Java Hello World Program

Java is one of the most popular and widely used programming languages and platforms. Java is fast, reliable, and secure. Java is used in every nook and corner from desktop to web applications, scientific supercomputers to gaming consoles, cell phones to the Internet. In this article, we will learn how to write a simple Java Program.

Steps to Implement Java Program

Implementation of a Java application program involves the following step. They include:

1. Creating the program
2. Compiling the program
3. Running the program

1. Creating Programs in Java

We can create a program using Text Editor (Notepad) or IDE (NetBeans)

```
class Test
{
    public static void main(String []args)
    {
        System.out.println("My First Java Program.");
    }
};
```

File Save: d:\Test.java

2. Compiling the Program in Java

To compile the program, we must run the Java compiler (javac), with the name of the source file on the “command prompt” like as follows

If everything is OK, the “javac” compiler creates a file called “Test.class” containing the byte code of the program.

3. Running the Program in Java

We need to use the Java Interpreter to run a program. Java is easy to learn, and its syntax is simple and easy to understand. It is based on C++ (so easier for programmers who know C++).

The process of Java programming can be simplified in three steps:

- Create the program by typing it into a text editor and saving it to a file – HelloWorld.java.

- Compile it by typing “javac HelloWorld.java” in the terminal window.
- Execute (or run) it by typing “java HelloWorld” in the terminal window.

The below-given program is the most simple program of Java printing “Hello World” to the screen. Let us try to understand every bit of code step by step.

- Java

```
// This is a simple Java program.

// FileName : "HelloWorld.java".



class HelloWorld {

    // Your program begins with a call to main().

    // Prints "Hello, World" to the terminal window.

    public static void main(String args[])

    {

        System.out.println("Hello, World");

    }

}
```

Output

```
Hello, World
```

The complexity of the above method

Time Complexity: *O(1)*
Space Complexity: *O(1)*

The “Hello World!” program consists of three primary components: the HelloWorld class definition, the main method, and source code comments. The following explanation will provide you with a basic understanding of the code:

1. Class Definition

This line uses the keyword **class** to declare that a new class is being defined.

```
class HelloWorld {

    //

    //Statements

}
```

2. HelloWorld

It is an identifier that is the name of the class. The entire class definition, including all of its members, will be between the opening curly brace “**{**” and the closing curly brace “**}**“.

3. main Method

In the Java programming language, every application must contain a main method. The main function(method) is the entry point of your Java application, and it's mandatory in a Java program. whose signature in Java is:

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

Explanation of the above syntax

- ***public**: So that JVM can execute the method from anywhere.*
- ***static**: The main method is to be called without an object. The modifiers are public and static can be written in either order.*
- ***void**: The main method doesn't return anything.*
- ***main()**: Name configured in the JVM. The main method must be inside the class definition. The compiler executes the codes starting always from the main function.*
- ***String[]**: The main method accepts a single argument, i.e., an array of elements of type String.*

Like in C/C++, the main method is the entry point for your application and will subsequently invoke all the other methods required by your program.

The next line of code is shown here. Notice that it occurs inside the main() method.

```
System.out.println("Hello, World");
```

This line outputs the string "Hello, World" followed by a new line on the screen. Output is accomplished by the built-in println( ) method. The **System** is a predefined class that provides access to the system and **out** is the variable of type output stream connected to the console.

Comments

They can either be multiline or single-line comments.

```
// This is a simple Java program.
// Call this file "HelloWorld.java".
```

This is a single-line comment. This type of comment must begin with // as in C/C++. For multiline comments, they must begin from /* and end with */.

