Chapter 01 - Introduction to Java

What is Java?

- Java is a programming language and a platform.
- Java is:
    - Object oriented (partially)
    - Concurrent
    - Secure
    - General purpose
    - High level
    - Robust

What is platform?

- Any hardware or software environment in which a program runs is called as a platform.
- Since Java has a runtime environment (JRE) and an API, it is also a platform.

History of Java

- Developed by Sun Microsystem which is now a subsidiary of Oracle.
- Was developed in the year 1995.
- James Gosling is known as Father of Java.

Applications of Java

- Java is used to develop:
    - Desktop applications
    - Web applications
    - Enterprise applications
    - Mobile applications
    - Embedded systems
    - Smart cards
    - Robotics
    - Games, etc

Types of Java applications

- There are mainly 4 types of applications that are created using Java:

Standalone applications

- Desktop application or window-based application.
- Traditional softwares that needs to be installed on the system to execute it.
- Can be developed using AWT and Swing technologies.
- Examples: Music player, antivirus software, etc.

Web applications

- Application which runs on web is called as web application.
- Java creates web applications that runs on server side and dynamically generates web pages.

- Servlet, JSP, Spring, JSF, Struts are some of the popular technologies used to create web applications using Java.

Enterprise applications

- An application which is distributed in nature like banking applications is called as an Enterprise application.
- It has features like high security, load balancing, etc.

Mobile applications

- Applications developed for mobile devices.

Java Editions

- There are 4 Java editions:

Java Standard Edition (Java SE or JSE)

- Also called as Core Java.
- Includes core topics such as OOPs, regex, exception handling, multi-threading, collection framework, etc.
- Used to develop console applications.

Java Enterprise Edition (Java EE or JEE)

- Used to create web applications and enterprise applications.
- Built on top of Java SE.
- Includes topics like servlet, jsp, spring, web services, etc.

Java Micro Edition (Java ME)

- Dedicated to mobile applications.

Java FX

- Used to develop rich internet applications.

Features of Java

Simple

- Easy to learn.
- Simple easy syntax.
- Syntax derived from C/C++.
- Java has removed some complicated concepts and rarely used features like pointers, operator overloading, etc that were present in C/C++.

Object Oriented

- Java is partially object oriented as primitive data types, static method, etc do exists.
- Object oriented methodology helps to easily and efficiently maintain, debug and scale applications.
- It simplifies software development.

Platform Independent

- Java code can be executed on multiple platforms like Windows, Linux, Mac, etc.
- The Java source code is compiled by the compiler and converted into bytecode.

- This bytecode is platform independent code because it can run of multiple platforms.
- Java follows WORA (Write Once Run Anywhere) coding standard.

Secured

- Java is best known for its security because we can develop virus-free systems.
- Java is secured because:
    - No explicit pointer.
    - Java programs run inside a virtual machine sandbox.

Robust

- Robust means strong and healthy.
- Java is robust because:
    - Uses strong memory management.
    - Lack of pointers that avoids security problems.
    - Provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
    - Exception handling and the type checking mechanism in Java.

Architecture-Neutral

- Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.
- In C and C++, `int` data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture.
- However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.

Portable

- Java is portable because it facilitates you to carry the Java bytecode to any platform.

High Performance

- Java is faster than other traditional interpreted programming languages because Java bytecode is close to native code.
- It is still a little bit slower than a compiled language (e.g., C++).
- Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

Distributed

- Java is distributed because it facilitates users to create distributed applications in Java.

Multi-threaded

- A thread is like a separate program, executing concurrently.
- We can write Java programs that deal with many tasks at once by defining multiple threads.
- The main advantage of multi-threading is that it doesn't occupy memory for each thread.
- It shares a common memory area.

Dynamic

- It supports the dynamic loading of classes ie classes are loaded on demand.
- It also supports functions from its native languages, i.e., C and C++.