ICT 2123
Object Oriented Development

Level 2 _ Semester 1 **Lab Sheet 01**

Checking the initial Setup for Java. Getting Started with JDK for Windows

This page will demonstrate how to compose, compile, and run a simple java program on your local PC running windows.

To ensure all the procedures work, JDK (Java Software Development Kit) must be installed properly on your PC.

To check whether Java Running environment is properly installed in the machine type the following command in your command prompt.

java -version

If the java environment is properly installed you may see the currently installed JAVA version in the machine. If nothing is given in then JDK may be need to install in the machine.

Installing JDK version

Follow the steps given in the following website to download JDK

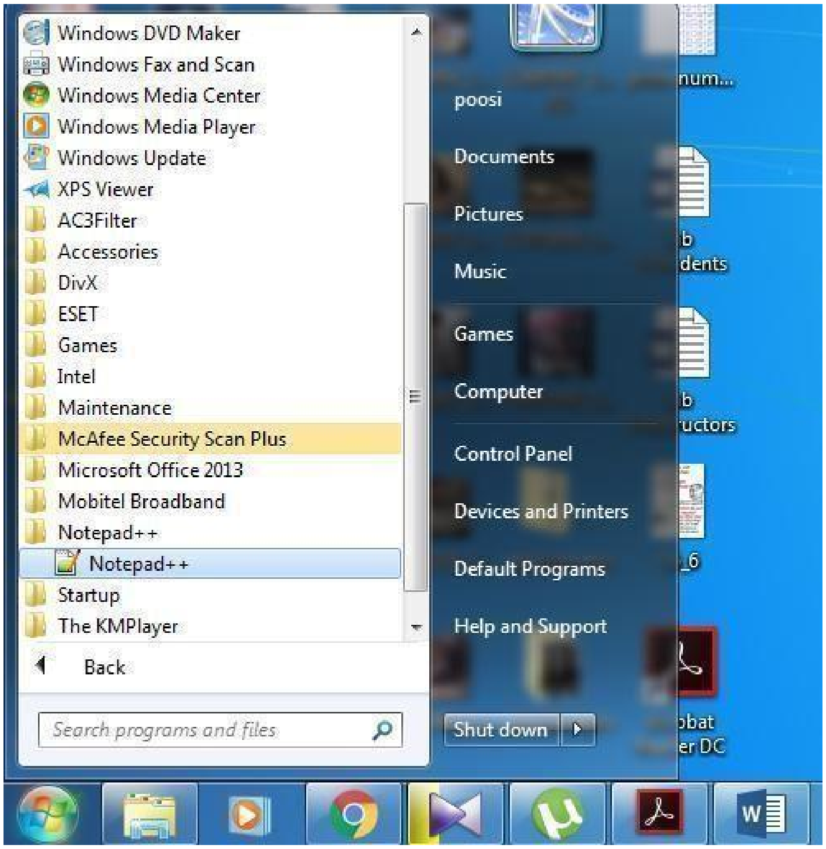
https://phoenixnap.com/kb/install-java-windows

