void contd...
* If we want to define a method body in java then return type is compulsory.
* It is a syntax rule so java compiler will not allow this statement.
* When we write method return type as a void then there is no communication between one module to another module.

Addition Module :

```
public static void doSum(int x, int y)
{
    System.out.println(x+y);
}
```

Main Module

```
public static void main(String []args)
{
    doSum(100, 200);
}
```

Here Main module is sending the value 100 and 200 to Addition module, Addition Module is consuming those values but not returning value, hence there is no communication between Main Module to Addition Module

Addition Module

```
public static int doSum(int x, int y)
{
    return (x+y);
}
```

Main Module

```
public static void main(String [] args)
{
    int result =  doSum(100,200);
    result = doSum(500,500);
}
```

In between these two modules, now communication is possible

* main method return type must be void otherwise code will compile but it **will not be executed** by JVM

main() :
----
* It is a user defined method because user is responsible to define the body of main method, Only the name is predefined.
* main method is the entry point of java program that means the execution of java program always starts and ends from main method.

Q) Can we write multiple methods with same name in java ?
----
* Yes, We can write multiple methods with same in java but **parameter must be different.**
* We can also write multiple main methods with different parameter but JVM will always execute the main method which accept String array as a parameter as shown below

```
public class Test
{
    public static void main(String batchName)
    {
        System.out.println(batchName);
    }

    public static void main(String []args)
    {
        System.out.println("Hello Everyone");
        main("Batch 46");
    }
}
```

String [] args :
----
* args is an array variable of type String.
* It is an array variable so it can hold multiple values.
* **S**trting is predefined class in java which is available in java.lang package(just like header file)

Why main method of java accept String array as a parameter ?
----
* String is a collection of alpha-numeric characeter. It must be enclosed with double quotes.
* It provides more wider scope so with the help of String array as a parameter we can accept **multiple values of different type.**
* In order to allow all different types of values and various type, Java decided to provide String array as a parameter to the main method.

```
public class Test
{
    public static void main(String args[])
    {
        String a = "NIT";
        String b = "B - 61, Ameerpet, Hyderabad";
        String c = "123";
        String d = "$%^&*&*&((";
        String e = "g";
        String f = "78.89";
        System.out.println("Hello Everyone");

    }
}
```

System.out.println() :
----
* It is an output statement in java through which we can print various types of data in the console.

System.out.println();

predefined class available in java.lang Package

out is a **reference variable** of type PrintStream class which is available in java.io package.

println() is a predefined method available in java.io.PrintStream class

* Actually System.out.println() represents HAS-A relation in java as shown below

```
public final class System
{
    public static final PrintStream out = null; //HAS-A relation
}
```

```
//WAP in java to add two numbers
public class  Addition
{
    public static void main(String[] args)
    {
        int x = 100;
        int y = 200;
        int z = x + y;
        System.out.println(z);
    }
}
```

We are getting the output 300 but it is not a user-friendly output.

How to get user friendly output :

```
public class  Addition
{
    public static void main(String[] args)
    {
        int x = 100;
        int y = 200;
        int z = x + y;
        System.out.println("Sum is :%d",z); //CE
    }
}
```

* In order to provide user friendly message java has provided String concatenation operator (+)

Behavior of String Concatenation Operator :
----
* It is also known as Arithmetic Operator the behavior of this '+' operator depends upon the operands
* IF ANY OF THE OPERANDS IS STRING TYPE THEN THE '+' OPERATOR WILL BEHAVE AS STRING CONCATENATION OPERATOR

Example :

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

"2" + 2 = 22   [Here + is working as String concatenation Operator]

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

1 + "123" = 1123 [Here + is working as String concatenation Operator]

```
//Addition of two numbers by using user friendly message
public class  Addition
{
    public static void main(String[] args)
    {
        int x = 100;
        int y = 200;
        int z = x + y;
        System.out.println("Sum is : "+z);
    }
}
```

```
//Program to add two numbers without using 3rd variable
public class AdditionOfTwoNum
{
    public static void main(String[] args)
    {
      int x = 100;
        int y = 200;
        System.out.println("Sum is :"+x+y); //100200
        System.out.println(+x+y); //300
        System.out.println(""+x+y);//100200
        System.out.println("Sum is :"+(x+y));
    }
}
```