Types of Programming languages

Procedural Language

Here a proper procedure is followed while writing code. For example if we had to add two numbers, first first number is taken, than second number is taken and those two will be added, it’s not like first number is taken then addition operation occurs and then only second number is taken

Functional Language

Here everything is a function and data moves along these functions as one function calls the another.

First class functions : If we can assign the functions just like we do for variables than the function are called the first class functions.

Object Oriented language

Here we have objects / classes, an object itself is a complete item. An object can contain data, functions, variables everything needed for it inside it and these objects can interact with each other

Commonly we say an object contain methods and properties inside it, methods is a fancy name for function inside the object or class. And properties are all the variables inside the class / object

Static vs Dynamic Languages

Static

checks the data type at the compile time, so errors will the shown at the compile time

Dynamic

checks the data type at the runtime

int num = 100;

num variable of int data type is created

Here num is statically typed

java , C++ are all statically typed language

for example

if we do int num = “Amrit”; error is shown during compile time which shows data type is checked during the compilation

Dynamic language

num = 100;

num = “Amrit”;

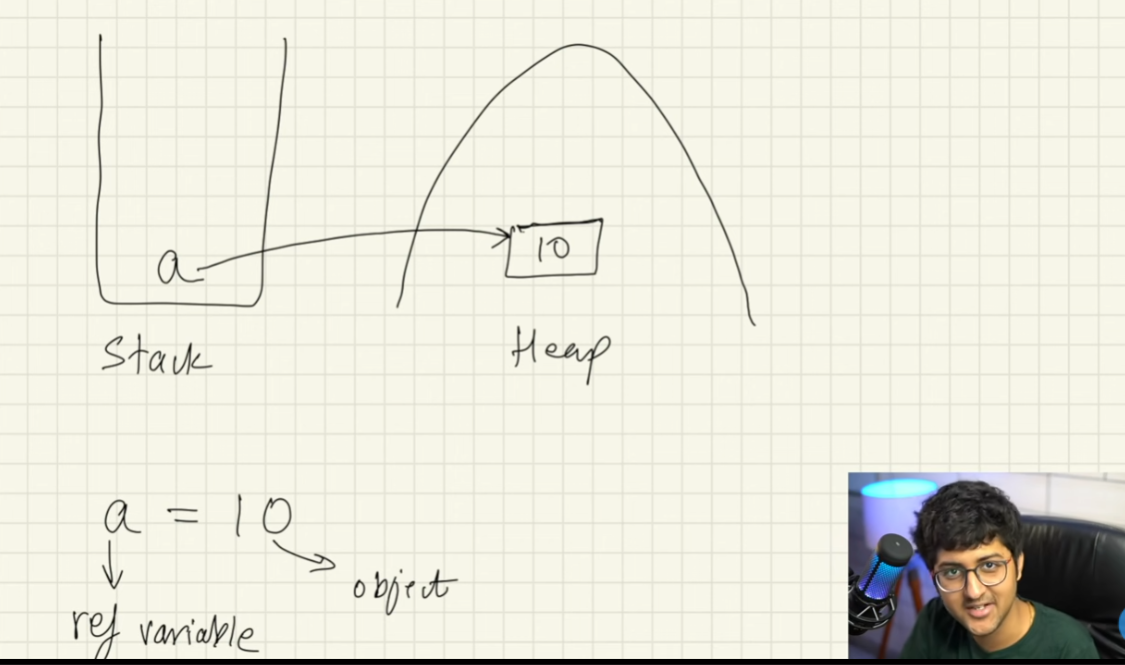
here we will not get any error

here num is actually pointing to a memory segment

first it was pointing to memory segment storing 100

second it is pointing to the memory segment storing “Amrit”

Stack and Heap memory



so let say

a = 100;

name = “Amrit”