Start a new command prompt and type the commands

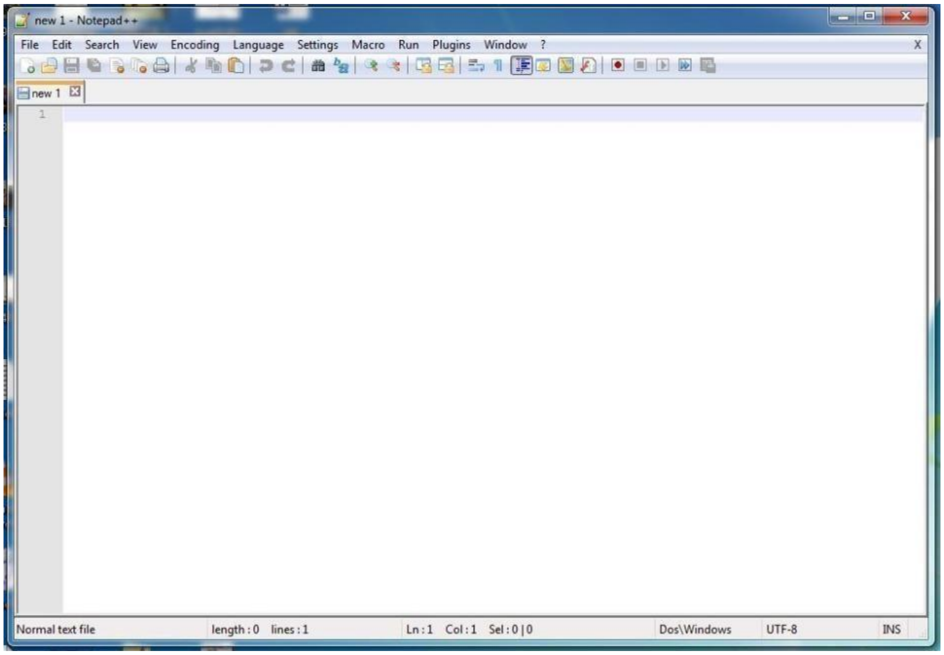
```
java -version
javac
```

Create a new Java Program using

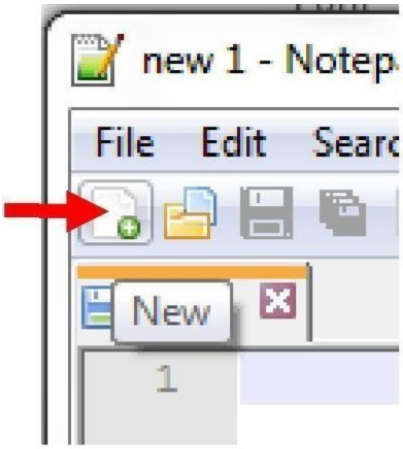
1. Click on Start → All Programs → Notepad++ → Notepad++ as below window.

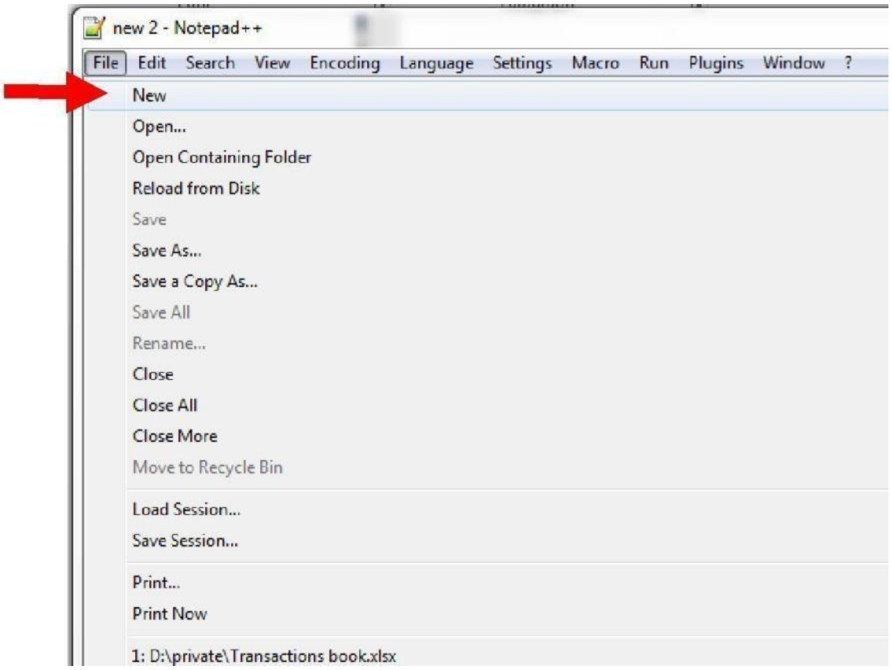


2. You will see the following screen:

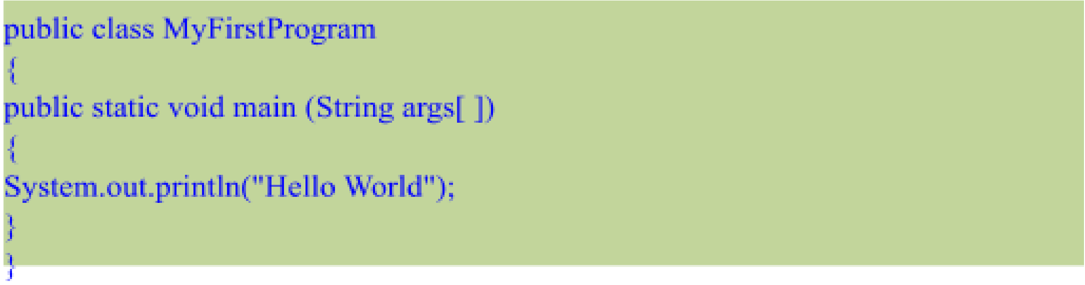

new 1 - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
1
Normal text file
length : 0 lines : 1
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8
INS

3. Now To create new Notepad file, click on New icon or click on File → New. Please find images below for your reference.



