FIRST PROGRAM IN JAVA

CODE FOR CODERS

SETTING UP JAVA RUNTIME ENVIRONMENT ON YOUR LOCAL MACHINE

Before we can write and execute any java program, we need to set up java runtime environment on our local machine. For that, we have to follow the following steps –

1- Download JDK -

To execute java programs, you need a compiler and interpreter, and they are provided by a software named **Java Development Kit (JDK)**.
SO first of all you need to download the latest version of JDK from Oracle website and install it. You can download it from the following link -
https://www.oracle.com/technetwork/java/javase/downloads/index.html

2- Set path for JDK -

After installing the jdk you need to set the path for java. To set path follow the following steps -

I. Go to ***MyComputer*** and then open ***Local Disk (C)***. There will be a folder named **Program Files.**

II. Open the folder and you will find another folder named **Java.**

III. Open that folder and you will find a folder named **jdk1.8.0_102** (the number following the jdk may vary depending upon the version of jdk you installed).

IV. Open that folder and you will find a folder named **bin,** after that just copy the complete address shown in address bar (that may look like this -

C:\Program Files\Java\jdk1.8.0_102\bin)

I. Now right click on ***MyComputer*** and go to ***Properties*** there you will find a option named ***Advanced system setting***, click on that.

II. A windows will open, find a option named ***Environment Variables*** and click on that.

III. A a new window will open click there on ***New*** option. It will open a new window named ***New User Variable***.

IV. It contains two option ***variable name*** and ***variable path***. Enter variable name as **'path'** and in variable value paste the path that you have copied previously.