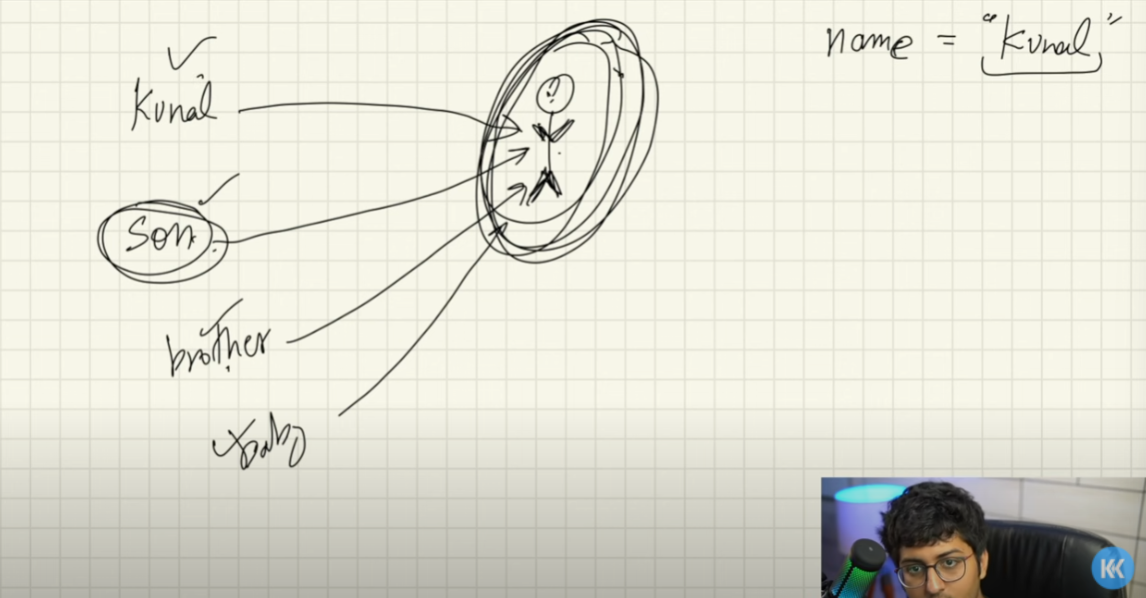
so variables like a, name are stored in stack memory and their values or objects are stored in heap memory

And the variables stored in stack points to the objects in heap memory

My name is Amrit, my mom calls me “son”, my vai calls me “brother”, my girlfriend calls me “baby”

So “Amrit”,”son”,”brother”,”baby” are all refering to the same person

so if “Amrit” gets hair cut than “son”, “brother”, “baby” all gets hair cut



so if many variables are pointing to the same value in the heap memory than changing one will change the value in the heap

So every other variables pointing to the heap will also point to the changed / updated values