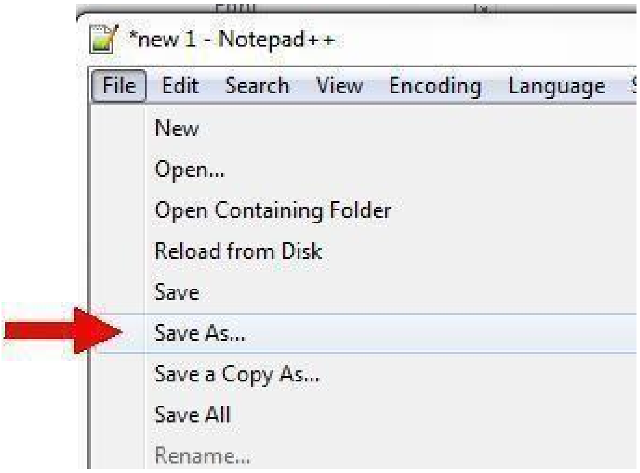


4. Write below first java Program using the Notepad++ editor.

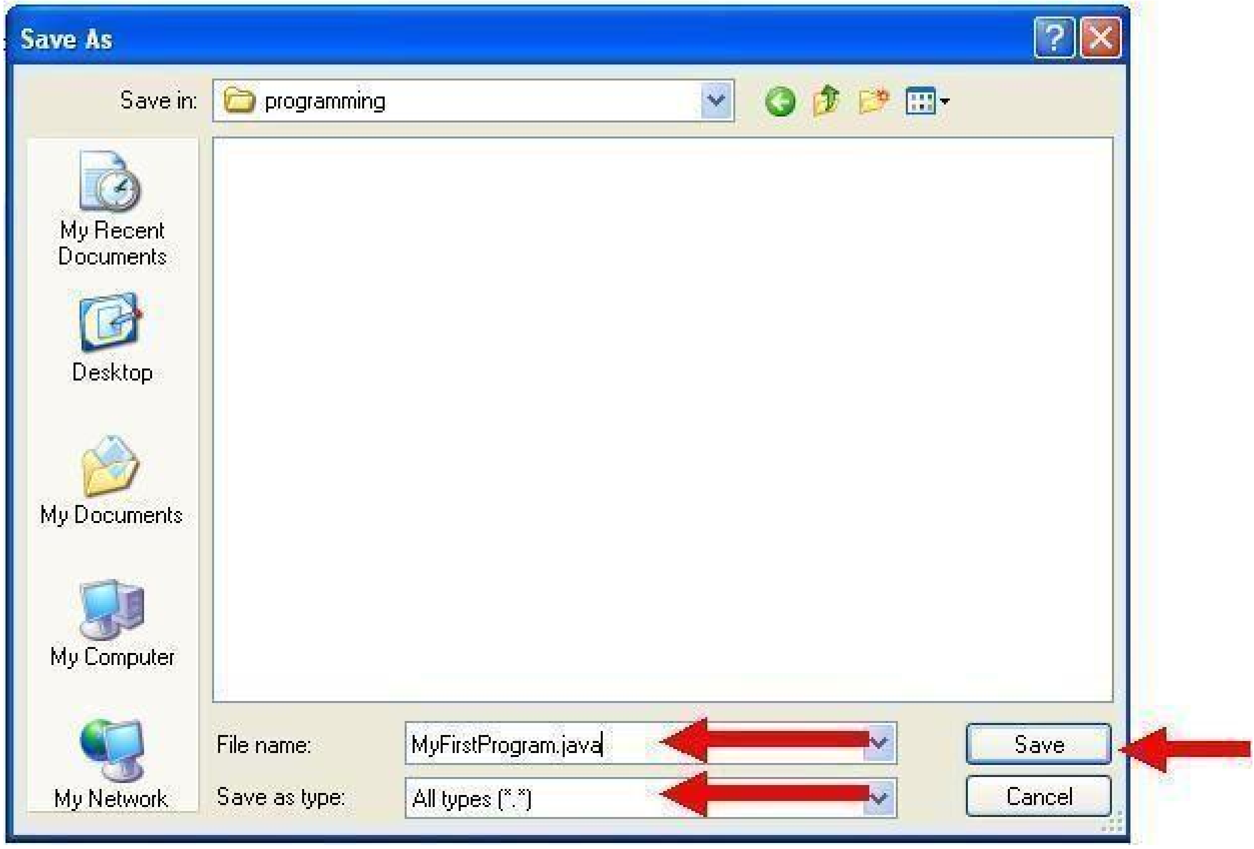


```
public class MyFirstProgram
{
public static void main (String args[ ])
{
System.out.println("Hello World");
}
}
```

5. Now save your file as a .java file. Click on File → Sava As



Select the location to save. In your case (D) drive → Student Data → Programming Inside student data folder create a folder and inside that folder save your java files.


