Object Oriented Thinking

Need for oop paradigm:

Two Paradigms of Programming:

As you know, all computer programs consist of two elements: code and data. Furthermore,a program can be conceptually organized around its code or around its data. That is, some programs are written around ―what is happening‖ and others are written around ―who is being affected.‖ These are the two paradigms that govern how a program is constructed.

The first way is called the process-oriented model. This approach characterizes a program as a series of linear steps (that is, code). The process-oriented model can be thought of as code acting on data. Procedural languages such as C employ this model to considerable success. Problems with this approach appear as programs grow larger and more complex. To manage increasing complexity, the second approach, called object-oriented programming, was conceived.

Object-oriented programming organizes a program around its data (that is, objects) and a set of well-defined interfaces to that data. An object-oriented program can be characterized as data controlling access to code. As you will see, by switching the controlling entity to data, you can achieve several organizational benefits.

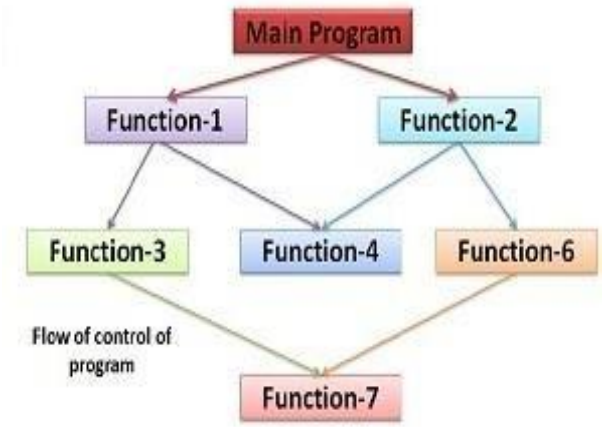
Procedure oriented Programming:

In this approach, the problem is always considered as a sequence of tasks to be done. A number of functions are written to accomplish these tasks. Here primary focus on ―Functions‖ and little attention on data.

There are many high level languages like COBOL, FORTRAN, PASCAL, C used for conventional programming commonly known as POP.

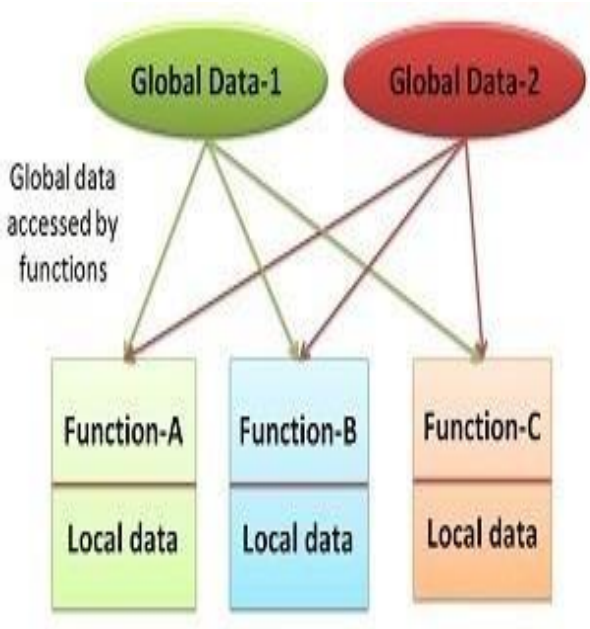
POP basically consists of writing a list of instructions for the computer to follow, and organizing these instructions into groups known as functions.

A typical POP structure is shown in below: Normally a flowchart is used to organize these actions and represent the flow of control logically sequential flow from one to another. In a multi-function program, many important data items are placed as global so that they may be accessed by all the functions. Each function may have its own local data. Global data are more vulnerable to an in advent change by a function. In a large program it is very difficult toidentify what data is used by which function. In case we need to revise an external data structure, we should also revise all the functions that access the data. This provides an opportunity for bugs to creep in.



Structure of procedure oriented program

Drawback: It does not model real world problems very well, because functions are actionoriented and do not really corresponding to the elements of the problem.



