



ASIF.I/O

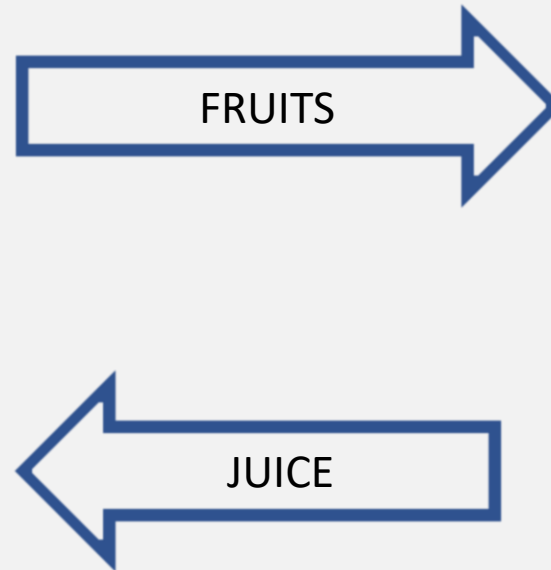
Introduction to Methods

E-mail : asif.io.edu@gmail.com

Methods

Methods/Functions are block of codes which do some specific task and run only when it is called.

Methods/Function are block of codes which so some specific task and run only when called.



Method with Arguments

```
public void eat(){  
    System.out.print("Eating something");  
}
```

```
ob.eat();           //calling the method  
ob.eat();
```

Eating something
Eating something

```
public void eat( String food){  
    System.out.print("Eating "+food);  
}
```

```
ob. eat("Pizza");   //calling the method  
ob. eat("Biryani");
```

Eating Pizza
Eating Biryani

Method with return value

```
public String eat(){  
    System.out.print("Feeling so Hungry");  
    return "Eating something";  
}
```

```
String result = ob.eat();    //calling the method  
System.out.print(result);
```

Feeling so Hungry
Eating something

```
public String eat( String food){  
    System.out.print("Feeling so Hungry");  
    return "Eating "+food;  
}
```

```
String result = ob.eat("Pizza");  
System.out.print(result);
```

