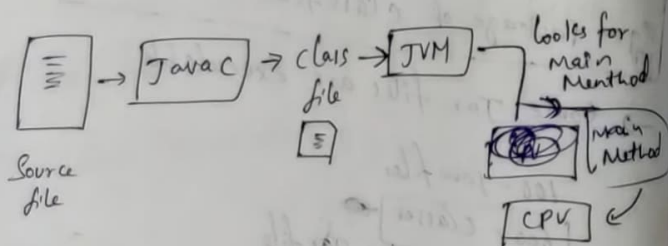


27/10/22

- Application/Software/Program all refer to something.
- Every Application have different Modules.
- Start point of Application should be defined.
 - ↳ main method comes into picture.
- OS → operating System
 - ↳ operate all the Application and Software.



→ JVM look for main method if it not found it will not give control of execution.

→ ~~Execution~~ Java program is not executed

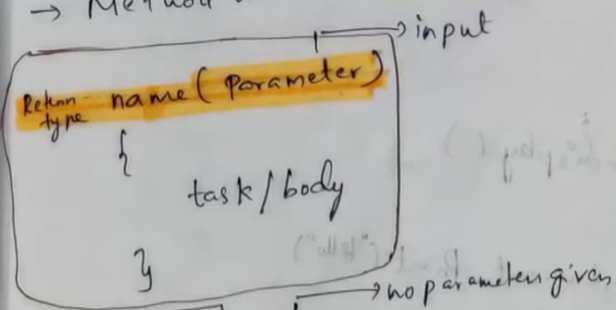
① What is Method

Application/Software → Multiple activities

① Java program :- Task (or) work (or) activity.
 ↳ we have write within method

② Java Method :-

- Method will have name
- Method will have parameter
- Method will have body
- Method will have return type



Eg
 ①

```

Void add ( )
{
    1+2
}
  
```

→ Adding two num.

add() :- Calling Method then Program in method get executed.

② Print Hello World

```
Void display()  
{  
    System.out.print("Hello");  
    // → no return  
}  
  
display();
```

→ ~~the~~ if there is nothing to return ~~the~~ any value then write Void as return type
→ it will not any value to Caller.

Example ③

```
int display()  
{  
    System.out.print("Hello")  
    return 10;  
}  
  
display();
```

↳ After Executing it also returns value to Caller

① O.S - to give Control to the program it expect Main Method. That task will be Executed

C-programming

```
Void main()  
{  
    Print f("Hello")  
}
```

② JVM - assistance to Execute Java program
↳ JVM starts when there is main method in program.

Eg. Class launch

```
{  
    Void main()  
    {  
        System.out.println("Hello");  
    }  
}
```

→ JVM Cannot see main method because it wrapped in between class. to make it visible we have to use public.

Eg.

```
Public Void main()  
{  
}
```

- if we ^{are} public for Method it will be visible in Entire project.
- What ever is visible it is not accessible
- for accessible we need to use **Static**.

```

class Launch() {
Static public void main()
    static
{
    System.out.println("Hello");
}
}

```

- it is accessible to JVM
- Everytime you create class in java we have to create object.

Eg

Class Demo

```

{
    = m1 → methods
    = m2 →
}

```

→ Object

m1, m2 Cannot be called outside class and Executed.

- Java will not allow
- First we have to create object of class
- Then we have call Method.

- Java will not allow to call Methods which are inside class directly.

- Objects have to create and then Method can be called.

⇒ Create object of class.

- Static - we can call Methods inside class with out creating object by Making Method Static.

- object Creation is not need.

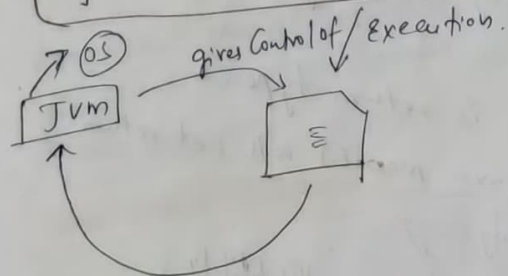
- Then JVM give you the Control Execution.

Eg

```

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

```



Class launch

```

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

```

Annotations: ④ Return type, ⑦ Parameter, ① name, ③ body

⑤ String [] args

→ To receive Command line arguments.
 → String [] args is a parameter which there in Main Method in java for the purpose of reliving Command line argument.

Class launch {
 public static void main (String [] args)
 {
 System.out.println ("Hello");
 }
}

① Public:- Main Method is visible to everyone across the project.

② Static:- it Can be accessed without Creating any object.

② Main method:-

① name of method and JVM will search for this name only.