Save in: programming

File name: MyFirstProgram.java

Save as type: All types (*.*)

Inside student data folder create a folder and inside that folder save your java files.

Give a File name

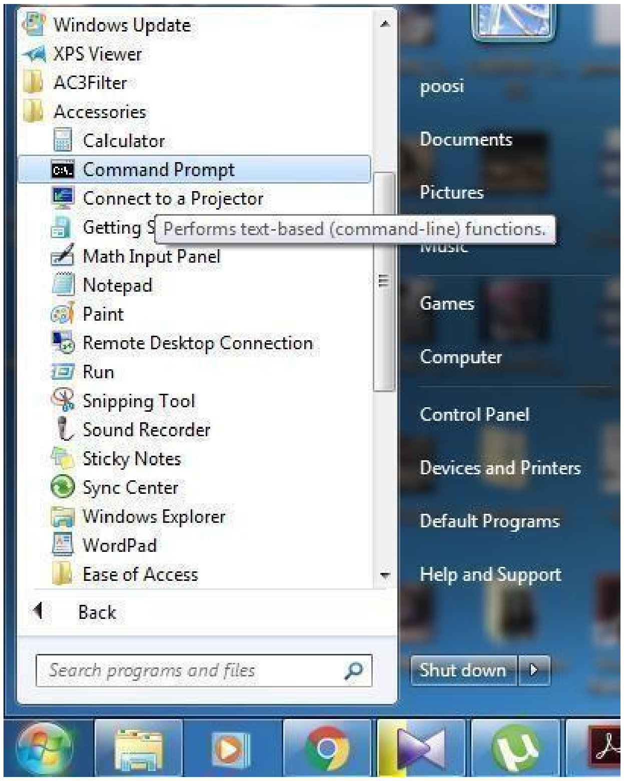
Eg: MyFirstProgram.java

Extension of the file name must be **.java**

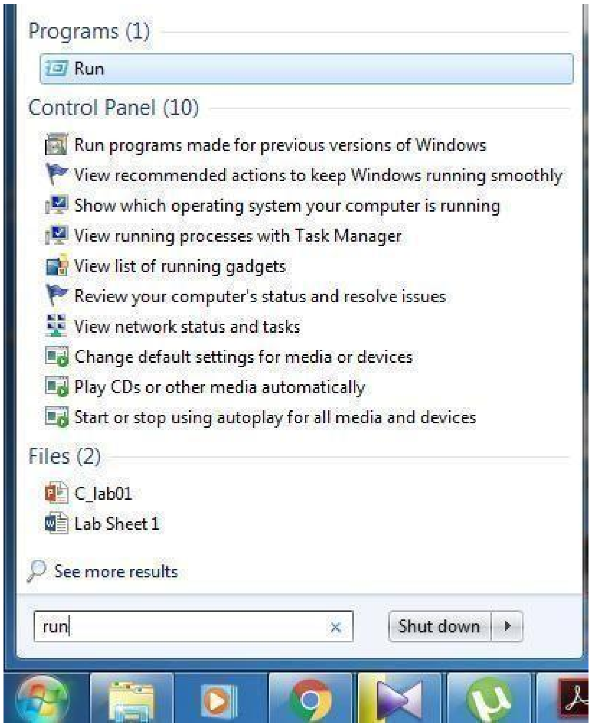
Save as type: All types (*,*) Then
Save.