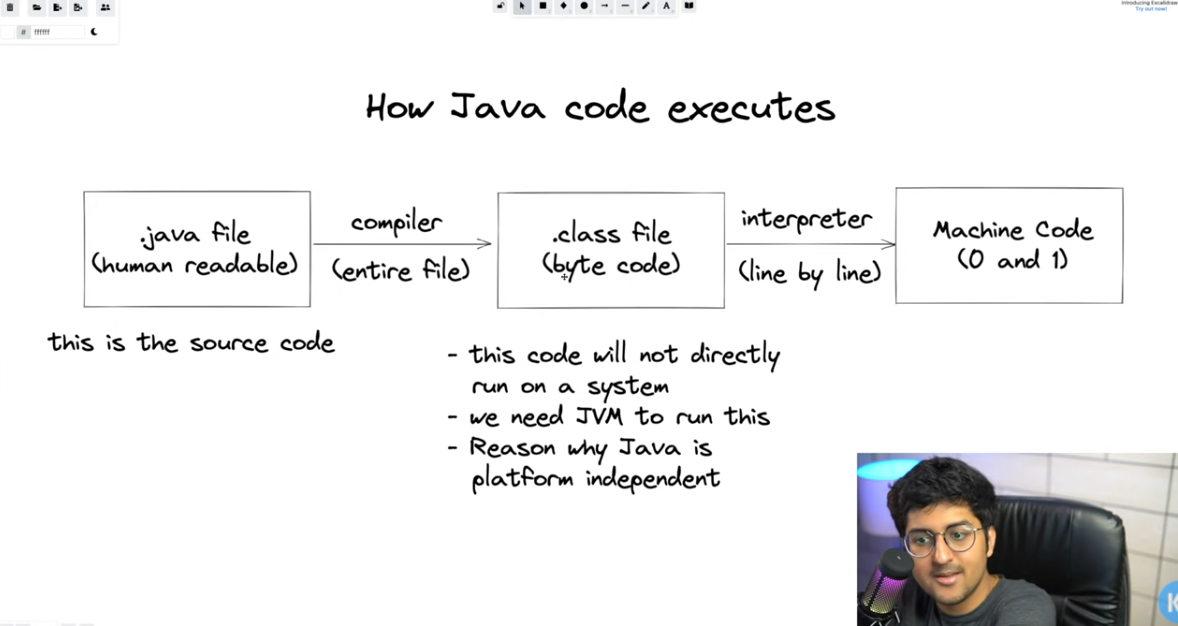
here when we have an object with no any reference variable pointing to it than this object is removed from the memory during the garbage collection

Execution of Java Code

Java code executes differently from the C++ code

In C++ the compiler converts the .cpp file in to the machine redable form

But in java things happen little bit differently



here first the java compiler converts the source code in to something known as byte code first

the byte code has .class extension

this byte code will not run directly in the system

we need JVM ( Java Virtual Machine ) to run this

The byte code is being interpreted line by line using something called Java Virtual Machine

in C++ we don’t have any byte code, the compiler will directly convert the source code in to the machine code

Java is Platform independent unlike C++

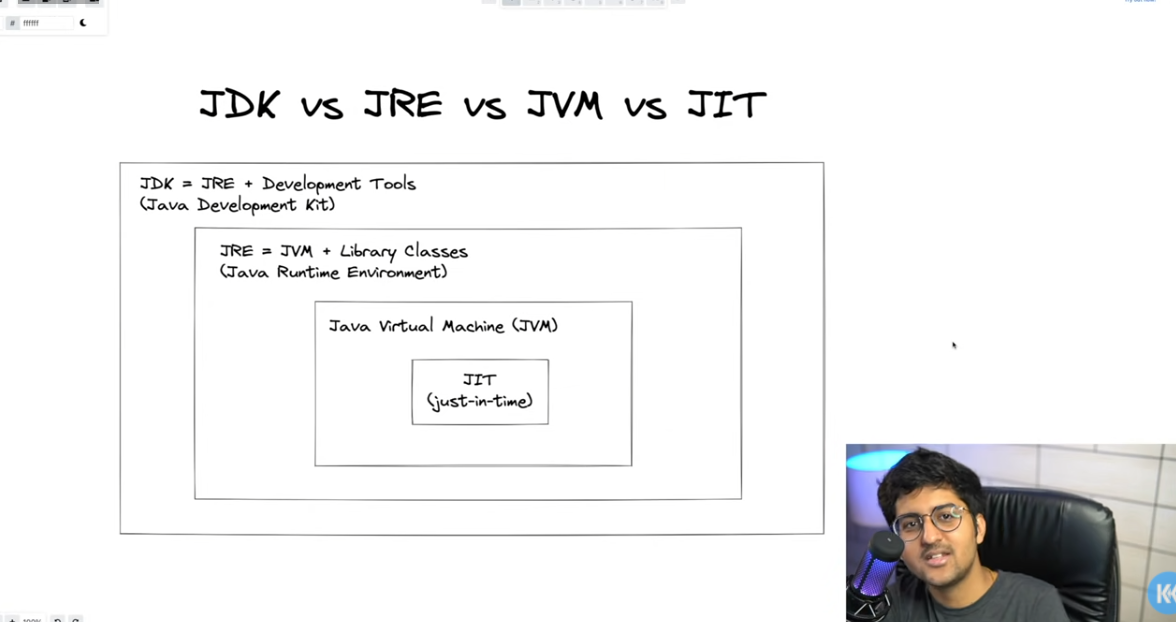
The intermediate byte code obtained in let say mac can run later on in windows, linux or any other platform