Important Points

- *The name of the class defined by the program is HelloWorld, which is the same as the name of the file(HelloWorld.java). This is not a coincidence. In Java, all codes must reside inside a class, and there is at most one public class which contains the main() method.*
- *By convention, the name of the main class(a class that contains the main method) should match the name of the file that holds the program.*
- *Every Java program must have a class definition that matches the filename (class name and file name should be same).*

Compiling the Program

- After successfully setting up the environment, we can open a terminal in both Windows/Unix and go to the directory where the file – HelloWorld.java is present.
- Now, to compile the HelloWorld program, execute the compiler – javac, to specify the name of the **source** file on the command line, as shown:

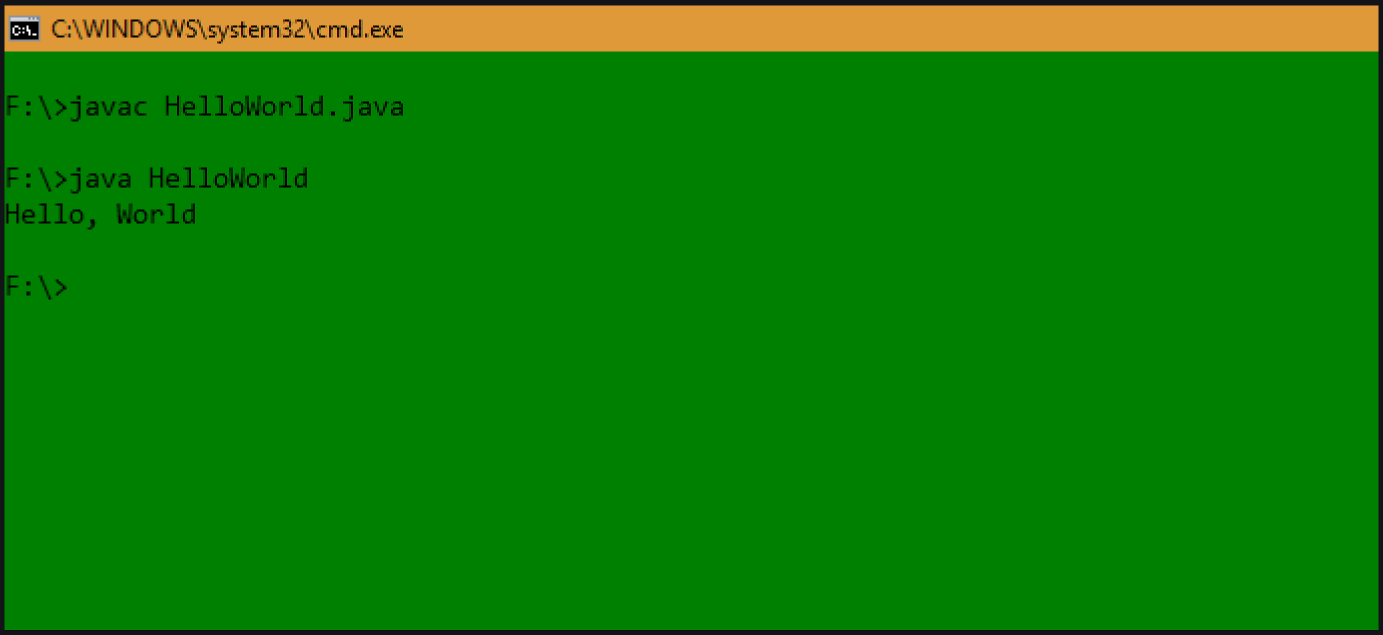
```
javac HelloWorld.java
```

- The compiler creates a HelloWorld.class (in the current working directory) that contains the bytecode version of the program. Now, to execute our program, **JVM**(Java Virtual Machine) needs to be called using java, specifying the name of the **class** file on the command line, as shown:

```
java HelloWorld
```

- This will print “Hello World” to the terminal screen.