~~void~~

② Void:- it is return type.
 → Java main() will not return anything

→ Public is to Increase visibility

→ Static → Can be accessed without object Creating.

→ Save .Java file
 → launch.java name/drive:
 → open cmd :- javac launch.java
 Java launch → giving Control to JVM to execute

→ Anything we instruct in cmd prompt it is Called Command line argument.

How to give arguments from Command Prompt:-

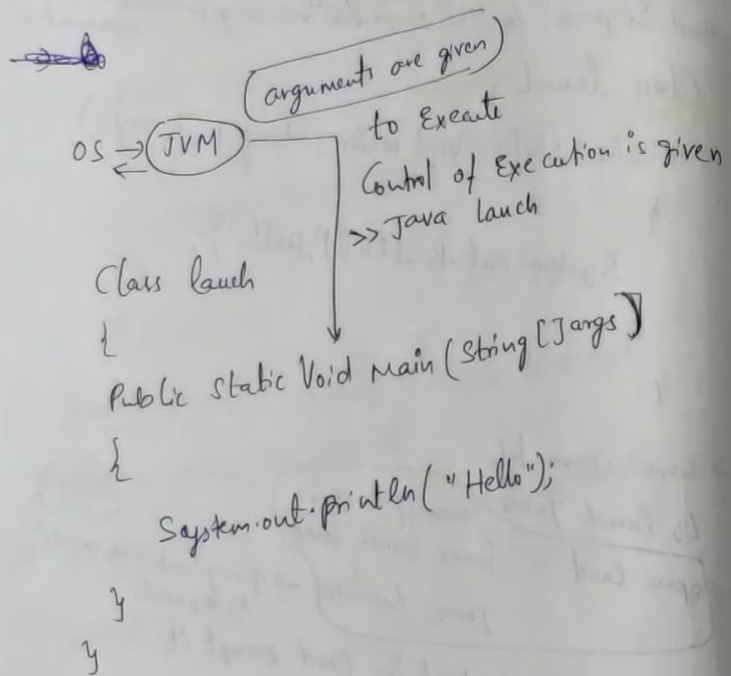
→ Arguments → Information or data.

→ from cmd prompt while Execution if you want to pass some information to Java program.

→ java launch incanon.ai
 Command line argument
 → to receive Command line arguments

→ from Command line we passing the arguments.

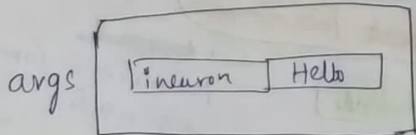
(→ whether you pass ~~any~~ Command line argument
(or) not we have to write String[] args).



→ arguments are given during execution.

→ arguments are stored in args

Eg: Java launch $\frac{\text{incuron}}{0}$ $\frac{\text{Hello}}{1}$
 $\underbrace{\hspace{1.5cm}}_{\rightarrow \text{arguments given}}$




→ We can print info on Arguments stored in
(String[] args).

(String[] args) :

Printing Values Stored in ~~args~~ String[] args

System.out.println("Hello"); Print

System.out.println("args[0]"); → neuron

System.out.println(args[1]); →  Hello

→ javac launch.java
java Launch neuron Hello

→ o/p = 1 neuron Hello

~~o/p~~ ~~branch~~ ~~in~~

→ irrespective of type of info passing in through Command line it will be taken a String in method.

Java Launch

inuron 10 0.5
↓
string dint float

All ~~type~~ value are stored as string type.

(

① Public - to increase visibility.

② static - for accessibility.

③ Void - return type.

④ main - JVM

⑤ String args() - to receive command line arguments.

→ This syntax is allowed

⇒ `Public static Void main(String[] args)` ✓

`Public static Void main(String args[])` ✓

`Static Public Void main()` ✓

`static Public main Void` X not allowed.

`Public static Void main(Strings... args)` ✓

Short Cut - `Ctrl + Space`

⑥ Variable - to store some information and data

Eg. `a = 10;`

What ever given on right side given to left side

a [10]

Difference between Statically typed and dynamically typed programming language.

Statically typed P.L	Dynamically typed P.L
<pre>int a = 10; S.O.P(a);</pre>	<pre>a = 10 Print(a);</pre>
<p>→ In C, C++, Java we have to specify what of data is stored in variable.</p> <p>→ Such P.L where before we compile we have to specify type of data.</p>	<p>→ no need to give type.</p> <p>Eg. Python, JS.</p>

(1) statically typed programming language

→ type of data being checked during the compile time itself. E.g. C, C++, Java.

(2) Dynamically typed P.L. Such a

→ type of data is not be checked during the compile time directly while executing behind scenes it will only understand

E.g. Python, is:

→ Java is statically typed programming language.

