Types of Method in Java :

* As of now, In Java we have only two types of method

  1) Static Method (Object is not required)
  2) Non Static Method (Object is required)

1) Static Method :

* In Java, If we declare a method with static keyword (static modifier) then it is called static method.
* In order to call the static method, **Object is not required, We can call static method with the help of class name.**

```
Example :

public staic void access()
{
  //static method
}
```

2) Non static method :

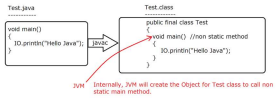
* In Java, If we declare a method without static keyword (static modifier) then It is called non static method.
* **In order to call non static method, Object is required.**

```
Example :

public void access()
{
  //Non static method
}
```

main() :

* main is user-defined method because user is responsible to write the logic inside main method.
* The execution of java program always starts from main method.[Entry Point]
* OUR JAVA MAIN METHOD IS A **NON STATIC** MAIN METHOD SO INTERNALLY JVM WILL CREATE THE OBJECT FOR THE CLASS WHICH IS GENERATED BY JAVA COMPILER TO CALL THE MAIN METHOD



Can we write multiple methods with same name in java source file :

Yes, We can write multiple methods with same name but parameter must be different otherwise code will not compile. [Known as Method Overloading]

Note: - We can also write multiple main methods with different parameter but JVM will always execute the main method which takes String [] args (String array) as a parameter as shown in the program below.

Test.java:

```
void main()
{
    IO.println("hyd");
}

void main(String [] args)
{
  IO.println("India");
}

//Output : India
```

Case 1:

```
void main ()
{
    IO.println ("Hello India");
}

public static void main(String [] args)
{
    IO.println("Hello World");
}
```

Output : Hello World

Case 2 :

```
void main(String [] args)
{
    IO.println("Hello India");
}

void main()
{
    IO.println("Hello World");
}
```

Output : Hello India

Case 3 :

```
void main(int [] args)
{
    IO.println("Hello India");
}

void main()
{
    IO.println("Hello World");

}
```

Output : Hello World

Why java has taken String [] args as a parameter to the main method :

* String is a predefined class available in java.lang package [like header file]. It is a collection of alpha-numbric character.
* Our main method accepts String array as a parameter so It can accept wide range of values. [Remember args is an array variable so It can hold multiple values].

Demo.java

```
void main(String [] args)
{
    String a = "NIT";
    IO.println(a);

    a = "123";
    IO.println(a);

    a = 'A';
    IO.println(a);

    a = "true";
    IO.println(a);

    a = "12.90";
    IO.println(a);
}
```

IO class :

* IO stands for Input & Output, It is a predefined class available in java.lang package from JDK 25V(LTS).
* It is mainly used to provide input and output operations in Java.
* IO class provides various static methods so we need not to create object for IO class to call these static method.

IO class static method :

1) public static void println(Object obj) :

* It is a predefined static method of IO class. [Object is not requied]
* It is used to print the data on the console.
* It accepts Object as a parameter so we can assign different value to this println() method.

WAP to provide Welcome message in Java :

Welcome.java

```
void main()
{
    IO.println("Welcome to Java Language");
}
```

WAP to add two numbers :

Addition.java

```
void main()
{
    int x = 10;
    int y = 20;
    int sum = x + y;
    IO.println(sum);
}
```

The above program is generating the output is 30, but It is not user friendly message

How to provide user-friendly message in Java :

* In order to provide user-friendly message we need to take the support of String concatenation operator

Behaviour of String concatenation Operator (+)

```
1 + 1 = 2 [Here + Operator is working as Arithmetic Operator]

1 + "1" = 11 [Here + Operator is working as String concatenation Operator]

"2" + 1 = 21 [Here + Operator is working as String concatenation Operator]

"Java " + 11 = Java 11 [Here + Operator is working as String concatenation Operator]
```

The '+' operator will work as String concatenation operator, If any of the operand is String type.

Sum.java

```
void main()
{
    int x = 10;
    int y = 20;
    int sum = x + y;
    IO.println("The Sum is :"+sum);
}
```

WAP to add two numbers without using 3rd Variable :

```
void main()
{
    int x = 10;
    int y = 20;
    IO.println("The Sum is :"+x+y); //The Sum is : 1020
    IO.println(x+x+y); //30
    IO.println(""+x+y); //1020
    IO.println("The Sum is :"+(x+y)); //The Sum is : 30
    IO.println(40 + 40+"NIT"+20 + 20); //80NIT2020
}
```