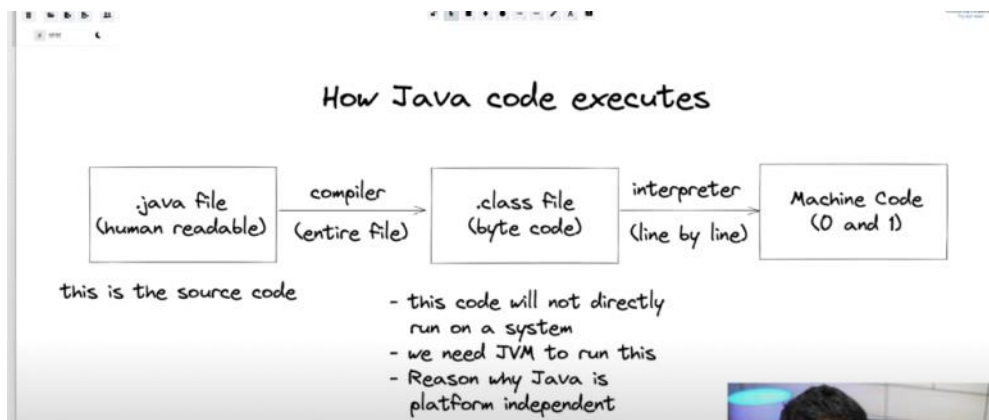


How Java Code Executes

1. All java code will be written in ____
2. Role of compiler?
3. What does compiler do in java?
4. ____ is an intermediate language of java.
5. Why do we need byte code?
6. We need ____ to run the byte code to convert it into machine code.
7. In C++ compiler converts the file directly to ____ code
Interpreter ____
8. Why java is platform independent?



Q → Architecture of Java

10 → JDK?

11 → JRE

1 → .java

2 → * Take the file and convert it into machine code

3 → * In Java compiler converts file into byte code (.class) file not in machine code

4 → * byte code > Intermediate language of Java

5 → * The java code cannot directly run on the system. we need JVM to run this once this JVM runs this code it will be converted into machine code that computer understands.

* How we can imagine is different language is being formed in the form of byte code in the middle stage & that is interpreted line by line using something called JVM & that is how it is converted into machine code. This is the reason java is platform independent.

In C++ this middle part is skipped

6 → * JVM

7 → * executable code

* will compile the file line by line.

8 → *

* It means byte code can run on all operating system. means platform independent

If I had written a .java file in my windows PC it will run on any OS linux, windows, mac etc

Executable code:- code that can run

More about platform independence

- It means that byte code can run on all operating systems.
- We need to convert source code to machine code so computer can understand
- Compiler helps in doing this by turning it into executable code
- this executable code is a set of instructions for the computer
- After compiling C/C++ code we get .exe file which is platform dependent
- In Java we get bytecode, JVM converts this to machine code
- Java is platform-independent but JVM is platform dependent

* This is the reason why Java is platform independent since it's not converting the source code directly to machine code. It's converting into byte code & this byte code can run on any machine

* Windows, macOS, Ubuntu → JVM is platform dependent → JVM could be Windows or Linux

* Java is platform independent | JVM is platform dependent

* Java is platform independent because it converts the source code not directly to the executable [machine] code it converts to byte code. the byte code can run on any machine

In C++ the exe file we get the executable code is nothing but the set of instruction given to the computer & this depends on the architecture of computer, the executable code will have set of instruction depending upon the platform. but byte code is not machine code you can run it on any machine

* How do we convert this byte code into machine code & run this program?

We use JVM in Java we get the byte code & JVM. let's say it is a program someone has written that converts,

it into machine code this machine code will run on the system then. Now if I have byte code I will share it on windows, Linux etc all you need in your system is JVM. now Java is platform independent JVM is platform dependent. obviously every OS JVM may have its own set of instructions to convert byte code into machine code for that particular OS

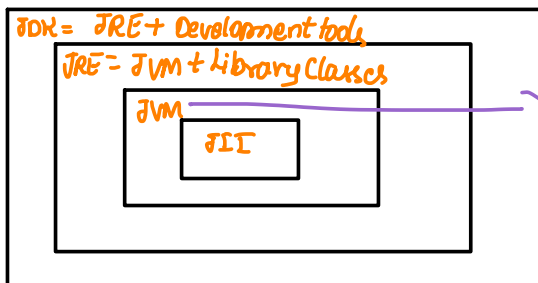
9 → Architecture of Java

JDK → Java development kit

JRE → Java Runtime Environment

JVM → Java Virtual machine

JIT → Just in Time



JVM needs JRE to provide Libraries.
JRE is a box
JVM is dependent

10 → JDK → Provides Environment to develop and run the Java Program

→ It is a package that includes

- * development tools - to provide an environment to develop your program
- * JRE - to execute your program
- * a compiler - Javac (Java code → byte code by compile)
- * archiver - jar
- * docs generator javadoc
- * interpreter/loader

* If you want to develop Java program JDK is required.

* library is there

11 → JRE

JRE → You can Run Java Program ✓

You can Create Java Program X

→ It is an installation package that provides environment to only run the program

→ It consists of

- * Deployment technologies
- * User interface toolkits
- * Integration Libraries
- * Base Libraries
- * JVM

→ After we get the class file the next things happen at runtime

1. Class loader loads all classes needed to execute the program
2. JVM sends code to Byte code verifier to check the format of code

