, ASP, and Servlet
- Lifecycle (callback) Methods Of Servlet
- Servlet Implementation & Configuration
- ServletRequest and ServletResponse Interface and their method
- The JSP (Java Server Pages)
- The concept of Java Beans
- Using Java Beans with the JSP

Struts 2.x

- Introduction Of MVC Design Pattern
- Introduction of Sututs2
- Struts2.x Architecture
- Introduction of Component of Struts 2 like: Action, Results & Interceptors
- Heart of Struts2.x i.e. Interceptor
- Working of Param Interceptor
- Working of Model Driven Interceptor

- Working of ServletConfig Interceptor
- Working of ExecAndwait Interceptor
- Working of Validation Interceptor

Spring2.x/3.x

- Spring 2.x Overview
- All Modules Of Spring
- Introduction to Inversion Of Control configuration of Spring
- Dependency Injection
- Constructor Injection
- Setter Injection
- Object Creation using Factory Method
- Lifecycle Management Of Objects
- Auto wiring Of Beans using Xml
- Auto Discovery Of Beans using Xml
- Introducing Annotation in Spring
- Auto wiring Of Beans using Annotation
- Introduction to Aspect Oriented Programming
- Using Spring with JDBC
- Use of JDBC Template

Project work is compulsory after the completion of training program.