We can compile and obtain byte code in one platform and actually run that byte code in any OS we want, we won’t have any problem

But

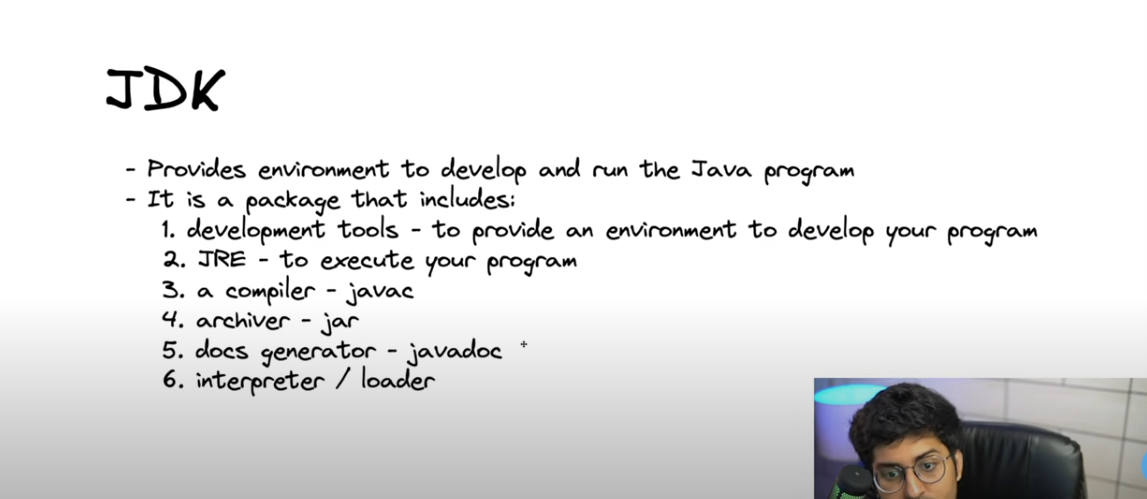
When we compile C or C++ source code we get .exe file which is platform dependent

JVM is platform dependent we will have JVM for different platform, and that JVM can convert the byte code in to machine code