1. Now open the command prompt.

Way 1: Start → Accessories → command prompt

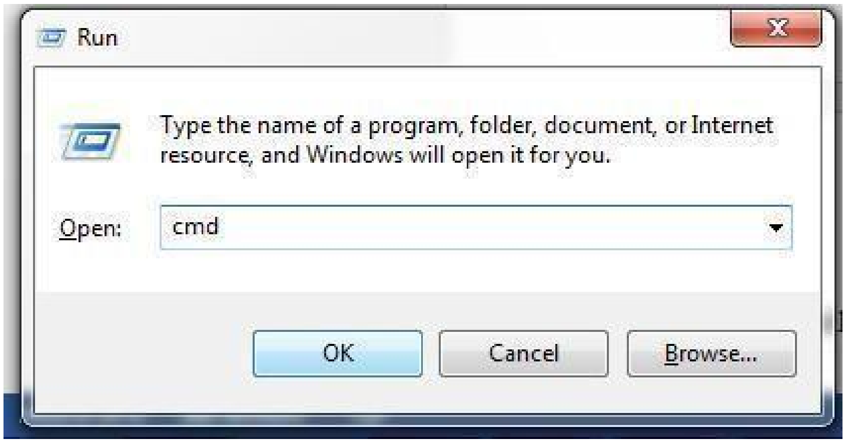


Way 2: Start → type "Run" in the search bar → enter


Programs (1)
Run
Control Panel (10)
Run programs made for previous versions of Windows
View recommended actions to keep Windows running smoothly
Show which operating system your computer is running
View running processes with Task Manager
View list of running gadgets
Review your computer's status and resolve issues
View network status and tasks
Change default settings for media or devices
Play CDs or other media automatically
Start or stop using autoplay for all media and devices
Files (2)
C_lab01
Lab Sheet 1
See more results
Shut down

run

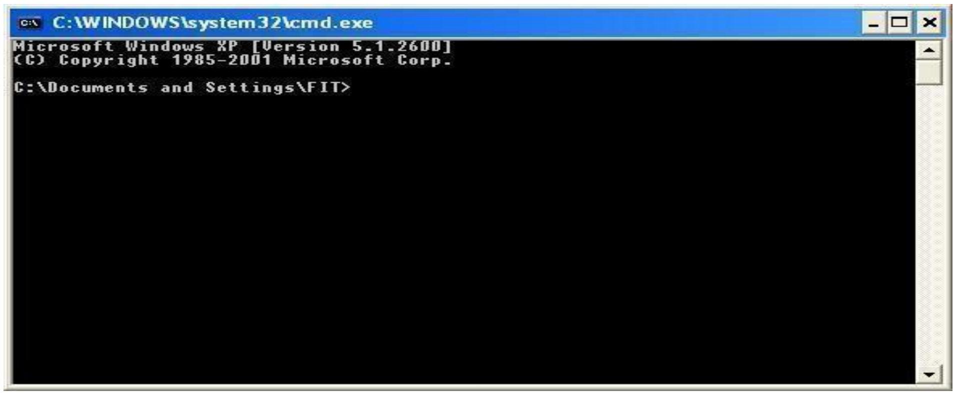
Then following dialog box will appear in the screen.
Then type “**cmd**” or “command” and press on OK button.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

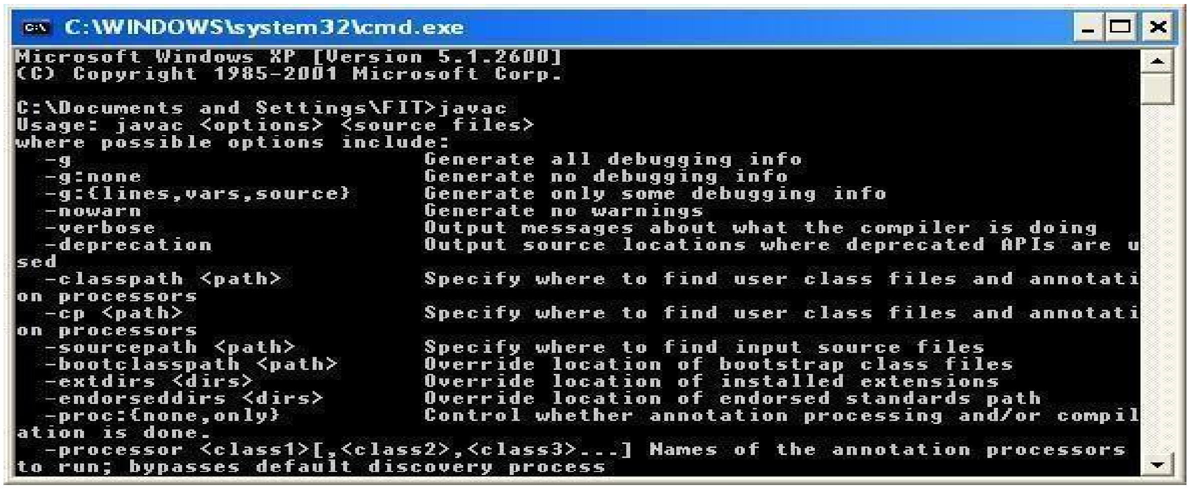
Then you can see the following screen.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\FIT>
```

2. Check whether the java compiler is installed in your computer. Type “**javac**” and press enter



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\FIT>javac
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                    Generate no debugging info
  -g:{lines,vars,source}     Generate only some debugging info
  -nowarn                    Generate no warnings
  -verbose                   Output messages about what the compiler is doing
  -deprecation               Output source locations where deprecated APIs are u
sed
  -classpath <path>          Specify where to find user class files and annotati
on processors
  -cp <path>                 Specify where to find user class files and annotati
on processors
  -sourcepath <path>         Specify where to find input source files
  -bootclasspath <path>      Override location of bootstrap class files
  -extdirs <dirs>            Override location of installed extensions
  -endorseddirs <dirs>       Override location of endorsed standards path
  -proc:{none,only}          Control whether annotation processing and/or compil
ation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors
to run; bypasses default discovery process
```

