Unit - II

Features of Java →

① Compiled and Interpreted
② Platform-Independent and Portable → (WORA)
③ Object oriented → fault tolerance
④ Robust & Secure → (Because of no pointer (no freedom to access the memory) use only reference data item
⑤ Distributed
⑥ Simple, Small and familiar
⑦ Multithreaded and Interactive → (Simultaneously running of Threads)
⑧ High Performance
⑨ Dynamic and extensible
⑩ ease of development
⑪ Scalability and performance
⑫ Monitoring and Manageability
⑬ Desktop client
⑭ Core XML Support
⑮ JDBC RowSet
⑯ Garbage Collector → to delete unused memory by using special program

Java & C →

① Java does not include c unique statement keyword sizeof and typedef.
② It ~~does not support~~ contain the data type struct and union.
③ It does not define the type modifier keywords auto, extern, register, signed & unsigned.
④ not support an explicit pointer type
⑤ It does not have a preprocessor and can not use #define, #include
⑥ It cannot use void keyword if no arguments passed in function
⑦ It adds new operators such as instanceof and >>>

Java and C++ → ① Java does not support operator overloading

② It does not have template classes as in C++

③ It does not support multiple inheritance of classes. It is accomplished with new feature called interface.

④ It does not support global variables. every variable and method is declared within a class and form part of that class.

⑤ It does not use pointers.

⑥ It has replaced the destructor function with a finalize() fn.

⑦ There are no header files in Java.

Java Environment → Java environment includes a large no. of development tools and hundreds of classes and methods. The development tools are part of the system known as Java development kit (JDK).

JDK → It comes with a collections of tools that are used for developing and running Java Program. They include

- → Javac (Java compiler)
- → Java (Java interpreter)
- → Javap (Java disassembler)
- → Javah (for c header files)
- → Javadoc (For creating HTML documents)
- → Jdb (Java debugger)

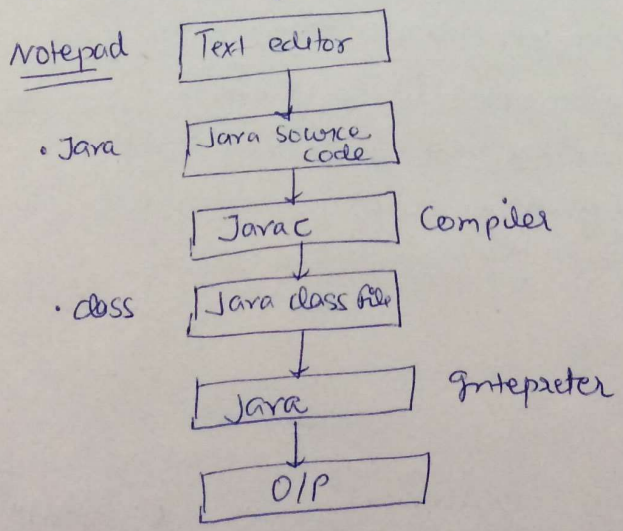
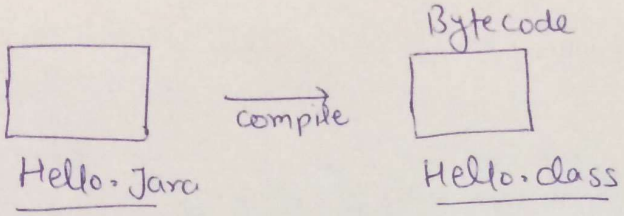


Fig. Process of Building and running Java programs

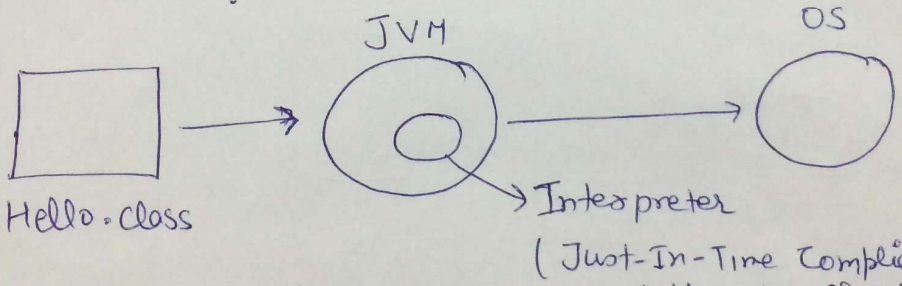
Compile and Run →

(1) How to compile ?



→ (1) There is no different compiler for different operating system
(2) we get byte code after compiling.
(3) we can distribute this Bytecode to different user for their execution
(4) if there are five classes in source code then after compiling .java file, different five .class ~~files~~ will be generated.

(2) How to Run ?



(1) To run the byte code on different Platform we need to install JVM.
(2) Compiler translate the whole program into another format & store it, for later execution.
(3) Intrepreter read every line one by one & translate it, send it to OS. It does not store for later execution
(4) Due to ~~lack~~ wastage in time execution of Java Programe by interpreter, ~~which a~~ They have done changes in Interpreter which will ~~be~~ work some what like compiler so called JIT Compiler. i.e Interpreter will read the next line and translate it, store it for next instruction execution.
(5) So same ByteCode can be run on different Platform but JVM must be installed.
(6) JVM is Platform dependent.