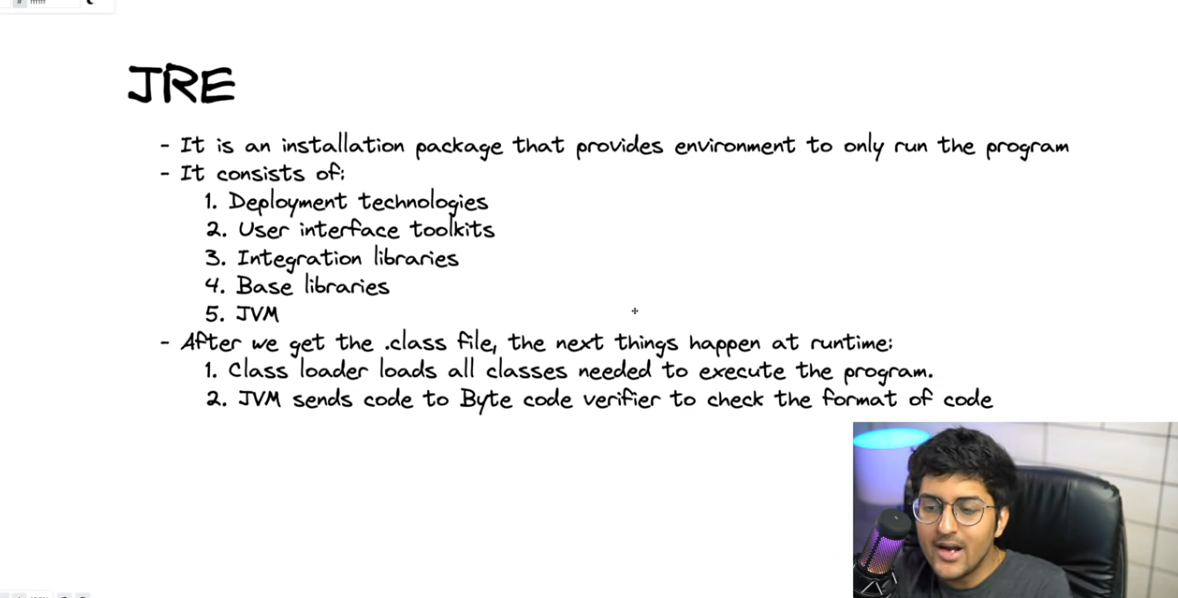


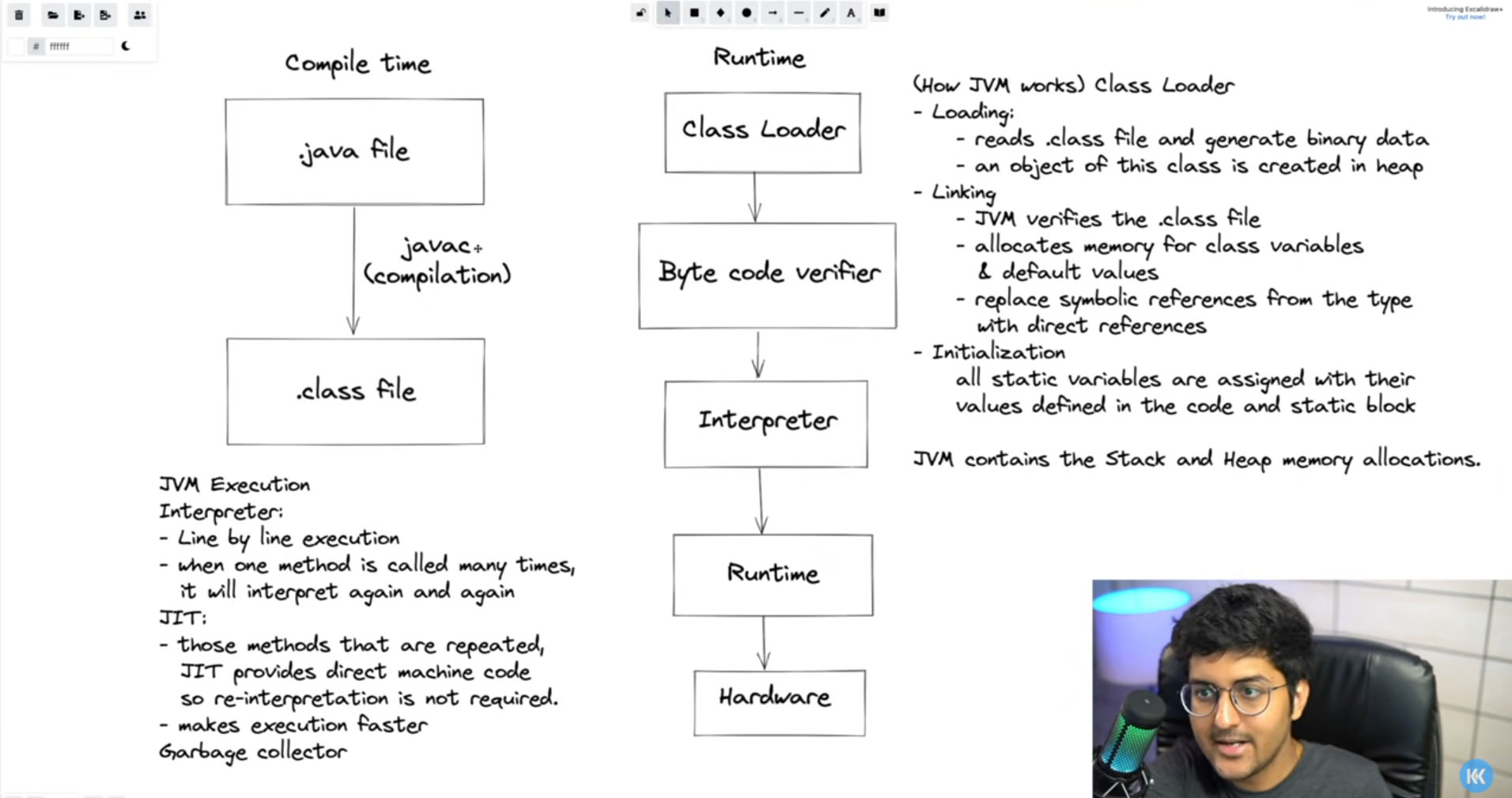
JDK ( Java Development Kit )