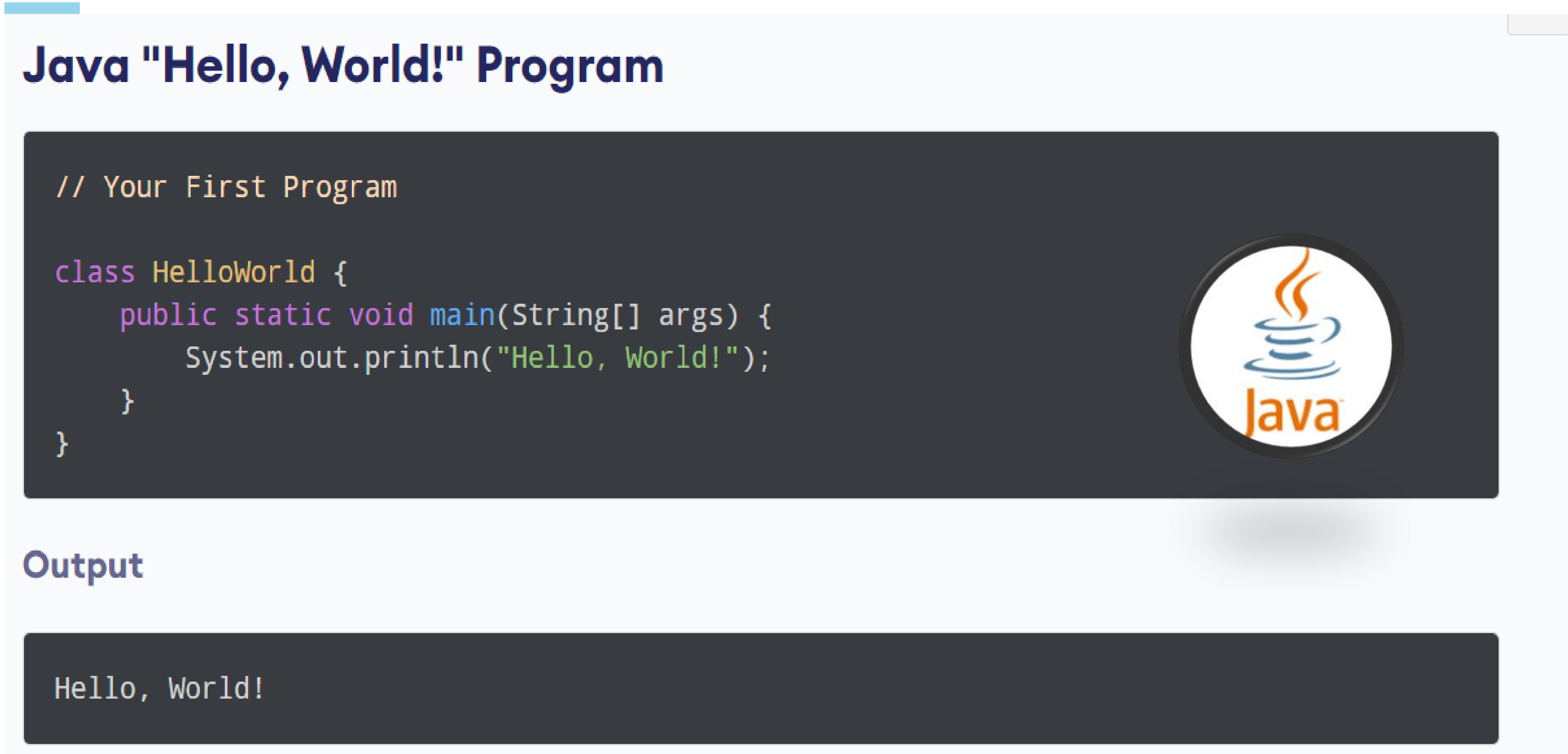
BASIC JAVA SYNTAX

A Java program can be considered as a collection of objects that communicate by invoking each other's methods.

Let's take a quick look at what class, object, methods are and what they mean.

