How to Compile and Run your First Java Program

BY CHAITANYA SINGH | FILED UNDER: LEARN JAVA

In this tutorial, we will see how to write, compile and run a java program. I will also cover java syntax, code conventions and several ways to run a java program.

Simple Java Program:

```
public class FirstJavaProgram {
  public static void main(String[] args){
    System.out.println("This is my first program in java");
  }//End of main
}//End of FirstJavaProgram Class
```

Output: This is my first program in java

How to compile and run the above program

> **Prerequisite:** You need to have java installed on your system. You can get the java from here.

Step 1: Open a text editor, like Notepad on windows and TextEdit on Mac. Copy the above program and paste it in the text editor.

> You can also use IDE like Eclipse to run the java program but we will cover that part later in the coming tutorials. For the sake of simplicity, I will only use text editor and command prompt (or terminal) for this tutorial

Step 2: Save the file as **FirstJavaProgram.java**. You may be wondering why we have named the file as FirstJavaProgram, the thing is that we should always name the file same as the `public class` name. In our program, the `public class` name is `FirstJavaProgram`, that's why our file name should be **FirstJavaProgram.java**.

Step 3: In this step, we will compile the program. For this, open **command prompt (cmd) on Windows**, if you are **Mac OS then open Terminal**.
To compile the program, type the following command and hit enter.

```
javac FirstJavaProgram.java
```

> You may get this error when you try to compile the program: "**javac' is not recognized as an internal or external command, operable program or batch file**". This error occurs when the java path is not set in your system

If you get this error then you first need to set the path before compilation.

Set Path in Windows:
Open command prompt (cmd), go to the place where you have installed java on your system and locate the bin directory, copy the complete path and write it in the command like this.

```
set path=C:\Program Files\Java\jdk1.8.0_121\bin
```

Note: Your jdk version may be different. Since I have java version 1.8.0_121 installed on my system, I mentioned the same while setting up the path.

Set Path in Mac OS X
Open Terminal, type the following command and hit return.

```
export JAVA_HOME=/Library/Java/Home
```

Type the following command on terminal to confirm the path.

```
echo $JAVA_HOME
```

That's it.

> The steps above are for setting up the path temporary which means when you close the command prompt or terminal, the path settings will be lost and you will have to set the path again next time you use it. I will share the permanent path setup guide in the coming tutorial.

Step 4: After compilation the .java file gets translated into the .class file(byte code). Now we can run the program. To run the program, type the following command and hit enter:

```
java FirstJavaProgram
```

> Note that you should not append the .java extension to the file name while running the program.

Closer look to the First Java Program

Now that we have understood how to run a java program, let have a closer look at the program we have written above.

```
public class FirstJavaProgram {
```

This is the first line of our java program. Every java application must have at least one class definition that consists of `class` keyword followed by class name. When I say keyword, it means that it should not be changed, we should use it as it is. However the class name can be anything.

I have made the class public by using public access modifier, I will cover access modifier in a separate post, all you need to know now that a java file can have any number of classes but it can have only one public class and the file name should be same as public class name.

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

This is our next line in the program, lets break it down to understand it:
public: This makes the main method public that means that we can call the method from outside the class.

static: We do not need to create object for static methods to run. They can run itself.

void: It does not return anything.

main: It is the method name. This is the entry point method from which the JVM can run your program.

(String[] args): Used for command line arguments that are passed as strings. We will cover that in a separate post.

```
System.out.println("This is my first program in java");
```

This method prints the contents inside the double quotes into the console and inserts a newline after.