

Introduction to Java

Program: A program is a set of instructions, which are executed by a machine to reduce the burden of a user or a human being by performing the operations faster without any mistakes.

Software: It is a set of programs, which can perform multiple tasks.

The softwares are classified into two types:

- 1) **System software:** The softwares which are designed to interact or communicate with the hardware devices, to make them work are called as system software. These softwares are generally developed in languages like C, C++ etc.

Example: Operating System, drivers, compilers etc.

- 2) **Application Software:** The softwares which are called designed to store data, provide entertainment, process data, do business, generate reports etc. are called as application software. These softwares are generally developed in languages like java, .net, etc.

The application softwares are further classified into two types.

- i. **Stand Alone Software:** The software which can execute in the context of a single machine are called as stand alone software.

Example: MS-Word, media player, etc.

- ii. **Web Based Software:** The software which can execute on any machine in the context of a browser are called as web based software.

Example: Gmail, Facebook, etc

The java language is released by SUN Micro Systems in the year 1995 in three editions:

- 1) **JSE (Java Standard Edition):**

This edition can be used for developing stand alone software.

- 2) **JEE (Java Enterprise Edition):**

This edition can be used for developing web based software.

- 3) **JME (Java Mobile Edition):**

This edition can be used for developing applications for mobile devices, wireless devices, embedded controllers etc, where memory is limited.

Features of Java

The features of any programming language are the services or the facilities provided by that language. The various features of java language are:

- 1) **Simple:** The java language is called as simple programming language because of the following reasons:
 - a) The syntax of java language is similar to other programming languages like C, C+ etc. & therefore, simple to migrate from other languages.
 - b) The complex topics like pointers, templates etc. are eliminated from java making it simple.
 - c) In the java language the programmer is responsible for only allocation of memory. The deallocation of the memory is done by the garbage collector.

- 2) **Object Oriented:** The java language is called as object oriented language. Any language can be called as object oriented if the development of the application is based on objects and classes.

Object: Any entity that exists physically in this real world which requires some memory is called as object. Every object contains some properties and some actions. The properties are the data which describes the object and the actions are the tasks or the operations performed by the objects.

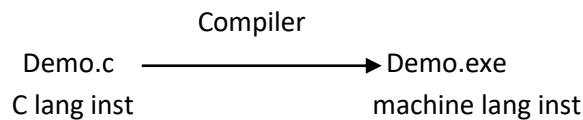
Class: A class is a collection of common properties, and common actions of a group of objects. A class can be considered as a plan or a model or a blue print for creating the objects. For every class we can create any number of objects and without a class object can't be created.

Example: Class: Student

Object: rajesh, amit

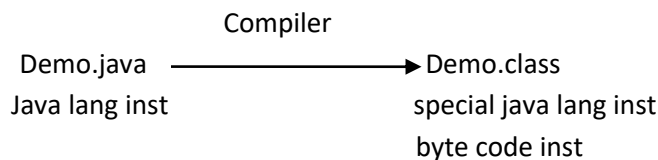
- 3) Secured: Security is one of the most important principles of any programming language. The java language contains in-built security programs for protecting the data from unauthorized usage.
- 4) Distributed: Using the distributed feature we can access the data available in multiple machines and provide it to the user. Using this feature we can improve the performance of the application by making the data more available and more accessible.
- 5) Platform Independent or Machine Independent or Architecture Neutral: The java program can be executed on any machine irrespective of their hardware, software, architecture, operating system etc therefore it is called as platform independent language.

C Language: When we compile a C program, the compiler verifies whether, the 'C' language instructions are valid not, if valid the compiler generates .exe file containing machine language instructions.



The machine language instructions available in the .exe files generated by the compiler can be executed only in that machine, where it is compiled. If we want to execute the C program in another machine, then we need to recompile and then execute. This nature of C language makes it machine dependent or platform dependent language.

Java Language: When we compile a java program, the compiler verifies whether the java language instructions are valid or not, if valid the compiler will generate .class file containing special java instructions (byte code instructions).