CODE FOR CODERS

JAVA PROGRAM



Java "Hello, World!" Program

```
// Your First Program

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

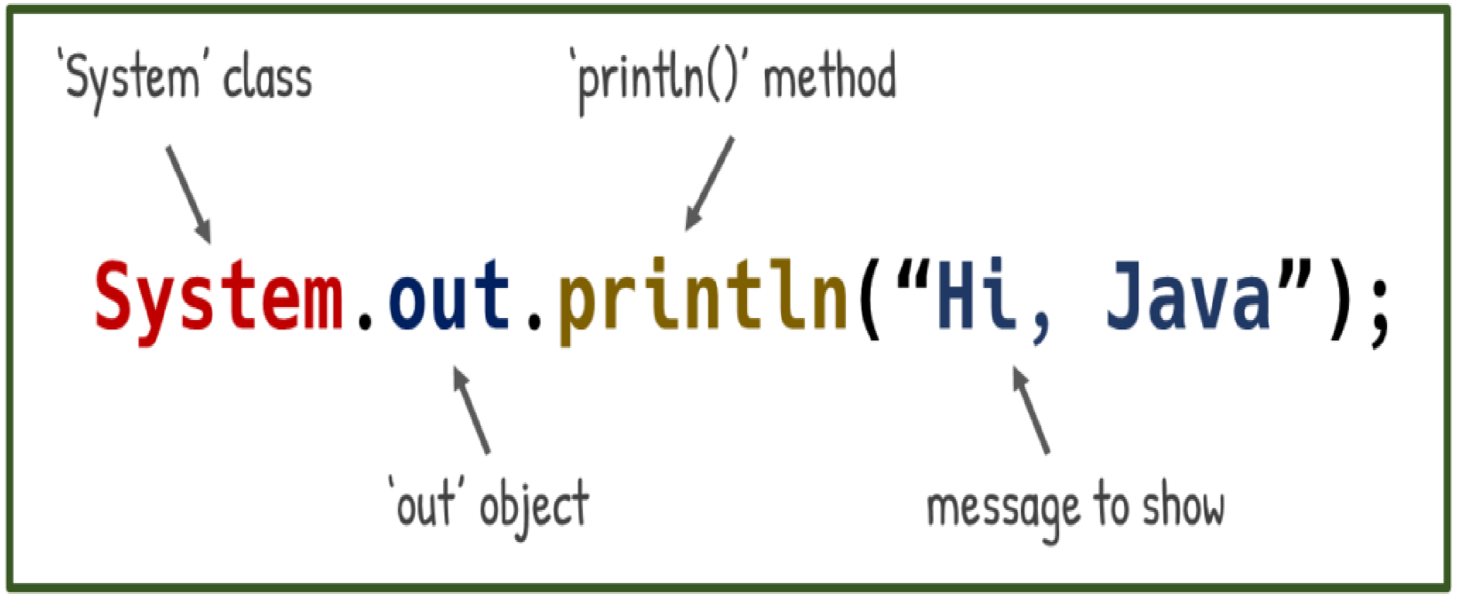
Output

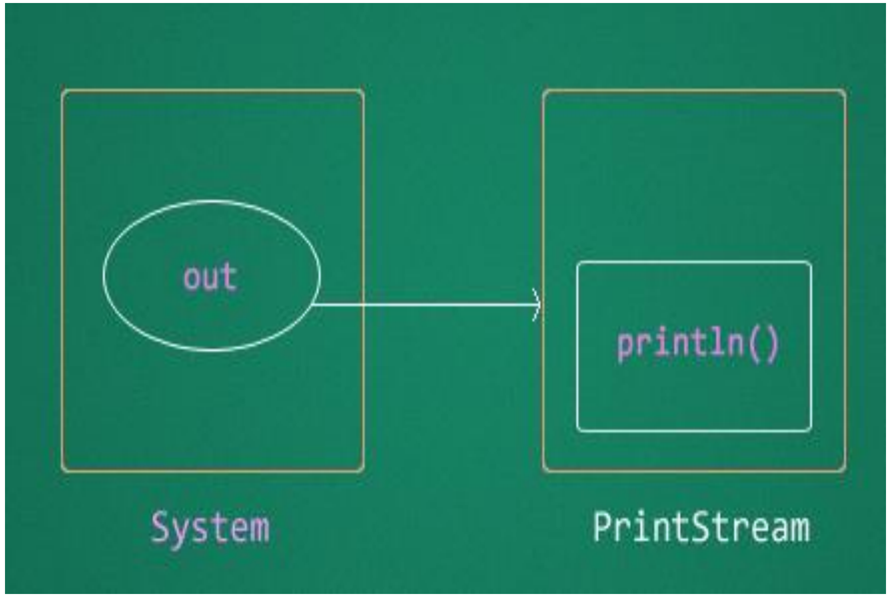
```
Hello, World!
```

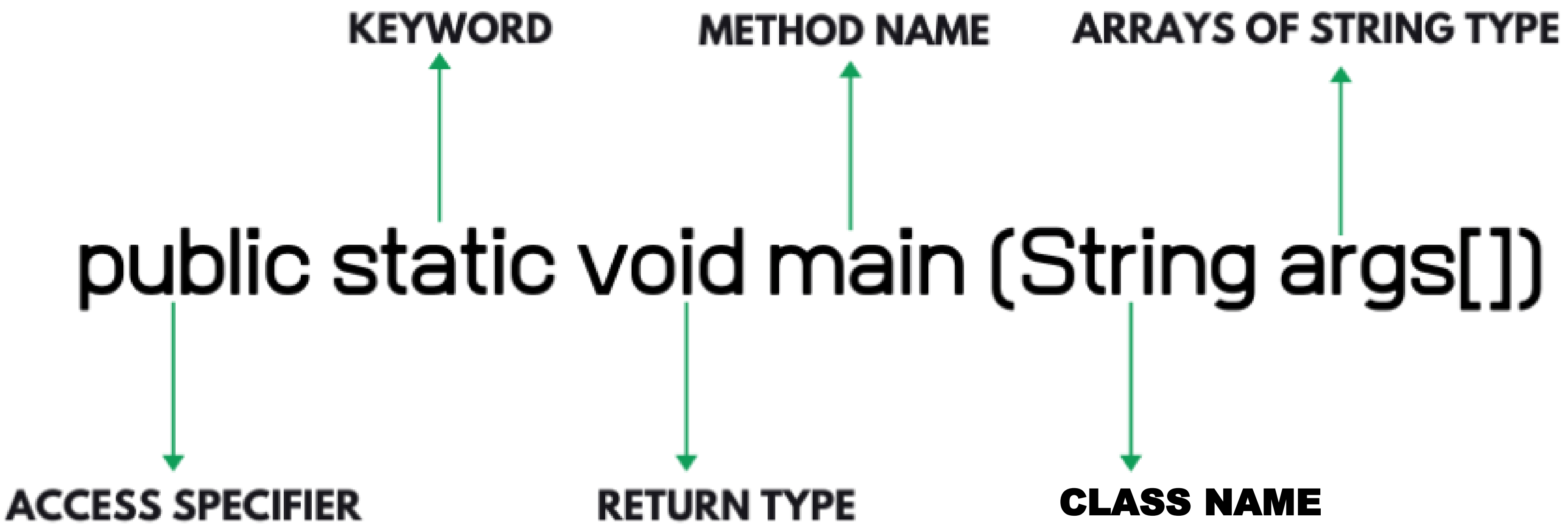
- **Class :-** A class can be defined as a template/ blueprint that describes the behavior/state that the object of its type supports.
- **Class Name :-** Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive.
- **Public :-** The public keyword is an access modifier used for classes, attributes, methods and constructors, making them accessible by any other class.
- **static :-** The static keyword is a non-access modifier used for methods and attributes. Static methods/attributes can be accessed without creating an object of a class.
- **void** - The void keyword specifies that a method should not have a return value.
- **main :-** main(): It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method.
- **String args[]:** The main() method also accepts some data from the user.

- **System.out.println() :-**Java System.out.println() is used to print an argument that is passed to it The statement can be broken into 3 parts which can be understood separately as

1. **System :** It is a final class defined in the java.lang package.
2. **Out :** is a static member field of System class and is of type PrintStream.
3. **Println :** is a method of PrintStream class.This is an upgraded version of print(). It prints any argument passed to it and adds a new line to the output. We can assume that System.out represents the Standard Output Stream.



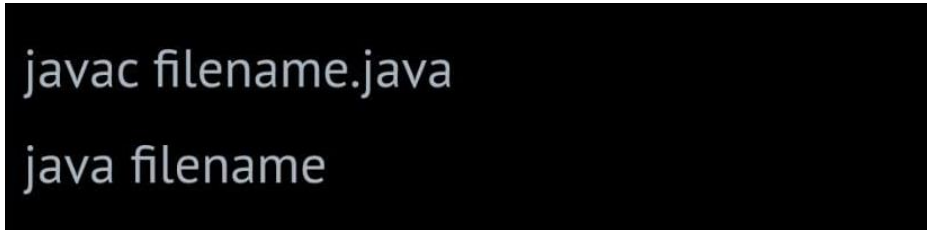