Characteristics of Procedure oriented Programming:

- Emphasis is on doing actions.
- Large programs are divided into smaller programs known as functions.
- Most of the functions shared global data.
- Data move openly around the program from function to function.
- Functions transform data from one form to another.

Employs top-down approach in program design

	Procedure Oriented Programming	Object Oriented Programming
Divided Into	In POP, program is divided into small parts called **functions**.	In OOP, program is divided into parts called **objects**.
Importance	In POP,Importance is not given to **data** but to functions as well as **sequence** of actions to be done.	In OOP, Importance is given to the data rather than procedures or functions because it works as a **real world**.
Approach	POP follows **Top Down approach**.	OOP follows **Bottom Up approach**.
Access Specifiers	POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
Expansion	To add new data and function in POP is not so easy.	OOP provides an easy way to add new data and function.
Data Access	In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system.	In OOP, data can not move easily from function to function,it can be kept public or private so we can control the access of data.
Data Hiding	POP does not have any proper way for hiding data so it is **less secure**.	OOP provides Data Hiding so provides **more security**.
Overloading	In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
Examples	Example of POP are : C, VB, FORTRAN, Pascal.	Example of OOP are : C++, JAVA, VB.NET, C#.NET.

Java Programming: Introduction

Born

This language was developed at SUN Microsystems in the year 1995 under the guidance of James Gosling and there team.

Overview of Java

Java is one of the programming language or technology used for developing web applications. Java language developed at SUN Micro Systems in the year 1995 under the guidance of James Gosling and there team. Originally SUN Micro Systems is one of the Academic University (Stanford University Network)

Whatever the software developed in the year 1990, SUN Micro Systems has released on the name of oak, which is original name of java (scientifically oak is one of the tree name). The OAK has taken 18 months to develop.

The oak is unable to fulfill all requirements of the industry. So James Gosling again reviews this oak and released with the name of java in the year 1995. Scientifically java is one of the coffee seed name.

Java divided into three categories, they are

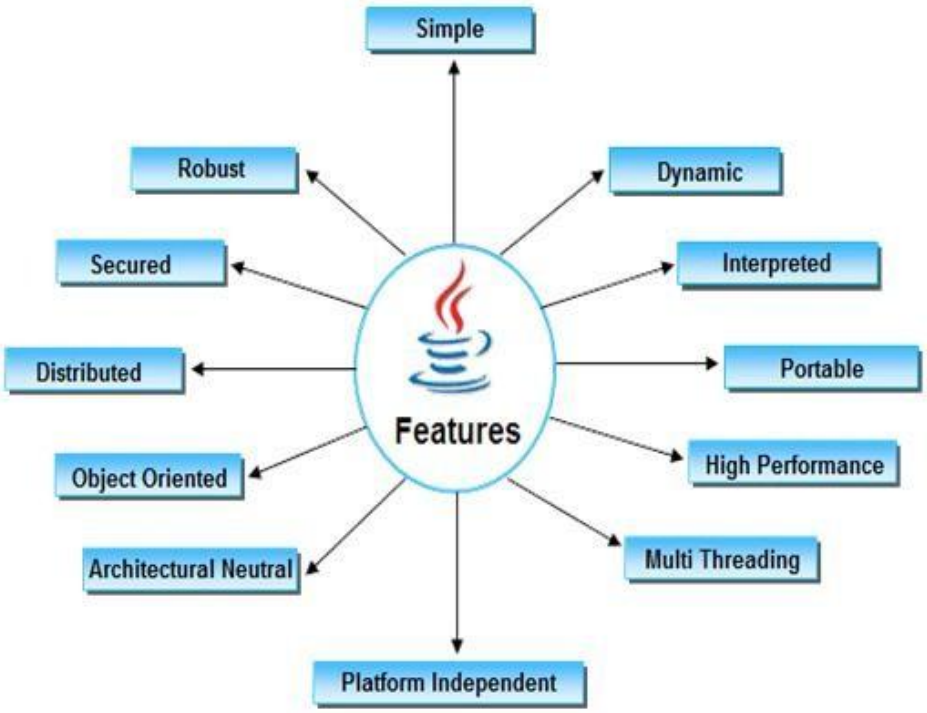
- J2SE (Java 2 Standard Edition)
- J2EE (Java 2 Enterprise Edition)

- J2ME (Java 2 Micro or Mobile Edition)

- J2SE is used for developing client side applications.
  J2EE is used for developing server side applications.
  J2ME
  J2ME is used for developing mobile or wireless application by making use of a predefined protocol called WAP (wireless Access / Application protocol).

