

## Chapter 7 - Methods in Java

Sometimes our program grows in size and we want to separate the logic of main method to other methods

For instance - If we are calculating average of a number pair 5 times, we can use methods to avoid repeating the logic.

→ DRY = Don't Repeat Yourself

### Syntax of a Method

A method is a function written inside a class. Since Java is an Object Oriented language, we need to write the method inside some class

```
dataType name () {  
    // Method body  
}
```

Following method returns sum of two numbers

→ Return type

```
int mySum(int a, int b) {  
    int c = a + b;  
    return c;  
}
```

→ Return value

### Calling a Method

A method can be called by creating an object of the class in which the method exists followed by the method call:

```
Calc obj = new Calc(); → Object Creation  
obj.mySum(a, b); → Method call upon an object
```



The values from the method call (a and b) are copied to the a and b of the function `mySum`. Thus even if we modify the values a and b inside the method, the values in the main method will not change.

### Void return type

When we don't want our method to return anything, we use void as the return type.

### Static keyword

Static keyword is used to associate a method of a given class with the class rather than the object. Static method in a class is shared by all the objects.

### Process of method invocation in Java

Consider the method `Sum`:

```
int Sum (int a, int b)
{
    return a+b;
}
```

The method is called like this:

```
Calc obj = new Calc();
c = obj.Sum(2, 3)
```

The values 2 and 3 are copied to a and b and then  $a+b = 2+3 = 5$  is returned in c which is an integer.

Note: In case of Arrays, the reference is passed. Same is the case for object passing to methods.



## Method Overloading

Two or more methods can have same name but different parameters. Such methods are called Overloaded methods.

```
Void foo()
```

```
Void foo(int a)
```

```
int foo(int a, int b)
```

⇒ Overloaded function foo

Method overloading cannot be performed by changing the return type of methods

## Variable Arguments (Varargs)

A function with vararg can be created in Java using the following syntax:

```
public static void foo(int ... arr)
{
```

```
// arr is available here as int[] arr
```

```
}
```

foo can be called with zero or more arguments like this:

```
foo(7)   foo(7, 8, 9)   foo(1, 2, 7, 8, 9)
```

We can also create a function bar like this

```
public static void bar(int a, int arr)
{
```

```
// code
```

```
}
```

↳ Atleast one integer is required now

bar can be called as bar(1), bar(1, 2), bar(1, 7, 9, 11) etc.



## Recursion

A function in Java can call itself. Such calling of function by itself is called recursion.

Example: Factorial of a number

$$\text{factorial}(n) = n * \text{factorial}(n-1) \quad \forall n \geq 1$$

Quick Quiz: Write a program to calculate (recursion must be used) factorial of a number in Java?