Java is Platform Independent

By Dr. Subrat Kumar Nayak

Associate Professor,

Department of CSE, FET, S’O’A (Deemed to be) University.

General Procedure

- What is a Platform?

Platform is combination of processor and OS(operating system). In general we can say the hardware or software component in which programs run.

- How program get executed?

When you write program in C/C++ and when you compile it, it is directly converted into machine readable language(.exe).

- Two types of codes get generated after compilation need to be focused.

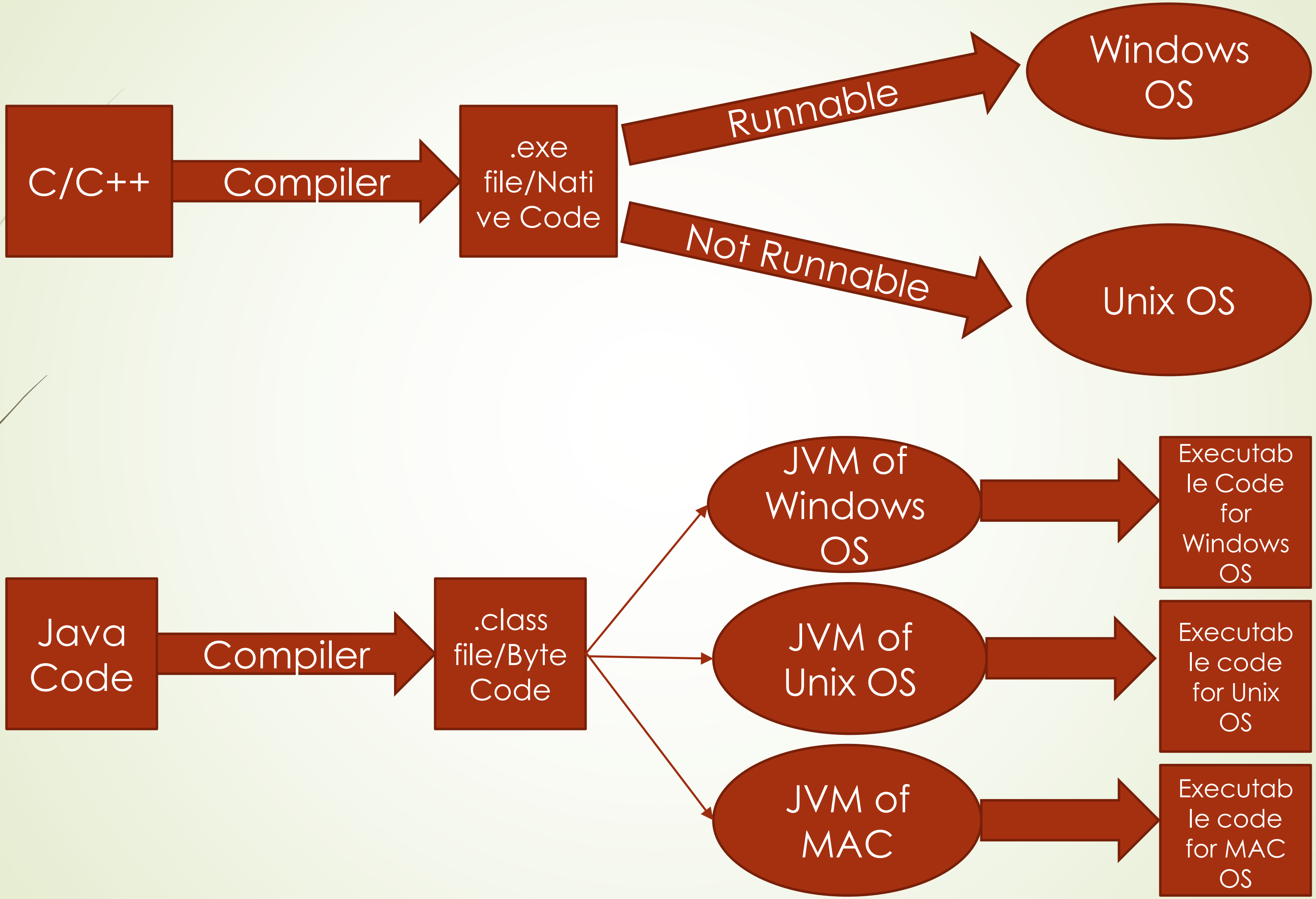
  1. Native code

  Native code is similar to machine code i.e codes that is understood by machine. Native codes are specific to platform i.e, Native code generated by program for Windows OS is different from Native code generated for the same program for Unix OS.

