# Software Development and Construction

# Lab Manual 01

## Java - Overview

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be Write Once, Run Anywhere.

Java is –

**Object Oriented** – In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

**Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

**Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

**Secure** − With Java's secure feature it enables us to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

**Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

**Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

**Robust** − Java tries to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

**Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

**Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

**High Performance** − With the use of Just-In-Time compilers, Java enables high performance.

**Distributed** − Java is designed for the distributed environment of the internet.

**Dynamic** − Java is more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

# Java - Environment Setup

## Local Environment Setup

If you are still willing to set up your environment for Java programming language, then this section guides you on how to download and set up Java on your machine. Following are the steps to set up the environment.

Java SE is freely available from the link Download Java. You can download a version based on your operating system.

Follow the instructions to download Java and run .exe to install Java on your machine. Once you installed Java on your machine, you will need to set environment variables to point to correct installation directories –

Setting Up the Path for Windows

Assuming you have installed Java in c:\Program Files\java\jdk directory –

- Right-click on 'My Computer' and select 'Properties'.
- Click the 'Environment variables' button under the 'Advanced' tab.
- Now, alter the 'Path' variable so that it also contains the path to the Java executable. Example, if the path is currently set to 'C:\WINDOWS\SYSTEM32', then change your path to read 'C:\WINDOWS\SYSTEM32;c:\Program Files\java\jdk\bin'.

Setting Up the Path for Linux, UNIX, Solaris, FreeBSD

Environment variable PATH should be set to point to where the Java binaries have been installed. Refer to your shell documentation if you have trouble doing this.

Example, if you use bash as your shell, then you would add the following line to the end of your '.bashrc: export PATH = /path/to/java:$PATH'

## Popular Java Editors

To write your Java programs, you will need a text editor. There are even more sophisticated IDEs available in the market. But for now, you can consider one of the following –

- **Notepad** – On Windows machine, you can use any simple text editor like Notepad (Recommended for this tutorial), TextPad.
- **Netbeans** – A Java IDE that is open-source and free which can be downloaded from https://www.netbeans.org/index.html.
- **Eclipse** – A Java IDE developed by the eclipse open-source community and can be downloaded from https://www.eclipse.org/.
- **Visual Source Code** – VS Code supports almost every major programming language. Several ship in the box, like JavaScript, TypeScript, CSS, and HTML, but extensions for others can be found in the VS Code Marketplace. You can download from https://code.visualstudio.com/
- **IntelliJ IDEA** – It is undoubtedly the top-choice IDE for software developers. It makes Java and Kotlin development a more productive and enjoyable experience. You can download from https://www.jetbrains.com/idea/.

# Java Development Kit

Download link: https://docs.oracle.com/en/java/javase/17/

Installation Guidelines: https://docs.oracle.com/en/java/javase/17/install/overview-jdkinstallation.html

# Setting Up Project

Link to setup the project: https://code.visualstudio.com/docs/java/java-project with option no build tools.

# Java - Basic Syntax

When we consider a Java program, it can be defined as a collection of objects that communicate via invoking each other's methods. Let us now briefly look into what do class, object, methods, and instance variables mean.

- **Object** – Objects have states and behaviours. Example: A dog has states - color, name, breed as well as behaviour such as wagging their tail, barking, eating. An object is an instance of a class.
- **Class** – A class can be defined as a template/blueprint that describes the behaviour/state that the object of its type supports.
- **Methods** – A method is basically a behaviour. A class can contain many methods. It is in methods where the logics are written, data is manipulated, and all the actions are executed.
- **Instance Variables** – Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

**First Java Program**

Let us look at a simple code that will print the words **Hello World.**

**Example public class MyFirstJavaProgram**

```
{
  /* This is my first java program.
   * This will print 'Hello World' as the output
   */
  public static void main(String []args) {
    System.out.println("Hello World"); // prints Hello World
  }
}
```

Let's look at how to save the file, compile, and run the program. Please follow the subsequent steps −

- Open notepad and add the code as above.
- Save the file as: MyFirstJavaProgram.java.
- Open a command prompt window and go to the directory where you saved the class. Assume it's C:\.
- Type 'javac MyFirstJavaProgram.java' and press enter to compile your code. If there are no errors in your code, the command prompt will take you to the next line (Assumption: The path variable is set).
- Now, type ' java MyFirstJavaProgram ' to run your program.
- You will be able to see ' Hello World ' printed on the window.

## Basic Syntax

About Java programs, it is very important to keep in mind the following points.

- Case Sensitivity − Java is case sensitive, which means identifier Hello and hello would have different meaning in Java.
- Class Names − For all class names the first letter should be in Upper Case. If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.

  Example: class MyFirstJavaClass
- Method Names − All method names should start with a Lower Case letter. If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.

  Example: public void myMethodName()
- Program File Name − Name of the program file should exactly match the class name.
- When saving the file, you should save it using the class name (Remember Java is case sensitive) and append '.java' to the end of the name (if the file name and the class name do not match, your program will not compile).
- But please make a note that if you do not have a public class present in the file then file name can be different than class name. It is also not mandatory to have a public class in the file.

  Example: Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as 'MyFirstJavaProgram.java'

- public static void main (String args[]) – Java program processing starts from the main() method which is a mandatory part of every Java program.

## Java Basics

Syntax Review – https://www.w3schools.com/java/java_syntax.asp

Java Outputs – https://www.w3schools.com/java/java_output.asp

Java Comments – https://www.w3schools.com/java/java_comments.asp

Java Variables – https://www.w3schools.com/java/java_variables.asp

Java Datatypes – https://www.w3schools.com/java/java_data_types.asp

## Self-Study

Study how to take input from keyboard and implement one example of it.

## Lab Exercise

1. Write a Java program to convert temperature from Fahrenheit to Celsius degrees.
2. Write a Java program to convert inches into meters.
3. Write a Java program to meters into kilometers.
4. Write a Java program to convert minutes into years and days.
5. Write a Java program to print ranges of all primitive data types.