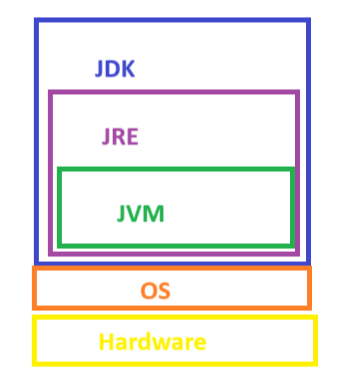
**8. 12th Oct Notes:**

1. **JVM is the one who actually executes Java programs, but JVM understands .class files(byte code) only to execute.**
2. **Java compiler compiles and converts .java files into .class files and which are sent to JVM to execute.**
3. **Now JVM has multiple class files which were received from Java Compiler, but from where(which class file) JVM needs to start the execution? JVM starts the execution with the file which has the main method in it.**
4. **So the main() method should be there in at least one of the classes to start the execution by the JVM.**
5. **We have a main method to start the execution but it was there inside in a class which our JVM does not know, meaning the main method was hidden in a class which JVM could not see.**
6. **To make the main class visible to JVM, we should specify the method as “public”.**
7. **Now the main method is public and JVM sees it but the access for the method to JVM we have not given. For that we use “static” in method declaration.**
8. **Static allows access to JVM for the main method.**
9. **Normally in JAVA os a OOPS language, to access a class an object for that class must be created and with the help of that object we can access the class and methods inside it.**
10. **Static is the one which allows JVM to access a class and methods inside it without creation of objects for the class.**
11. **Now everything is ready for JVM to execute the program, but one more thing is there which provides an environment( libraries, dependencies…..) for JVM to run the Java program and that is JRE.**

**JDK(java dev kit) = (Java Compiler + JRE + JVM)**

**JRE(java runtime env) = (JVM + Libraries)**

**JVM(java virtual machine) = the actual executor**

****

**hello.java→JavaCompiler→.class files(bytecode)→JVM→output**

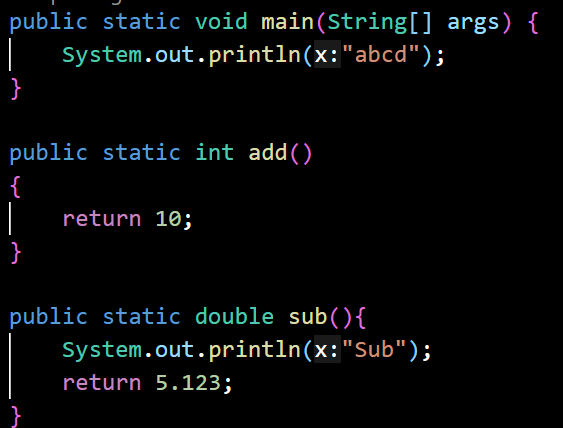
1. **Java is said to be a Platform independent language as the bytecode(.class files) are independent of OS/platform they just need a JVM to execute irrespective of whichever platform it is.**

**Public static void main(String args[])**

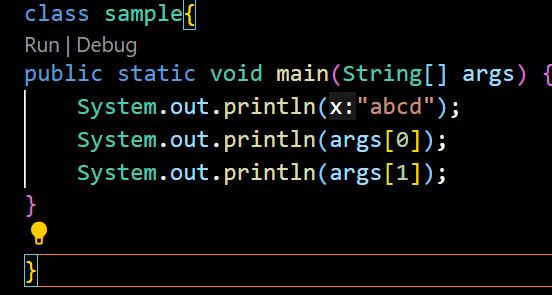
**{**

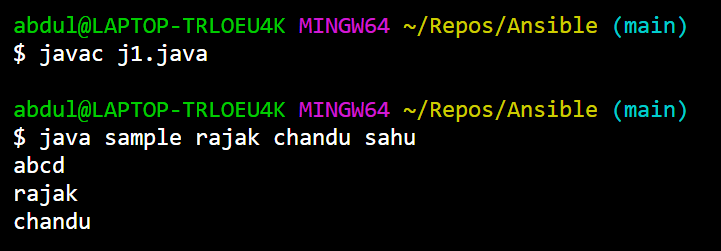
**}**

1. **We are done with public,static,main. Void is for type the data which the method is returning. If nothing is returning then void.**

****

1. **String args[] → string of elements in an array. It is used to receive the cmd line arguments. Whatever we pass from the cmd line most of the languages treats them as Strings only. That’s the reason we mentioned type of array as string**

****

****

**Note: whenever you are running a java program make sure to compile it first to get the latest class file otherwise it will execute the old class file if present.**