First - Java Programme -

Firstly every program in java is in classes.
ie; every file with ".java" extension is a class

```
public class Main {                    //Save file as Main.java.
    public static void main (String [] args) {
        System.out.println ("First Java Program");
    }
}
```

1. In java file name should be the class name which contains main method in it.
2. In Class name, the starting letter always has to be capital (it is a convention, but not a rule.) So following convention is a good practice

Explanation of terms in program

public :- It is one of the access specifiers; which provides scope to access variables & methods
↳ Here public means; all the functions & properties in that class/method can be accessible by anyone.
↳ Main method containing class should always be public.

class :- class is a datatype.

↳ We can also say it is a ~~—~~

↳ It is a named group of properties & functions.

↳ Some times we can also create user-defined data-type using class

static : It is a key word.

↳ It doesn't requires objects help to access properties & function.

ie, In order to run main() method without creating object of class, so we static.

↳ All static variables, blocks, methods are executed first

main : main() is the name of the method.

↳ It always should be main only.

↳ Program execution starts from main() only.

↳ It is the starting point of program execution

↳ It should always be public. So that program can be run & accessible by anywhere & anyone.

void : It is a return type.

↳ Which the method doesn't return anything.

String[] args :

It is command line arguments.

↳ It is also a string array of name args.

if we System.out.println(args[0]);

↳ While compilation.

javac Main.java

java Main 30

→ It prints '30' // first element of args array.

ie, What ever arguments, we are giving in terminal, those are stored in string array.

// javac Main.java.

// java Main 30 "Anonymus"

// prints - Anonymus.

S.o.p(args[1])

System : It is a pre-defined class in java, which is present in java.lang package.

↳ Which is mainly used to printing some stuff on the screen.

↳ It also contains some fields & methods as it is a class.

out : Variable of system class.

↳ It is a type of printstream.

println : Is a method which is used to print the message in new line.

→ It takes a string / anything & outputs it.

keyboard ← System.in } Standard input
display ← System.out } output stream

Inputs in java:

(We can provide inputs in java through Scanner class.

```
import java.util.Scanner;
public Main {
public static void main(String [] args) {
    Scanner input = new Scanner(System.in);
    System.out.println("Enter an integer" + input.nextInt());
    System.out.println(input.nextLine());
    System.out.println(input.nextFloat());
}
}
```

=> Enter an integer
19
Man
98.0

|| i-new - keyword, allocates memory.

~~Output~~ 19
Man
98.0

Data-Types. [Nothing but the type of the data].

Primitive

- integer — `int rno = 19`
- float — `float mark = 91.0f`
- double — `double d = 3343467.456;`
- short int, long int — `long l = 3456789 10L;`
- ~~long double~~
- char — `char ch = 'r';`
- boolean — `boolean check = true;`

Non-primitive.

- Strings
- Arrays
- Classes.

Primitive: Is a datatype, which we cannot break further.

↳ We add f in float because, every decimal values are by default double, inorder to differentiate we use f.

↳ We add L in long because to specify the larger integer values.

↳ If we don't add L, then they are basically int [anyways that depends on size].

integer – 4 byte
float – 4 bytes.
boolean – 1 bit
char – 2 bytes.
double – 8 bytes
long – 8 bytes.
short – 2 bytes

Type Conversion / Casting:

Converting one type of data into other type of data.

→ There are 2 types of type casting.

Implicit TC / Automatic / Broadening

(*) ~~float a = 10~~

```
float a = input.nextFloat();
```

↳ Here even if we the input taken in int formart is automatically assigned to float by compiler.

Rules for implicit TC

↳ Destination type should be greater than source type.

↳ L.H.S > R.H.S.

Another ex:

```
float a = 10;
S.o.p(a);
Prints 10.0
```

10 - is an integer but it is implicitly casted to float.

```
int num = 'A';
S.o.p(num)
```

O/P : 65.

Explicit TC / Narrowing

(*) Conversion of higher data types to lower

↳ Should be done by us

```
float f = 10.2f;
int marks = (int) (f);
Prints :- 10
```