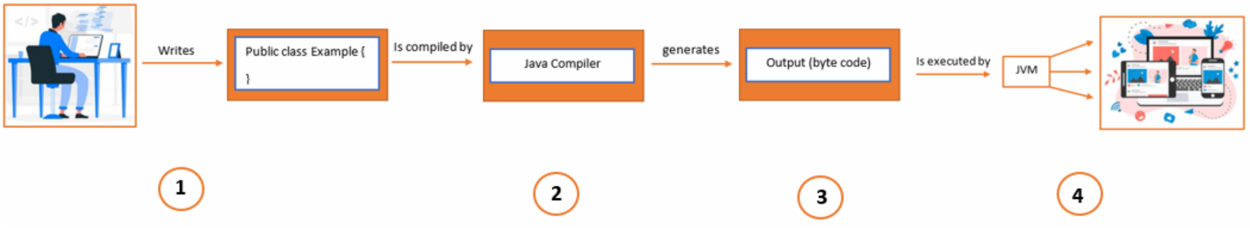
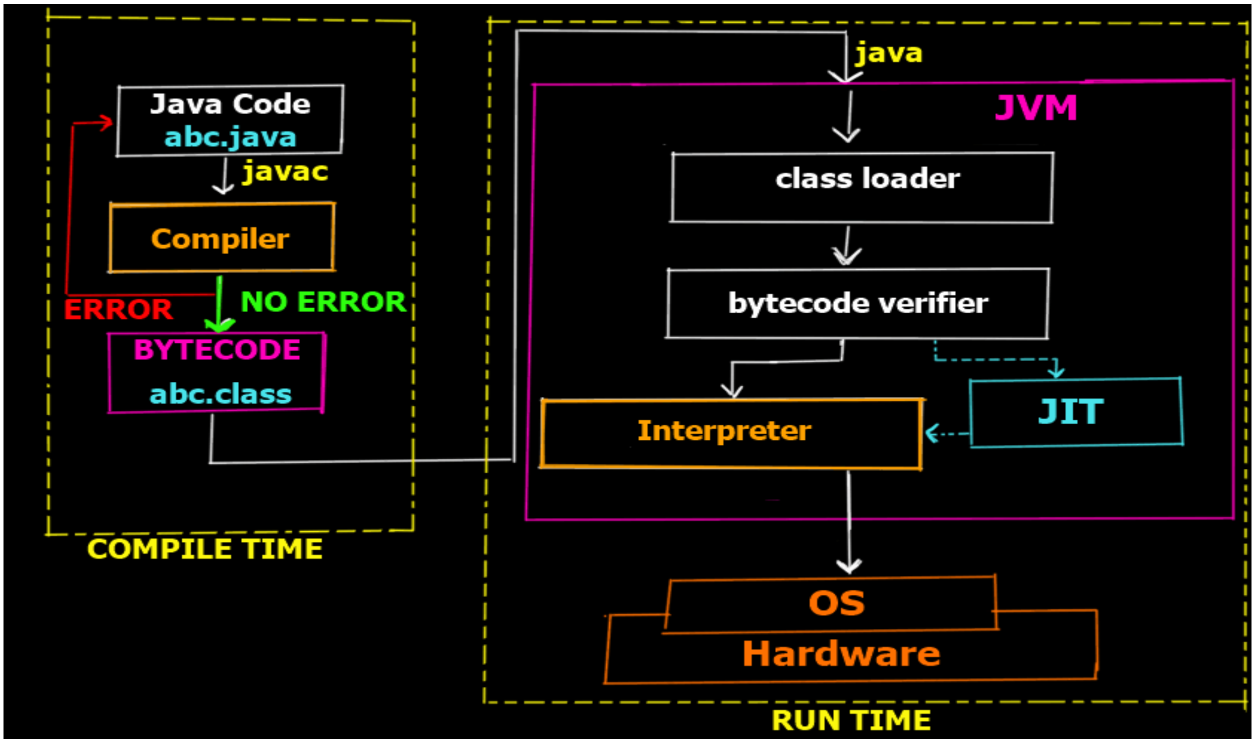
Java Architecture

Java Architecture is a collection of components, i.e., **JVM, JRE,** and **JDK**. **It** integrates the process of interpretation and compilation.