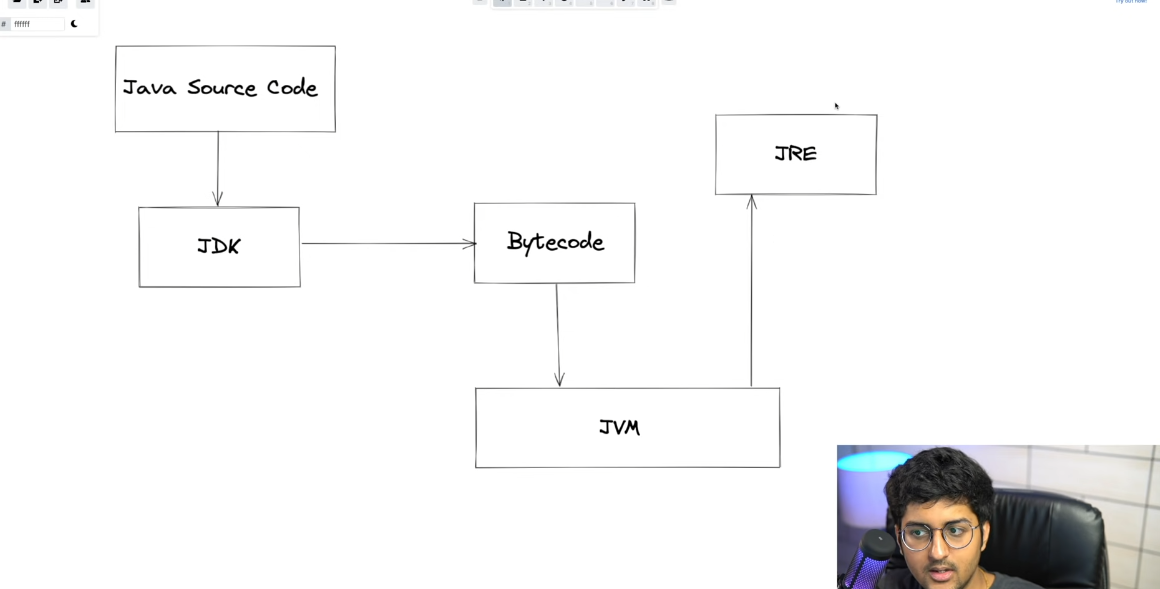
It consist of JRE ( Java Runtime Environment ) and some development tools



this compiler javac will convert the .java file in to the .class file ( byte code )







The source code that we write goes to JDK, and there is a compiler in JDK, JDK compiles it in to bytecode

The bytecode is interpreted with JVM and finally program runs