Feeling so Hungry
Eating Pizza

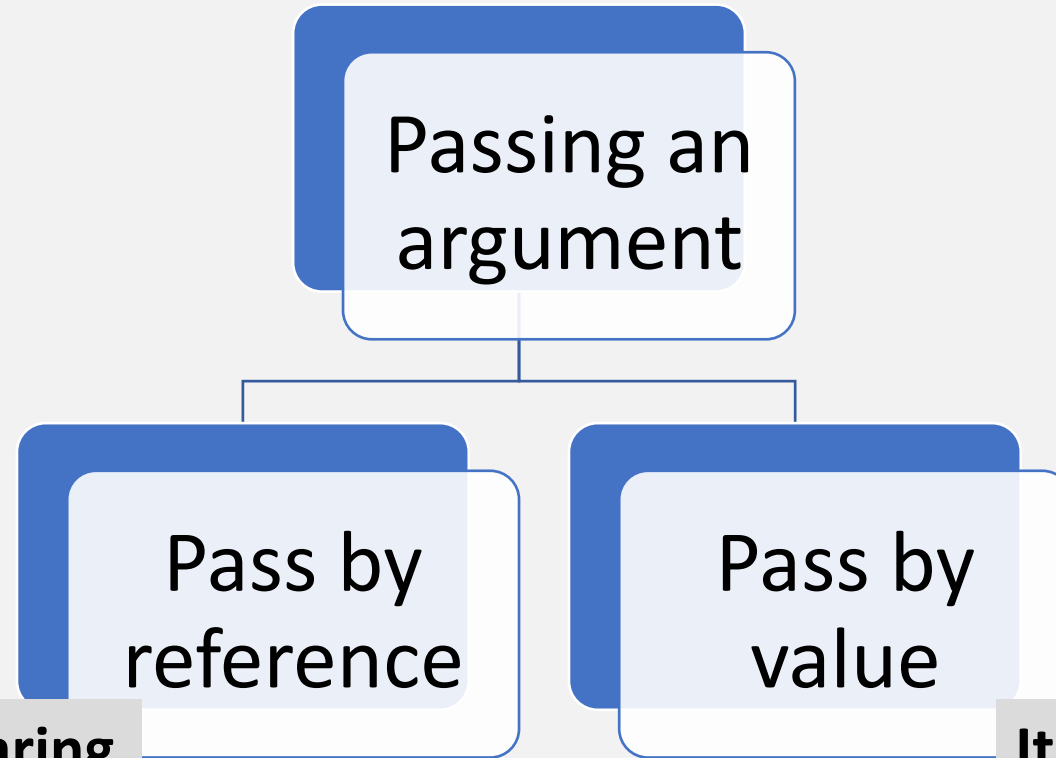
Return Statement

- 👉 return keyword is used to terminate the method.
- 👉 No statement in a method block will be executed after the return statement.
- 👉 return statement can only return a single value from a method.
- 👉 In case of multiple return statement only one return statement will be executed.
- 👉 return type methods are also call pure methods

```
1 public class Demo {
2
3     public static void main(String[] args) {
4         int num = checkMax(a: 12, b: 28);
5         System.out.println("Max number is : "+num);
6     }
7     public static int checkMax(int a, int b){
8         if (a > b)
9             return a;
10        else
11            return b;
12    }
```

C:\Users\mahmad5\.jd
Mam number is : 28
Process finished with...

Passing the argument



It is the process of sharing address of actual parameters with the formal parameters.

It is a process of passing the copy of the actual parameters to the formal parameters.

Code - Pass by value

```
1  ▶ public class Demo {
2
3  ▶  public static void main(String[] args) {
4      int a=20;
5      int b=30;
6      int x=sum(a,b);
7      System.out.println("Addition = "+x);
8      System.out.println("Value of a and b -> "+a+" | "+b);
9
10     }
    1 usage
11     public static int sum(int a, int b){
12         a++;
13         b++;
14         System.out.println("Value of a and b -> "+a+" | "+b);
15         return a+b;
16     }
17 }
```

C:\Users\mahmad5\jdk\liberica-1

Value of a and b -> 21 | 31

Addition = 52

Value of a and b -> 20 | 30

Process finished with exit code 0

Passing the argument

It is the process of sharing address of actual parameters with the formal parameters.

It is a process of passing the copy of the actual parameters to the formal parameters.

pass by reference



`fillCup()`

pass by value



`fillCup()`

Pure vs Impure Functions

A function which **return** a value to its caller and **do not change the state** of object is called **Pure Function**

A function which **may or may not return** a value but **change the state of object** is called **Impure Function**

Function Overloading

Function overloading is the process of defining functions/methods with the same function name but with different **number of parameter** and **type of parameter**.

Code - Overloading

```
3 ▶ public class OverLoading {
4 ▶   public static void main(String[] args) {
5       System.out.println("Volume of cube"+volume(s: 5));
6       System.out.println("Volume of spherical = "+volume(r: 3.0));
7       System.out.println("Volume of cuboid = "+volume(l: 3, b: 4, h: 5));
8   }
9
10      1 usage
11      public static double volume(int s){
12          double result = Math.pow(s,3); //s*s*s
13          return result;
14      }
15      1 usage
16      public static double volume(double r){
17          double result = 4/3 * 3.14 * r*r*r;
18          return result;
19      }
20      1 usage
21      public static double volume(double l,double b, double h){
22          double result = l*b*h;
23          return result;
24      }
25 }
```

Recursion

The process in which a function calls itself directly or indirectly is called **recursion**.

Recursion Code

```
3 ▶ public class Recursion {
4 ▶     public static void main(String[] args) {
5         System.out.println("Go");
6         System.out.println(factorial(n: 4));
7
8     }
    2 usages
9     public static int factorial(int n ){
10         if(n==1)
11             return 1;
12         int p = n * factorial(n: n-1);
13         return p;
14     }
```