Even though you have installed the compiler, you haven’t set the variable path.
In labs you don’t have permission to set variable path manually, but you can set the path temporally.

Type: **path= C:\Program Files\Java\jdk1.7.0_51\bin**

1. Now set the location to where your c files are saved.

To change the drive:	**D:**
To Change the directory/folder:	**cd folderName**
	cd path

2. Now compile the java file

javac filename.java

E.g. javac MyFirstProgram.java

If you get any errors fix them and again compile the file.

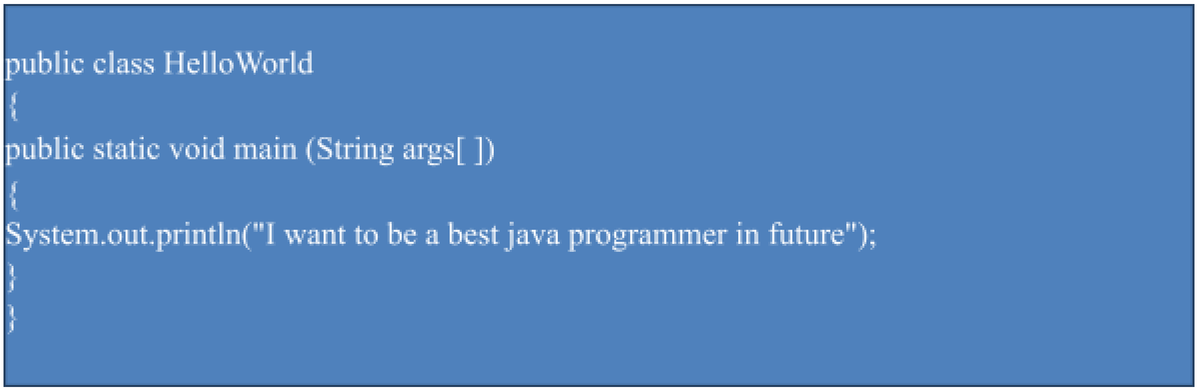
If there are no compile error messages, the compiler will create a file called **"MyFirstProgram.class"** in your current working directory.

3. The class file created by the compiler can be executed by the Java virtual machine on the PC. Now run the file

   java filename

   Now you can see the output of the java program on the command prompt.

Note that all commands and filenames in Java are case sensitive, so pay close attention to the case of file that you name. **Now try out with this example yourself.**



```
public class HelloWorld
{
public static void main (String args[ ])
{
System.out.println("I want to be a best java programmer in future");
}
}
```

For more details about java follow this link.

http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

Getting familiar with JAVA api and tutorial available

For any program available in JAVA language it contained many libraries which need to import and these may contain many different types of methods which we can used in our day to day programming.

To get an understanding on these methods available we can refer to the JAVA documentation which is known as the JAVA API. You can follow the following url to visit the java api and get hands on familiarization with it.

http://docs.oracle.com/javase/7/docs/api/

Example 1

Write a program to print:

