

UNIT-1

INTRODUCTION TO JAVA

Features of JAVA

- Simple
- Secure
- Portable
- Object-oriented
- Platform - independent
- Robust

- Multithreaded
- Architecture-neutral
- Interpreted
- High- performance
- Distributed
- Dynamic

Simple

- Java is **Easy to write** and more **readable**
- Java has an **easy syntax** to learn and use.
- Most of the concepts are drew from C++ thus making Java learning simpler.
- Java remove confusing concepts like **explicit pointers, operator overloading** etc.
- There is no need to remove unreferenced objects because there is **Automatic Garbage Collection** in java.

Secure

- Java provides secure way to access web applications.
- No use of explicit pointer
- Java Programs run inside **java virtual machine**
- Use many security algorithms for security perspective

Portable

- Java is portable because it use **bytecode**.
- It doesn't require any type of implementation specification.
- Java programs can execute in **JVM**.
- Java programs can be run on any platform (Linux, Window, Mac)

Object-oriented

- Java programming is object-oriented programming language.
- Like C++ java provides most of the object oriented features. But **Java is pure OOP Language**, while C++ is semi object oriented
- Everything in Java is an object. It means any software in JAVA is a combination of function and data.

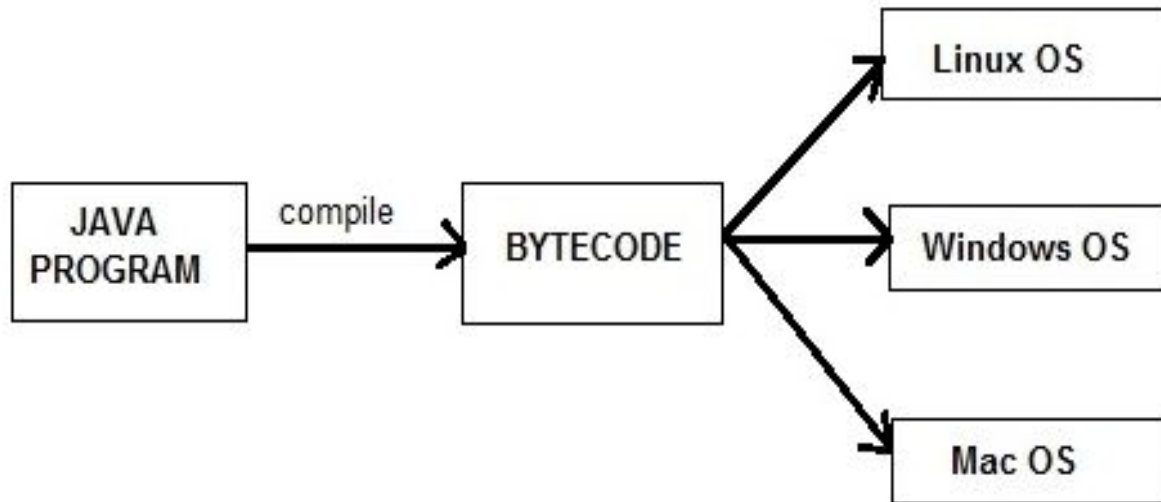
Object-oriented

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Platform-independent

- Other programming languages such as C, C++ etc which are **compiled** into **platform specific machines**. Java is guaranteed to be **write-once, run-anywhere(WORA)** language.
- Java program is compiled into **bytecode**.
- This **bytecode is platform independent** and can be run on any machine, plus this bytecode format also provide **security**.
- Any machine with Java Runtime Environment can run Java Programs.

Platform-independent



Robust

- Robust simply means **strong**. Java is robust because:
 - It uses **strong memory management**.
 - There are **lack of pointers** that avoids security problems.
 - There is **automatic garbage collection** in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
 - There is **exception handling** and **run time checking** mechanism 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.
- Benefit of multithreading is that it **utilizes same memory and other resources to execute multiple threads at the same time**, like While typing, grammatical errors are checked along.

Architecture-neutral

- Compiler generates bytecodes, which have **nothing to do with a particular computer architecture**, hence a Java program is easy to interpret on any machine.
- **Ex:-** it doesn't matter if your operating system is 32-bit or 64-bit, the Java byte code is exactly the same. You don't have to recompile your Java source code for 32-bit or 64-bit. (So, "architecture" refers to the CPU architecture).

Architecture-neutral

- No implementation of **dependent features**
- **Example:** the size of an int does not vary based on platform.

Interpreted

- JAVA is both **compiled and interpreted** language. The source code in JAVA program is **compiled to byte code** (instead of native machine code as we see in C/C++ or other languages).
- **At run time, JVM interprets the byte code and executes them.**

Interpreted

- As JAVA use JAVA compiler, it's compilation speed increase in the application. But, the problem was found in the interpretation. As compared to java compiler interpreter's speed becomes slow down the application.
- **JIT (Just In Time)** compiler used to interpret the bytecode to increase speed. It makes the execution faster.

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.
- Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, **Java enables high performance with the use of just-in-time compiler.**

Distributed

- **Java is a distributed language which means that the program can be design to run on computer networks.**
- Java provides an extensive library of classes for communicating ,using TCP/IP protocols such as HTTP and FTP. This makes creating network connections much easier than in C/C++.

Distributed

- It's easy to read and write objects on the remote sites via URL with the same ease that programmers which are used when read and write data from and to a file.
- This helps the programmers at remote locations to work together on the same project.
- **Example: RMI (Remote Method Invocation)**

Dynamic

- **Java is dynamic language because it is capable of linking in new class libraries, methods and objects.**
- It support native methods(methods which is written in another language like c/c++).Native methods are linked dynamically at runtime.

Dynamic

- Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time.
- **Dynamic binding of objects**