Features(Buzzwords) of Java

Features of a language are nothing but the set of services or facilities provided by the language vendors to the industry programmers. Some important features of java are;

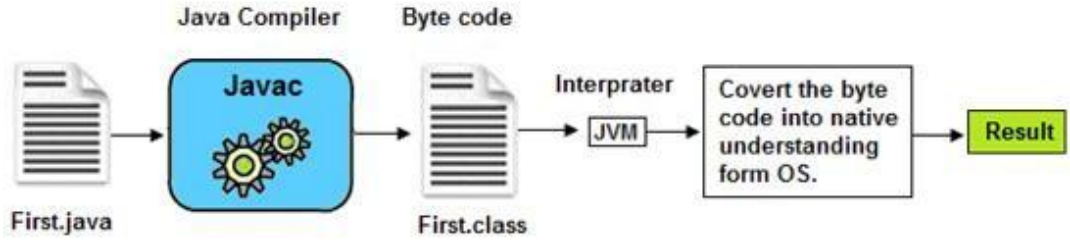


Important Features of Java

- Simple
- Platform Independent
- Architectural Neutral
- Portable
- Multi Threading
- Distributed
- Networked
- Robust
- Dynamic
- Secured
- High Performance
- Interpreted
- Object Oriented

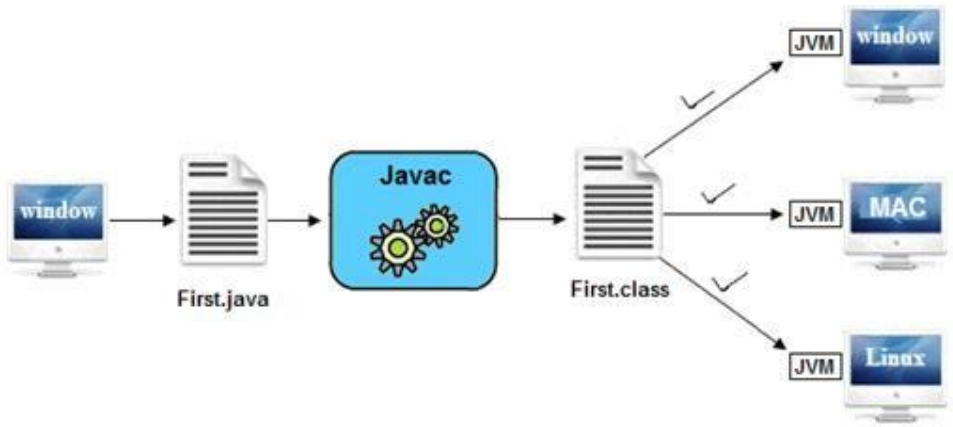
1. Simple
It is simple because of the following factors:

- It is free from pointer due to this execution time of application is improve. [Whenever we write a Java program without pointers then internally it is converted into the equivalent pointer program].
- It has Rich set of API (application protocol interface).
- It has Garbage Collector which is always used to collect un-Referenced (unused) Memory location for improving performance of a Java program.
- It contains user friendly syntax for developing any applications.