Java Architecture can be explained by using the following steps:

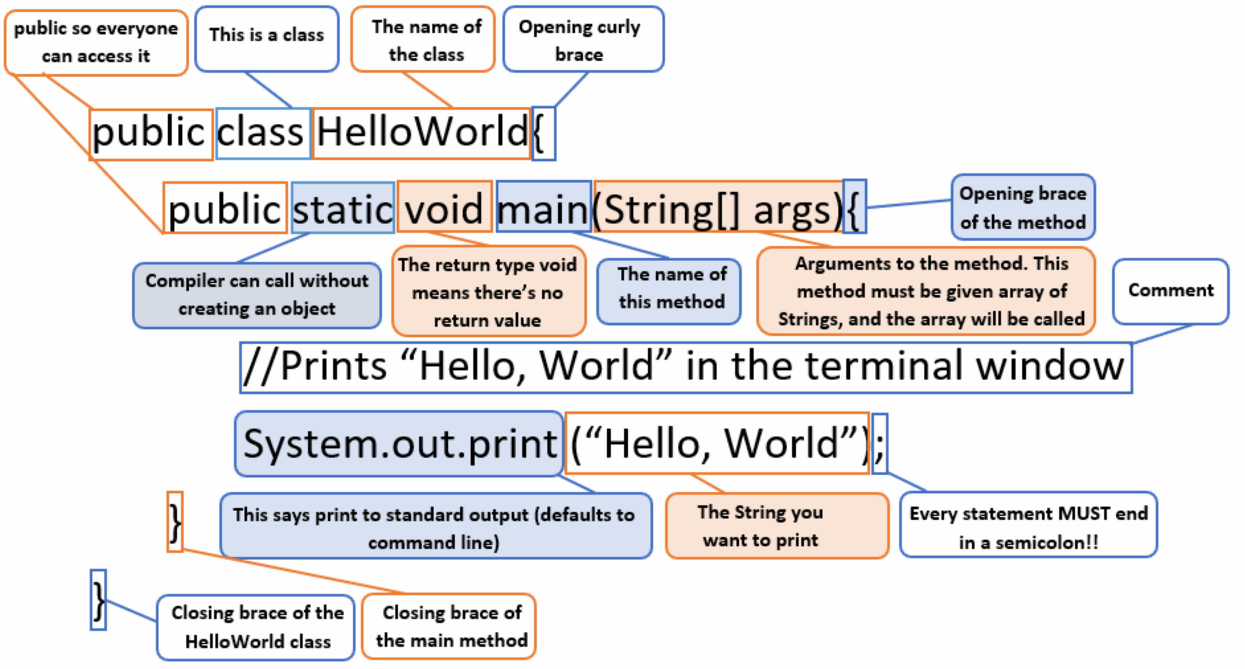
- There is a process of compilation and interpretation in Java.
- Java compiler converts the Java code into byte code.
- After that, the JVM converts the byte code into machine code.
- The machine code is then executed by the machine.





- **Step 1:** Writing java code (Here abc.java).
- **Step 2:** Once you have written the code you save it and click Run .This invokes the Java Compiler. The compiler checks the code for syntax errors and any other compile time errors and if no error is found the compiler converts the java code into an intermediate code (**abc.class** file) known as **bytecode .**This **intermediate code is platform independent** (you can take this bytecode from a machine running windows and use it in any other machine running Linux or MacOS etc.). Also this bytecode is an intermediate code; hence it is only understandable by the **JVM** and not the user or even the hardware /OS layer.
- **Step 3:** This is the start of the Run Time phase, where the bytecode is loaded into the JVM by the **class loader** (another inbuilt program inside the JVM).
- **Step 4:** Now the **bytecode verifier** (an inbuilt program inside the JVM) checks the bytecode for its integrity and if not issues are found passes it to the interpreter.
- **Step 5:** Since java is both compiled and interpreted language, now the interpreter inside the JVM converts each line of the bytecode into executable machine code and passed it to the OS/Hardware i.e. the CPU to execute.

Anatomy Of a Class



```
public class HelloWorld{
    public static void main(String[] args){
        //Prints “Hello, World” in the terminal window
        System.out.print(“Hello, World”);
    }
}
```

public so everyone can access it
This is a class
The name of the class
Opening curly brace
Opening brace of the method
Compiler can call without creating an object
The return type void means there’s no return value
The name of this method
Arguments to the method. This method must be given array of Strings, and the array will be called
Comment
This says print to standard output (defaults to command line)
The String you want to print
Every statement MUST end in a semicolon!!
Closing brace of the HelloWorld class
Closing brace of the main method