In Windows

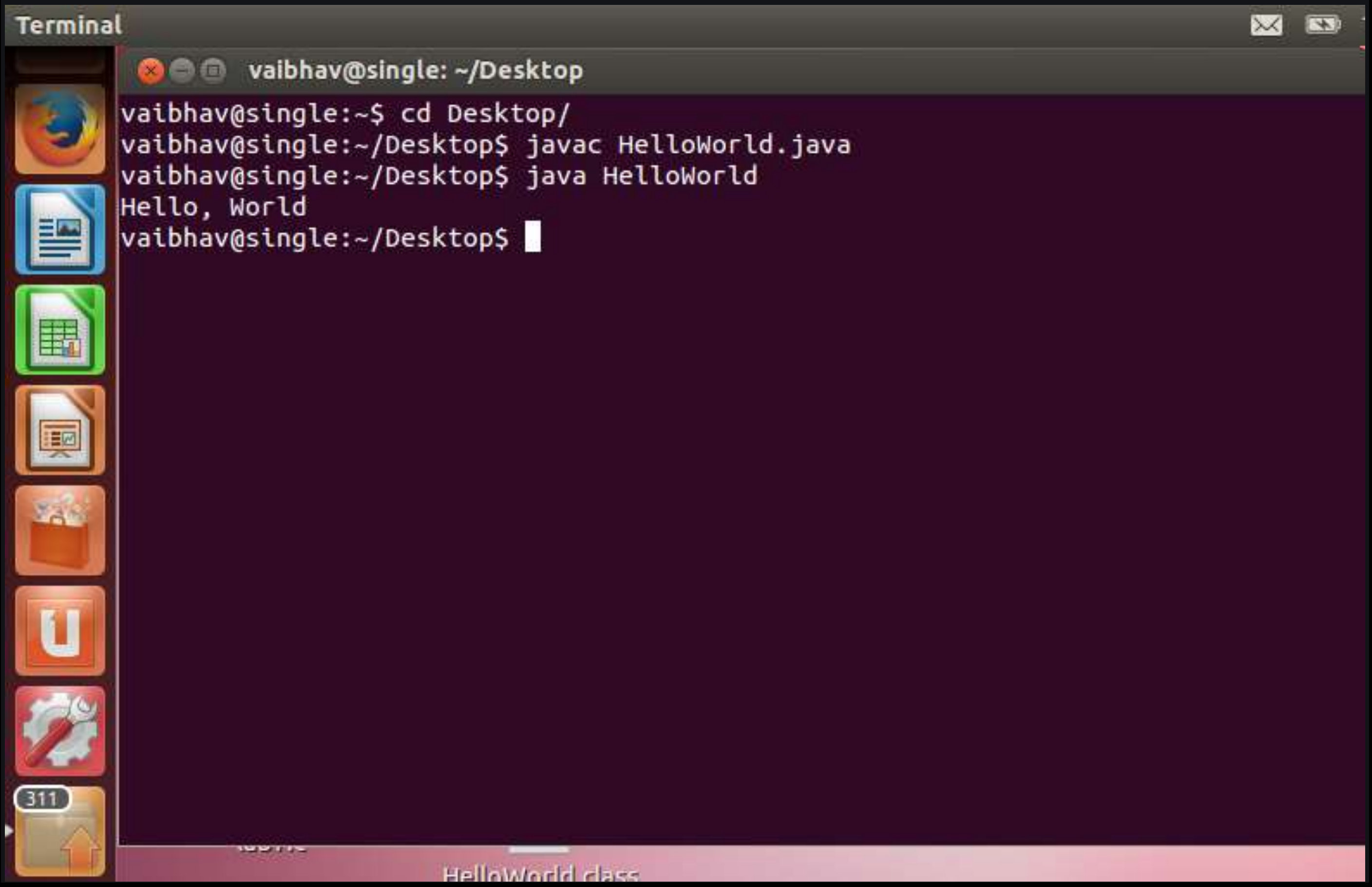


```
F:\>javac HelloWorld.java

F:\>java HelloWorld
Hello, World

F:\>
```

In Linux



```
vaibhav@single:~$ cd Desktop/
vaibhav@single:~/Desktop$ javac HelloWorld.java
vaibhav@single:~/Desktop$ java HelloWorld
Hello, World
vaibhav@single:~/Desktop$ 
```