Java Programming

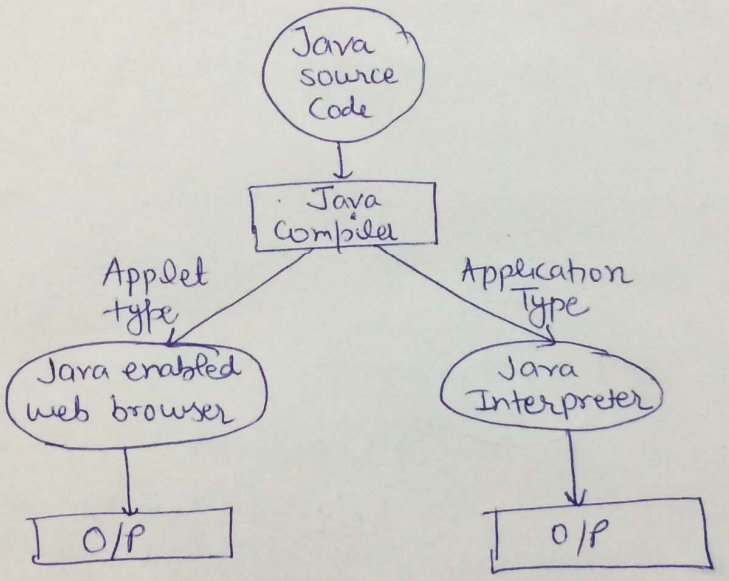
We can develop two types of Java Programs

(1) Stand-alone application (Desktop App^n)

(2) Web applets.

→ It includes two steps (1) compiling using Javac
(2) executing using Java

→ These are small Java programs developed for internet application. An applet located on a distant computer (server) can be downloaded via internet and executed on a local computer (client) using a Java capable browser. It can only run within a web browser.



Note (1) Java resides in mobiles, client, server machines, Smart phones cloud etc.

(2) Principle of Java → Write Once, Run anywhere (WORA)

Java flavours : (1) Java SE – Core Java

(2) Java EE – Advance Java

(3) Java ME → (Micro edition for Mobiles)

Installation of Java → (1) Java SE download from internet

(2) After installation, two folder will be there.

(A) JDK → It includes javac.exe (compiler), java.exe (App^n launcher) etc.

(B) JRE → It contains JVM, Java Package classes (Java library)

JVM → (1) It is platform dependent

(2) It provides a platform-independent way of executing code.

(3) JVM interprets the byte code into the machine code depending upon the underlying operating system and hardware combination.

Java Program → (1) Java is Case-Sensitive language, like C/C++

(2) It is nearly 100% OOP

(3) In Java, it is not possible to make a function which is not a member of any class (as we can do in C++)

For eg →

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
}
```

(1) Access specifier in Java are four types (Public, Private, Protected, default)

(2) outer class can be public or default (can't Private or Protected)

(3) name of class must be same as .java file.

(4) Public → main fn can be accessible from outside

(5) Static → It means main function can be called without creating an object.

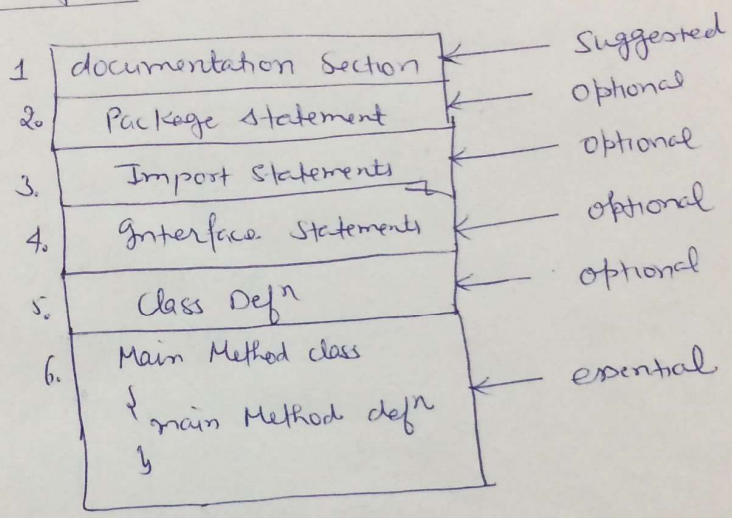
(6) void → no return value.

⑦ System → It is predefined class. if we place dot after class name then its means we calling static member function.

out → static reference variable

System.out.println
↓ ↓ → member function
class reference variable

Structure of Java Program →



1. → This section consist of set of comment lines giving the name of program, author name and other details which programmer would like to explain. //___ , /*___*/ , /**___*/
2. It declares a package name & informs the compiler that the classes defined here belongs to this package.

Package Student;

3. It is similar to #include statement in C.

Import Student.test;

It instructs the interpreter to load the test class contained in the package student.

4. It is used to implement multiple inheritance
5. declaration of class and their definition
6. every java program requires a main method as its starting point this is essential part of java program.