First Java Program

- Create a file Main.java
- every 'filename that ends with' .java' is a class in itself.
- whichever variable starts with a capital letter, it means its a class. Good convention.
- If the name of the file is Main.java then the Main class that is in this file has to be 'public'.

  public means this class can be accessed from anywhere.
- We have to create a function inside this class, where our program is called for starting. This function is known as ~~Main~~ 'main' and can't be changed.

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

Main.java

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

go to compiler →

javac. Main. java.

This creates a file 'Main. class' into our folder which is the bytecode.

Run →

java Main.

Output →

Hello World!

- All the functions inside a class is known as methods.
- main function is public → it is function at which the program starts running so it only makes sense to make it available to be accessible from anywhere
- main function is static → in order to run a function inside a class we have to create an object of that class, if we don't want main function to stop running (if it doesn't have any object), we make it static
- void determines the return type of the function.
- String [0] is the array & args define the command line arguments.

- eg. sysout(*args[0])

javac ~~demo.java~~ Main.java.
java Main 40.

o/p (40)

- javac -d .. Main.java

Stores our byte code into previous directory.

best practice → Source code in src.
byte code → outside.

- $ echo $PATH

Stores our path variables. This is where our executable (javac, git etc. are stored).

$ where javac
$ where git

Intellij Idea

psvm + enter
sout + enter.
Alt + Insert → create new file / package.

- System.out.println

'System' is a class, it has a variable 'out' which is of type 'printstream' and 'out' has a method 'println'. By default the standard output stream is the command line.

- Input

```
Scanner input = new Scanner(System.in);
Sout(input.nextInt());
```

Note: Every class in Java extends the object class.

- Sout(input.next()); prints string till space
  sout(input.nextLine()); prints entire line.

Primitives

Data types that can't be broken into any other data type.
- string is not a primitive data type.

```
float rounds = 98.67f;
long integer = 3435763625L;
```

By default decimal values are of type double so we add f same goes long with Lut.

```
int a = 10;
```

int → data type, a → identifier, 10 → Literal

```
Lut a = 234,000,000;   X
Lut a = 234_000_000;   ✓
```

Debug

Sem. 2 aug

Type Conversion (Type Casting in Java)

- The two Types should be compatible } Automatic type casting
- Destination type > Source type } condition.

```
int a = 257;
byte b = (byte) a; // 257 % 256 = 1
```

(∵ Max storage = 256)

```
byte a = 40;
byte b = 50;
byte c = 100;
int d = (a*b)/c;
```

- a*b exceeds storage value of byte
- while computing, java automatically promotes (a*b) to int type.

```
byte b = 50;
b = b*2;
```

- This will give an error as b*2 is of int type so we can't assign it to byte type.
- double > float > int > ~~byte~~ char > byte.

- Prims.java
- Basics.java (eg. for, while)
- Temperature.java