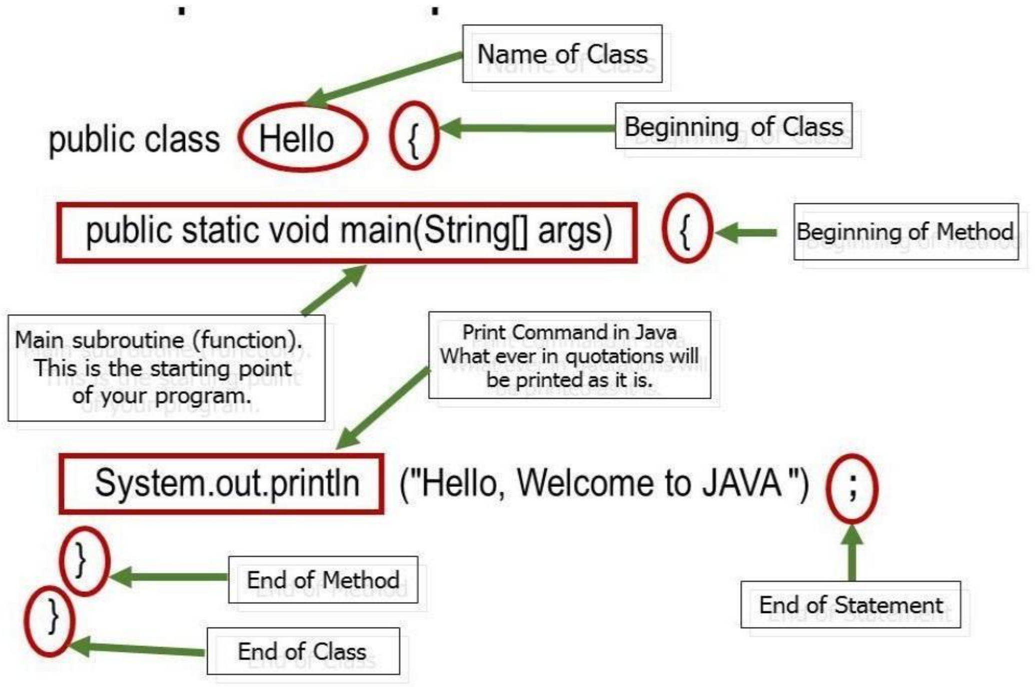
Hello, Welcome to JAVA.

This program demonstrates how to write a simple java program.

```
public class Hello {
    public static void main(String[] args) {
     System.out.println("Hello, Welcome to JAVA");
  }
}
```

- Use Notepad to type this program
- Save the program in your Z: drive as "Hello.java"

Important: Use the name for the .java file and for the class


Name of Class
public class Hello {
Beginning of Class
public static void main(String[] args) {
Beginning of Method
Main subroutine (function). This is the starting point of your program.
Print Command in Java What ever in quotations will be printed as it is.
System.out.println ("Hello, Welcome to JAVA ") ;
}
End of Method
}
End of Class
End of Statement

Java like C and C++ is case sensitive i.e. if you type the first part of the as

Public static void

Your program will not work

You must have saved the *Hello.java* program inside Z:\OOP folder

Running a Java Program

□ To run your Java Programs you need to call the Java Interpreter.

```
Z:\OOP>java Hello
```

Compiling a Java Program

□ Open a command prompt and type following commands. If JDK is properly installed, you should get error messages.

```
Z:\OOP>java
Z:\OOP>javac
```

□ If you do not get an error message set the “PATH” environment variable as follows. (Find the path to JDK installation in your machine & use it here)

```
Z:\OOP>set path="c:\Program Files\Java\jdk1.8.0\bin“
```

□ Type the following command. You should get the given output.

```
Z:\OOP> path
Output→PATH="c:\Program Files\Java\jdk1.8.0\bin“
```

□ When you type ***JAVA*** and ***JAVAC*** commands, now you should not get an error message.

□ To Compile your Java Program use the JDK’s javac tool.

```
Z:\OOP>javac Hello.java
```

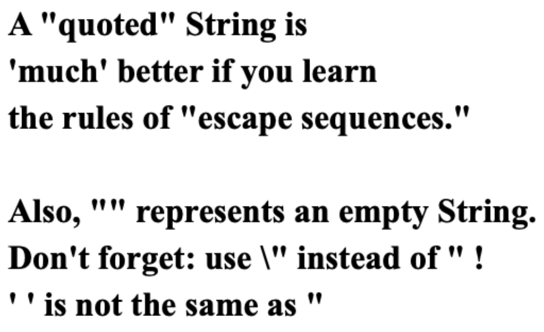
□ The compiler would create java byte code. These files have an extension of .class println( ) and print( ) are included of the **java.lang** package. The print keyword generates output without a carriage return and the println keyword does include a carriage return.

Use the following programs to check the difference between print() method and println() methods.

```
public class MyMessage1 {
        public static void main(String[ ] args) {
                System.out.print("JA");
                System.out.print("VA");
                System.out.println();
        }
}
public class MyMessage2 {
        public static void main(String[ ] args) {
                System.out.println("JAVA");
                System.out.print("World");
                System.out.println();
        }
}
```

Exercise 1

Try to obtain the following output by using the learned Backslash character Constant.