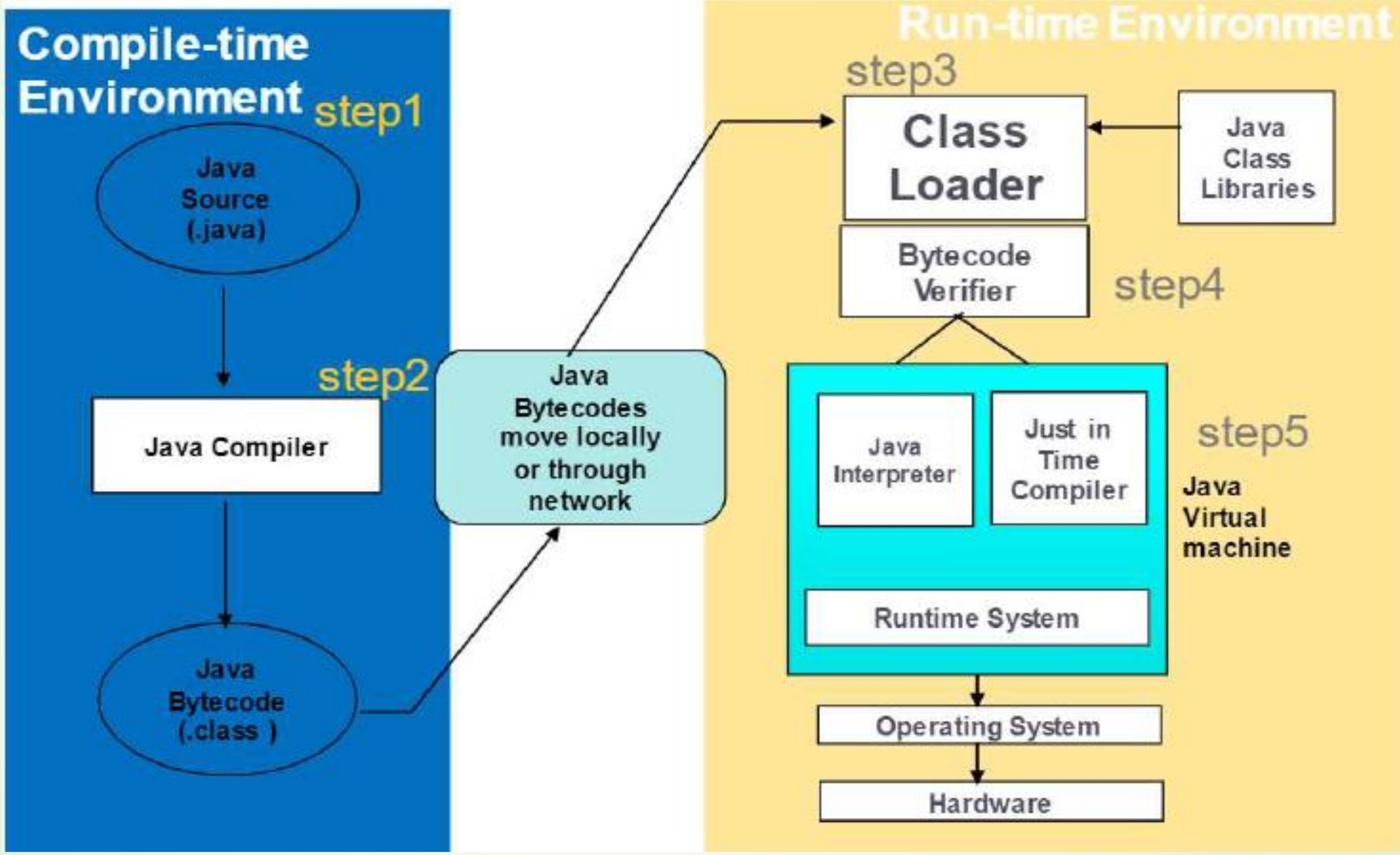
  1. Byte code

  Byte codes are nothing but intermediate codes generated after compilation and it is not the executable code like Native code. The Byte code requires a virtual machine to execute in machine. Byte codes generated by one platform can be executed in another platform also.

- Java compiler converts the source code to .class file (byte code).
- Whether JVM is platform independent?


C/C++
Compiler
.exe file/Native Code
Runnable
Not Runnable
Windows OS
Unix OS
Java Code
Compiler
.class file/Byte Code
JVM of Windows OS
JVM of Unix OS
JVM of MAC
Executable Code for Windows OS
Executable code for Unix OS
Executable code for MAC OS

Java Architecture



Java main()

- `public static void main(String[] args)` is the most important java method.

`public`: This is access modifier of `main()`. It has to be public so that java runtime can access this method.

`static`: `main()` function need to be static so that JVM can load the class to memory and call the main method.

`void`: Java `main()` does not return anything.

`String[] args`: Java `main()` accepts a single argument of type String array. This is also called as java command line arguments.

End of Class