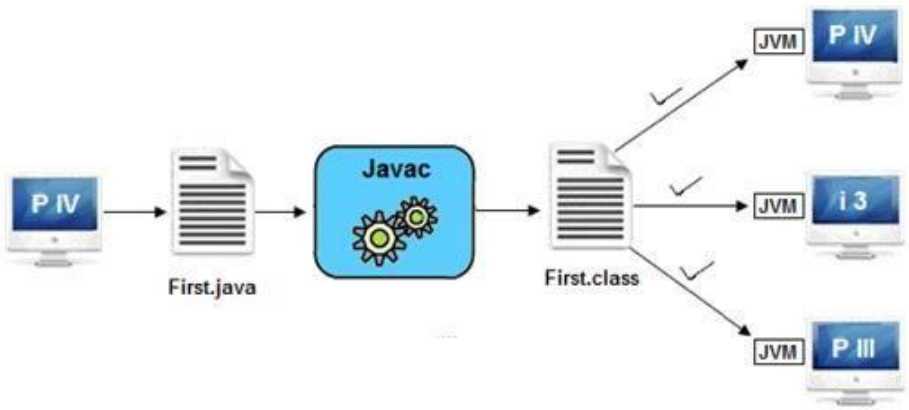
2. Platform Independent

A program or technology is said to be platform independent if and only if which can run on all available operating systems with respect to its development and compilation. (Platform represents O.S).



3. Architectural Neutral

Architecture represents processor. A Language or Technology is said to be Architectural neutral which can run on any available processors in the real world without considering there architecture and vendor (providers) irrespective to its development and compilation.



The languages like C, CPP are treated as architectural dependent.

4. Portable

If any language supports platform independent and architectural neutral feature known as portable. The languages like C, CPP, and Pascal are treated as non-portable language. It is a portable language.

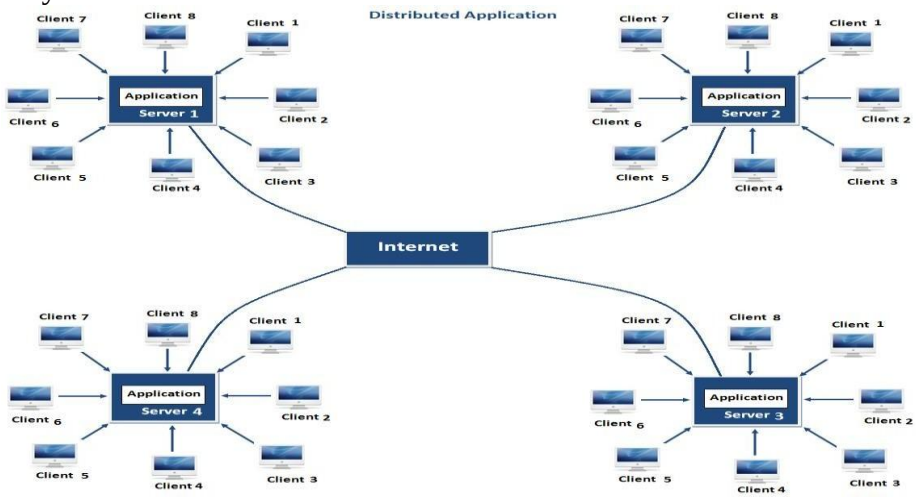
According to SUN Microsystems.

5. Multithreaded

A flow of control is known as thread. When any Language executes multiple threads at a time that language is known as multithreaded Language. It is multithreaded Language.

6. Distributed

Using this language we can create distributed application. RMI and EJB are used for creating distributed applications. In distributed application multiple client system are depends on multiple server systems so that even problem occurred in one server will never be reflected on any client system.



Note: In this architecture same application is distributed in multiple server system.

7. Networked

It is mainly design for web based applications; J2EE is used for developing network based applications.

8. Robust

Simply means of Robust is strong. It is robust or strong Programming Language because of its capability to handle Run-time Error, automatic garbage collection, lack of pointer concept, Exception Handling. All these points make it robust Language.

9. Dynamic

It supports Dynamic memory allocation, due to this memory wastage is reduced and performance of application is improved. The process of allocating the memory space to the input of the program at a run-time is known as dynamic memory allocation. To allocate memory space dynamically we use an operator called 'new'. 'new' operator is known as dynamic memory allocation operator.

10. Secure
It is more secured language compare to other language; in this language all code is converted into byte code after compilation which is not readable by human.