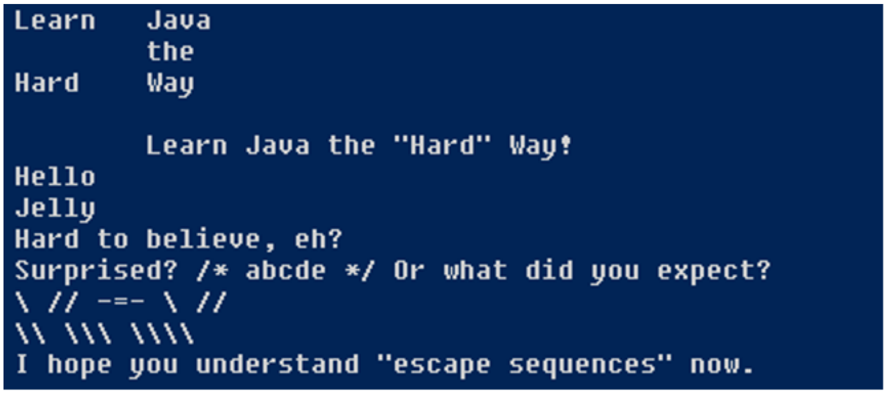


```
A "quoted" String is
'much' better if you learn
the rules of "escape sequences."

Also, "" represents an empty String.
Don't forget: use \" instead of " !
' ' is not the same as "
```

Exercise 2

Write a java program to print the following output using escape characters.



```
Learn   Java
        the
Hard    Way

        Learn Java the "Hard" Way!
Hello
Jelly
Hard to believe, eh?
Surprised? /* abcde */ Or what did you expect?
\ // -=- \ //
\\ \\\ \\\\
I hope you understand "escape sequences" now.
```

Exercise 3

Write a Java program to print 'Hello' on screen and then print your name on a separate line.

Expected Output :
Hello
Nimal

Exercise 4

Write a Java program by declaring three variables as follows and find the results after evaluating the following expressions.
int x ;
int y;
int result;

a) x = 10 ; y = 3 ; result = x + y / 3
b) x = 3 ; y = 57 ; result = (y % x) + 1;
c) x = 3 ; y = x+1 ; result = 1 ; result + = y– x;

Evaluate the final value (x) of following Arithmetic expressions. (Use BODMAS rule)
a) x = -4 + 5 * 6 - 10 / 2
b) x = 5*5/5 + 1 * (10 % 2)

Exercise 5

Take a four digit number. Write a program to calculate the sum of the first and the second last digit of a 5 digit number of the number taken.
E.g.- NUMBER : 12345   OUTPUT : 1+4=5

Exercise 6

Suppose 'a' and 'b' are int type variables. Write two programs to swap the values of the two variables.
Note that the program should be written in a manner to get user inputs for variables ‘a’ and ‘b’.
1 - first program by using a third variable
2 - second program without using any third variable

Exercise 7

Write a program to reverse a 3-digit number.
E.g.-Number : 132   Output : 231

Exercise 8

Write the following program in a text document and save it with the name PrePostDemo.java. Compile the program and explain why the value "6" is printed twice in a row when it is executed:

```
class PrePostDemo {
    public static void main(String[] args){
    int i = 3;
    i++;
    System.out.println(i);
      ++i;
      System.out.println(i);
      System.out.println(++i);
      System.out.println(i++);
      System.out.println(i);
   }
}
```

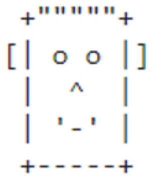
Exercise 9

Write a program that asks the user's name, and then greets the user by name.

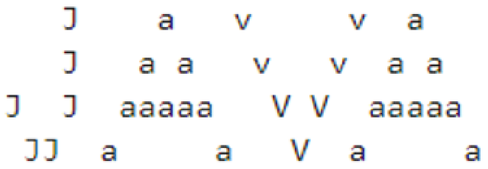
E.g : Please enter your name:
“Hello Chanduni, nice to meet you…!”

Exercise 10

Write a method to print each of the followings patterns WITHOUT using nested loops in a class called PrintPatterns.



```
 +"""""+
[| o o |]
 |  ^  |
 | '-' |
 +-----+
```



```
   J    a   v     v  a
   J   a a   v   v  a a
J  J  aaaaa   V V  aaaaa
 JJ  a     a   V  a     a
```

```
#                 # # # # # # # #       # # # # # # # #      #
# #               # # # # # # #           # # # # # # #      # #
# # #             # # # # # #               # # # # # #      # # #
# # # #           # # # # #                   # # # # #      # # # #
# # # # #         # # # #                        # # # #      # # # # #
# # # # # #       # # #                           # # #      # # # # # #
# # # # # # #     # #                               # #     # # # # # # #
# # # # # # # #   #                                   #     # # # # # # # #
```

Extend the above program to print above patterns using a single print command.