```
public static void main (String args[])
```

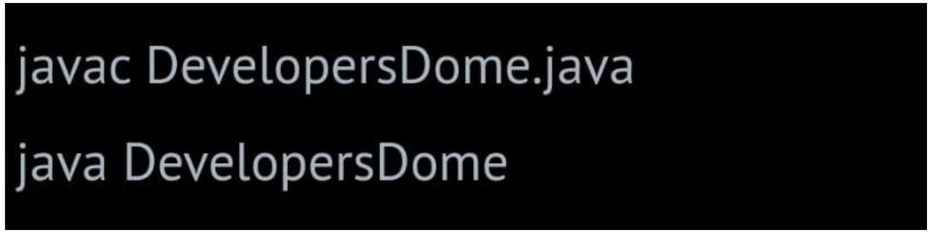
Note: The Keyword Public and Static position can be changed but the Void & Main should be declared like void main. If you have mentioned the main void then you will get errors.

To run a java program and compile in a console- Run this Command:



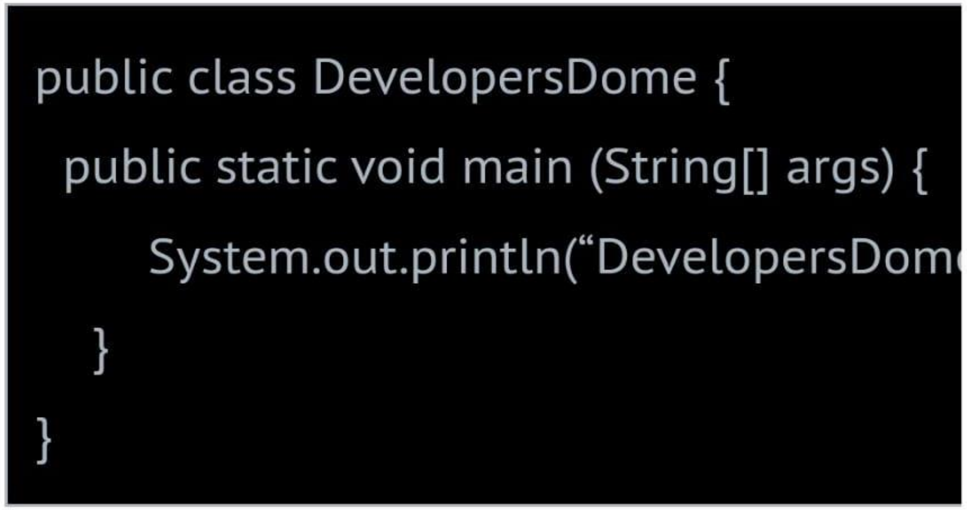
```
javac filename.java

java filename
```

Example to run :