11. High performance
It has high performance because of following reasons;

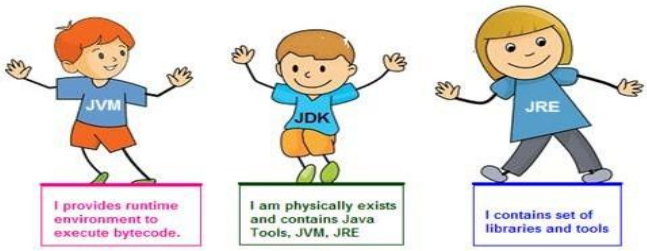
- This language uses Byte code which is faster than ordinary pointer code so Performance of this language is high.
- Garbage collector, collect the unused memory space and improve the performance of application.
- It has no pointers so that using this language we can develop an application very easily.
- It support multithreading, because of this time consuming process can be reduced to execute the program.

12. Interpreted
It is one of the highly interpreted programming languages.

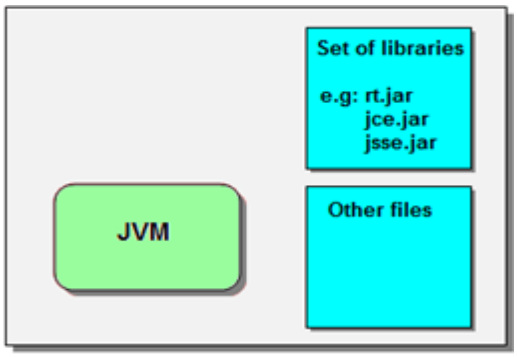
13. Object Oriented
It supports OOP's concepts because of this it is most secure language.

Difference between JDK, JVM and JRE

Jvm, Jre, Jdk these all the backbone of java language. Each component has separate works. Jdk and Jre physically exists but Jvm is abstract machine it means it not physically exists.

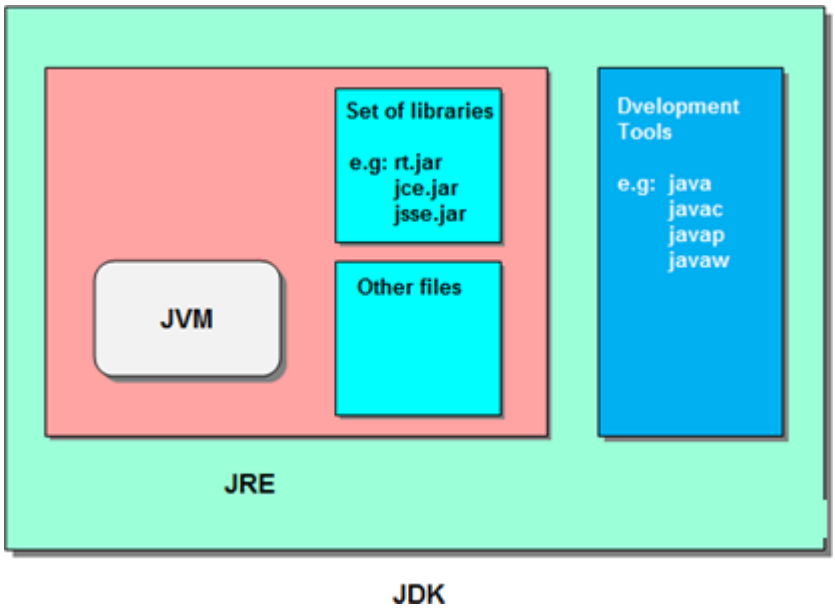


JVM
JVM (Java Virtual Machine) is a software. It is a specification that provides runtime environment in which java byte code can be executed. It not physically exists.
JVMs are not same for all hardware and software, for example for window os JVM is different and for Linux VJM is different. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.
JRE
The Java Runtime Environment (JRE) is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application. It physically exists. It contains set of libraries + other files that JVM uses at runtime.