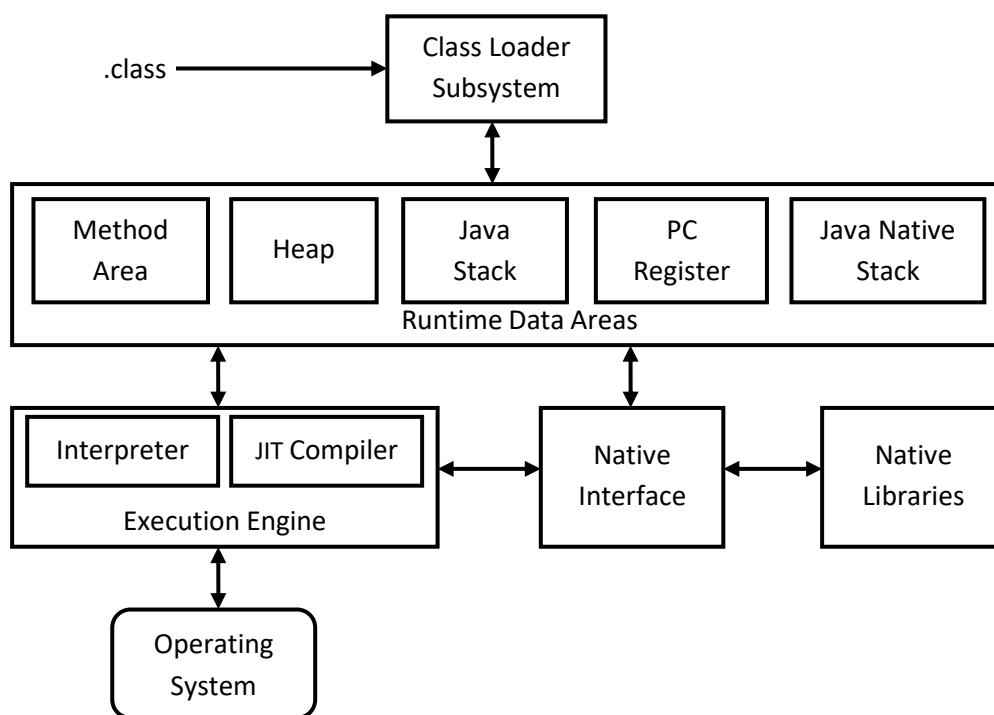


The special java instructions available in the .class file generated by the compiler can be executed on any machine with help of JVM, without recompiling it. This nature of java language makes it platform independent.

- 6) Interpreted: The java language is said to interpreted language as the execution of the program is done by the interpreter available inside the JVM.
- 7) High Performance: The execution of a java program is done by an interpreter along with a special compiler called JIT compiler, thereby reducing the execution time and improving the performance of the application.
- 8) Portable: The java language is said to be portable language using which we can develop an application which is a collection of small components which can be replaced and reused.
- 9) Multithreaded: A language is said to be multithreaded, if it supports multithreading. Every thread in java program is a control. If the program contains multiple controls then we can reduce the waiting time, and provide response faster and thereby improving the performance.

- 10) Dynamic:** The java language is said to be dynamic because the allocation of memory is done at execution time according to the requirement.
- 11) Robust:** The java language is said to be strong programming language, because of the following reasons:
- Memory management:** In java language the allocation of memory and deallocation of memory, both are efficient. During the memory allocation time, there will be no wastage of memory and deallocation is done by garbage collector which is also efficient as the unused memory will be removed.
 - Exception handling:** The errors that occur at runtime because of the logical failure or invalid inputs are called as exceptions. When an exception occurs, the application will be terminated abnormally and executed incompletely. In order to execute the code completely and terminate normally, we take the help of exception handling. The process of exception handling in java is simple and efficient.

JVM Architecture



Class loader subsystem: The class loader subsystem will take .class file as the input and performs the following operations:

- It will load the .class file into the JVM's memory
- Before loading the .class file, it will verify whether the byte code instructions are valid or not with the help of byte code verifier.
- If the byte code instructions are valid then it will load a byte code instruction into different areas of the JVM called runtime data areas.

Runtime data areas: These areas are available at runtime to store different code. The various runtime data areas are:

- Method area:** This area can be used for storing all the class code along with their method code.
- Heap:** This area can be used for storing the objects.

- 3) Java Stack: This area can be used for storing the methods that are under execution. The java stack can be considered as a collection of frames, where each frame will contain the information of only one method.
- 4) PC (Program Counter) Register: This register will contain the address of the next instruction. that has to be executed.
- 5) Java Native Stack: This area can be used for storing non-java code during the migration of the application from non java code to java code. Non-Java code will be called as native code.

Execution Engine: The execution engine is responsible for executing the java program. It contains two parts:

- 1) Interpreter
- 2) JIT (Just-In-Time) Compiler

Both the parts of the execution engine are responsible for executing the code parallelly, reducing the execution time, and thereby improving the performance of the application.

Note: The JIT compiler is designed by Hot Spot Technologies and the code executed by a JIT compiler will be called as hot spots.

Library: A library is a collection of pre-defined programs of a language.

Native Libraries: The collection of libraries of the non java languages will be together called as native libraries.

Native Interface: The native interface will help to load the native code from native libraries into the java native stack.

OS (Operating System): To execute a java program the JVM requires some resources like memory, processor etc from the machine. To get those resources from the machine, the JVM has to communicate with the operating system.