```
javac DevelopersDome.java

java DevelopersDome
```

File Name → Developers.java

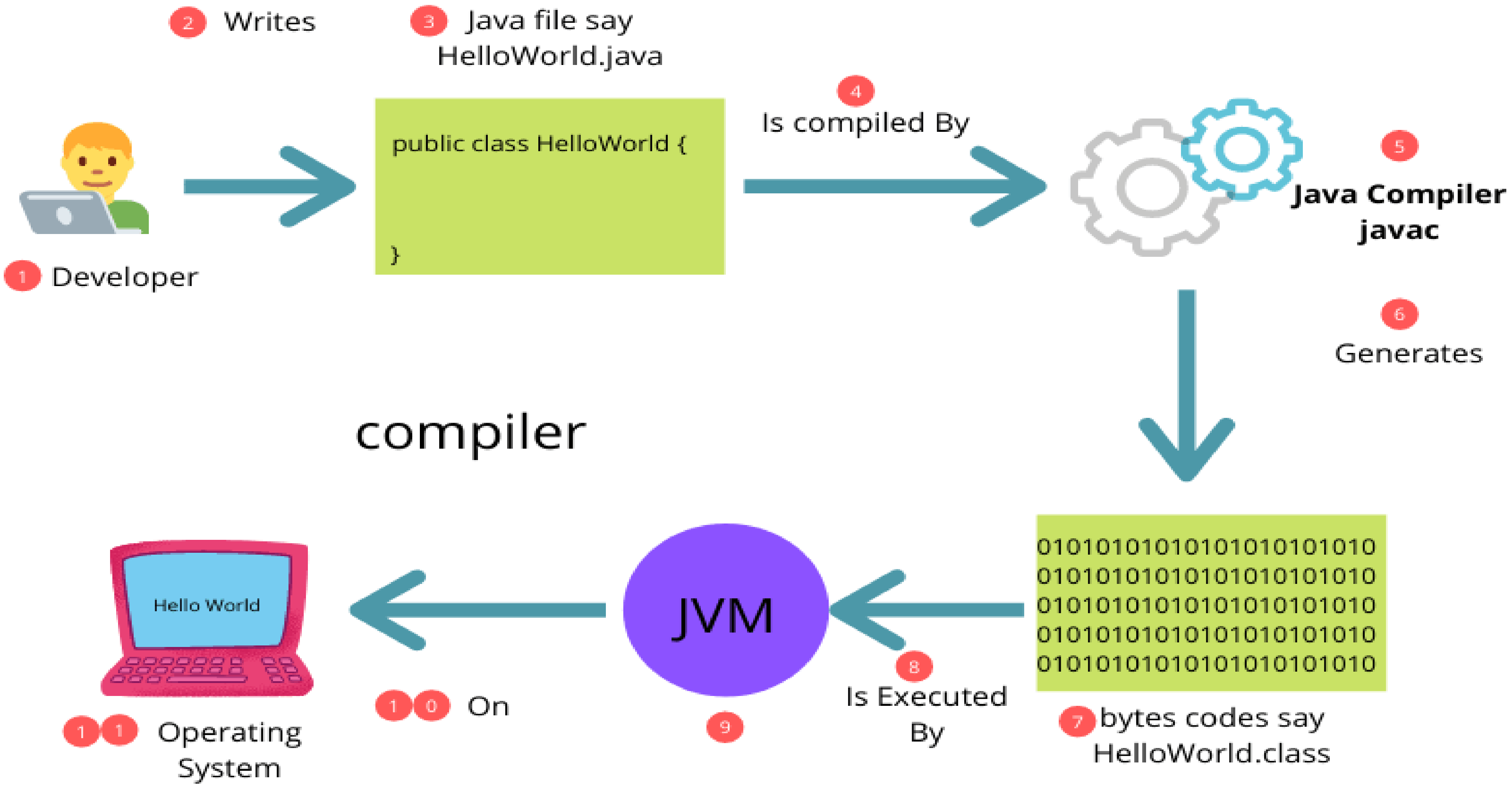