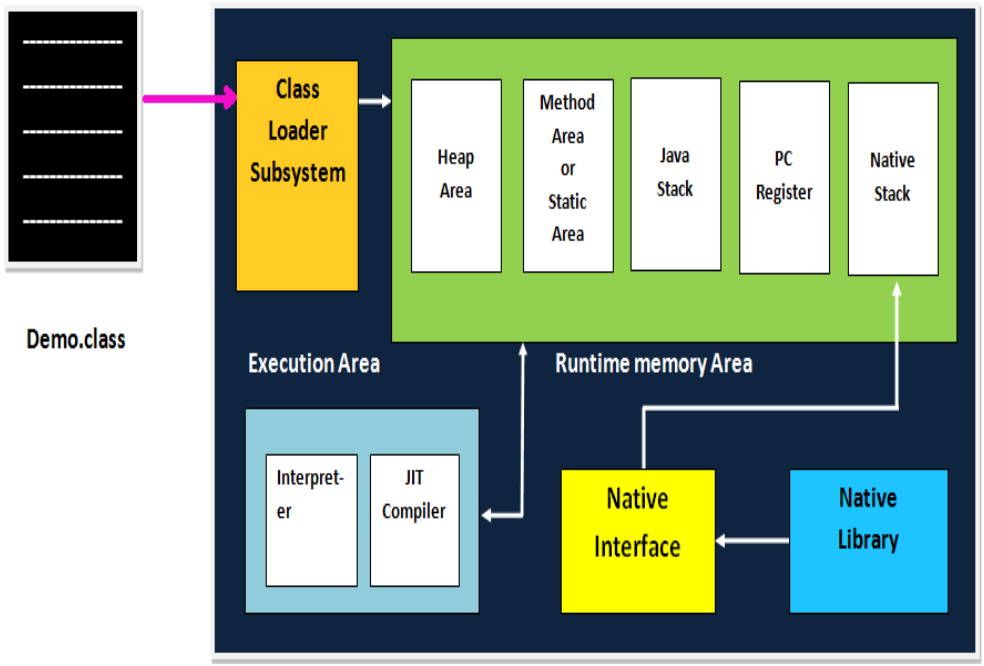
JDK

The Java Development Kit (JDK) is primary component. It physically exists. It is collection of programming tools and JRE, JVM.



JVM Architecture in Java

JVM (Java Virtual Machine) is software. It is a specification that provides Runtime environment in which java byte code can be executed.



Operation of JVM

JVM mainly performs following operations.

- Allocating sufficient memory space for the class properties.
- Provides runtime environment in which java byte code can be executed
- Converting byte code instruction into machine level instruction.

JVM is separately available for every Operating System while installing java software so that JVM is platform dependent.

Note: Java is platform Independent but JVM is platform dependent because every Operating system has different-different JVM which is install along with JDK Software.

Class loader subsystem:

Class loader subsystem will load the .class file into java stack and later sufficient memory will be allocated for all the properties of the java program into following five memory locations.

- Heap area
- Method area
- Java stack
- PC register
- Native stack

Heap area:

In which object references will be stored.

Method area

In which static variables non-static and static method will be stored.

Java Stack

In which all the non-static variable of class will be stored and whose address referred by object reference.

Pc Register

Which holds the address of next executable instruction that means that use the priority for the method in the execution process?

Native Stack

Native stack holds the instruction of native code (other than java code) native stack dependson native library. Native interface will access interface between native stack and native library.

Execution Engine

Which contains Interpreter and JIT compiler whenever any java program is executing at thefirst time interpreter will comes into picture and it converts one by one byte code instructioninto machine level instruction JIT compiler (just in time compiler) will comes into picture from the second time onward if the same java program is executing and it gives the machinelevel instruction to the process which are available in the buffer memory.

Note: The main aim of JIT compiler is to speed up the execution of java program.What is JIT and Why use JIT

JIT is the set of programs developed by SUN Micro System and added as a part of JVM, tospeed up the interpretation phase.

In the older version of java compilation phase is so faster than interpretation phase. Industryhas complained to the SUN Micro System saying that compilation phase is very faster and interpretation phase is very slow.

So solve this issue, SUN Micro System has developed a program called JIT (just in time compiler) and added as a part of JVM to speed up the interpretation phase. In the current version of java interpretation phase is so faster than compilation phase. Hence java is one ofthe highly interpreted programming languages.