

Java Hello World Program

Difficulty Level : Basic Last Updated : 10 Nov, 2021

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.

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.

```
// 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 “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
```

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 whose signature is:

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

- **public:** So that JVM can execute the method from anywhere.
- **static:** The main method is to be called without an object. The modifiers public and static can be written in either order.
- **void:** The main method doesn't return anything.
- **main():** Name configured in the JVM.
- **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.

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

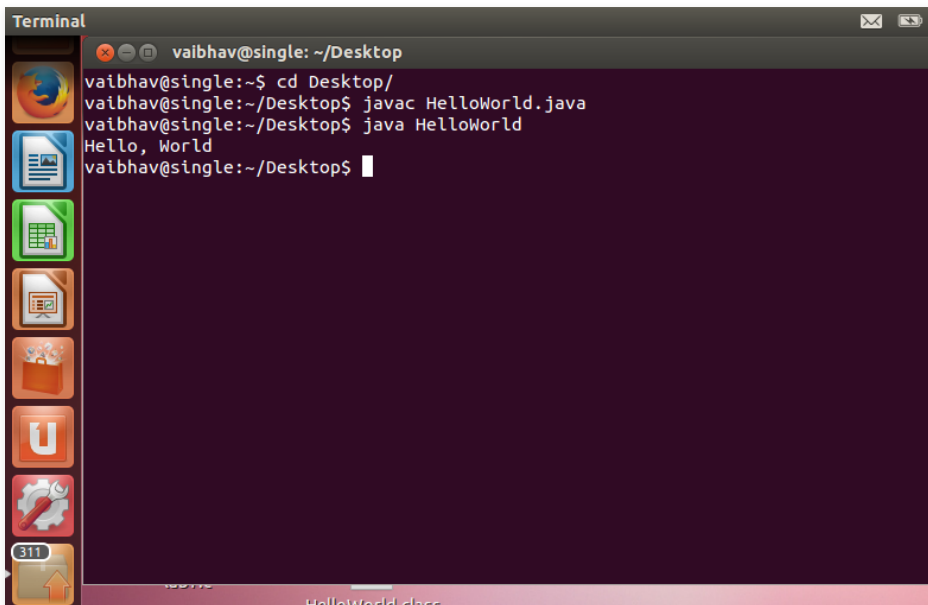
```
C:\WINDOWS\system32\cmd.exe
```

```
F:\>javac HelloWorld.java
```

```
F:\>java HelloWorld  
Hello, World
```

```
F:\>
```

In Linux



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

This article is contributed by **Gaurav Miglani**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [write.geeksforgeeks.org](https://www.geeksforgeeks.org/write-a-better-article/) or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.