```
public class DevelopersDome {

  public static void main (String[] args) {

        System.out.println(“DevelopersDom

    }

}
```

Output

```
DevelopersDome
```

HOW JAVA WORKS



1 Developer

2 Writes

3 Java file say HelloWorld.java

```
public class HelloWorld {

}
```

4 Is compiled By

5 **Java Compiler javac**

6 Generates

7 bytes codes say HelloWorld.class

```
0101010101010101010101010
0101010101010101010101010
0101010101010101010101010
0101010101010101010101010
0101010101010101010101010
```

8 Is Executed By

9 JVM

10 On

11 Operating System

compiler

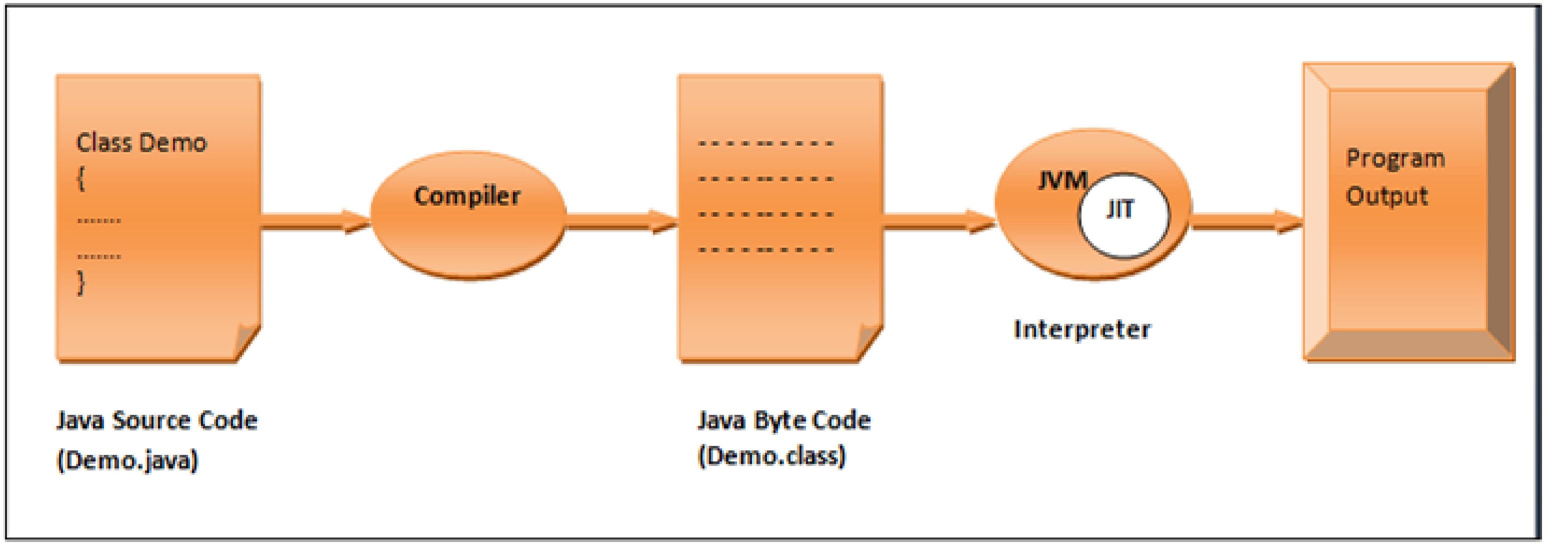


fig: java compilation and interpretation process.

- Java is compiled and interpreted language.

- First of all compiler converts the java source code into a form called Byte Code. This is a file with .class extension and this Byte Code is called platform independent.

- This Byte Code is interpreted by Java Virtual Machine (JVM) into executable code. Because JVM is available for almost every available device that's why Java is called platform independent language.

- And this Byte Code also provides the security as well as portability to java language.
- As java is compiled and interpreted language so one may think that its execution speed is slow than other compiled languages like C or C++. But there is not much difference because Byte Code is highly optimized and JVM can execute it much faster than you think.
- Also JVM contain a JIT (Just In Time) compiler that in place of compiling the whole program, compiles only that part of the program that is needed at that time.

CODE FOR CODERS

THANKS FOR WATCHING

DON'T FORGET LIKE, COMMENT AND SUBSCRIBE

COODE FOR CODERS