

DON BOSCO SCHOOL, RANCHI

COMPUTER APPLICATIONS

CLASS – 10

MORE ABOUT METHODS/FUNCTIONS

What is Pure Function?

A function is a Pure function if it does not changes the state of an object. It takes primitive data type and objects as an arguments but does not modify the objects. It does not changes the values of any variable of objects inside the function.

```
e.g., void display( int x )
      {
          System.out.println("Values of x = " + x );
      }

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

What is Impure Function?

A function is an Impure function if it changes or modifies the state of an object. It changes the variable of an object inside the function.

```
e.g., public void increment( Time time, double secs )
      {
          Time.second = Time.second + secs ;
          Time.minute = Time.minute – 1 ;
          Time.hour = Time.hour + 1 ;
      }
```

Types of Arguments/Parameters to Methods/Functions

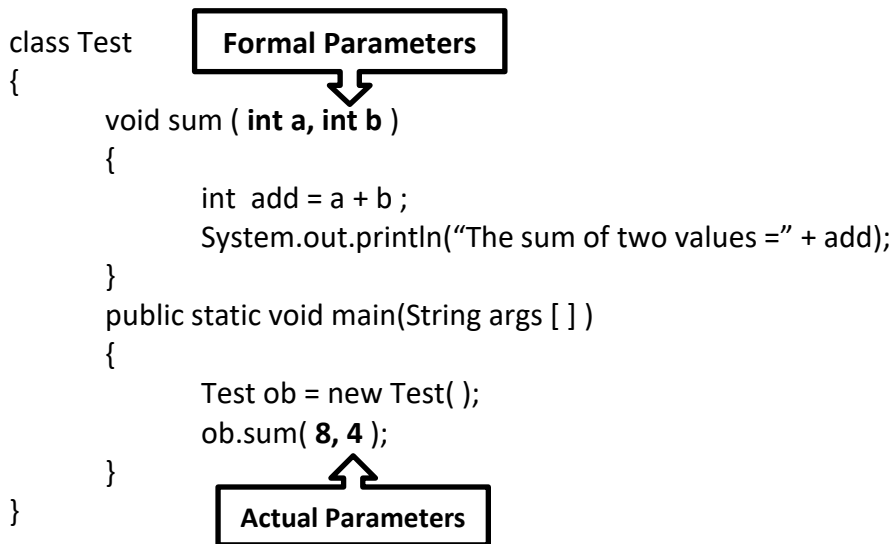
There are two types of arguments or parameters to passed to the functions:

1. **Formal Parameters**
2. **Actual Parameters**

1. Formal Parameters : The parameters or arguments which is passed to the functions at the time of function definition is called as Formal Parameters. It can be of any primitive data types passed to the functions.

2. Actual Parameters : The parameters or arguments which is passed to the functions at the time of actual usage or function invoking is called as Actual Parameters. It can be of any identifiers or variables or any fixed constant values.

The following examples shows the Formal and Actual Parameters :



Types of Methods/Functions Calling

There are two types of invoking or calling a Methods or Functions:

1. **Call by Value or Pass by Value**
2. **Call by Reference or Pass by Reference**

1. Call by Value or Pass by Value: The call by value method copies the values of actual parameters into the formal parameters, i.e., the method creates its own copy of argument and then uses them. If any changes take place in values of argument, then it is not reflected back to the outside of functions.

The main benefit of call by value method is that you cannot alter the variables that are used to call the method. The original copy of the argument value remains intact. In call by value generally primitive data type values is passed.

e.g.,

```

class Test
{
    void increase( int x )
    {
        x = x + 20 ;
        System.out.println(" Inside function x = "+ x );
    }
    public static void main(String args[ ])
    {
        Test ob = new Test( );
        int a = 50;
        System.out.println(" Values of a = "+ a );
        ob.increase( a );
        System.out.println(" Values of a = "+ a );
    }
}

```

Output:

Values of a = 50

Inside function x = 70

Values of a = 50

In above example, we can see the values of a variable do not changes outside the functions.

2. Call by Reference or Pass by Reference: In Call by reference, a reference to the original variable is passed to the function in place of passing a value to the function. This means that in the call by reference method, the called method does not create its own copy of original values, it refers to the original values only. Thus if any changes occurs inside the function then it also reflect back to the outside of functions also.

e.g.,

```
class Test
{
    static int x ;
    void increase( Test m )
    {
        m.x = m.x + 20 ;
        System.out.println(" Inside function x = "+ m.x );
    }
    public static void main(String args[ ])
    {
        Test ob = new Test( );
        ob.x = 50;
        System.out.println(" Values of x = "+ ob.x );
        ob.increase( ob );
        System.out.println(" Values of a = "+ ob.x );
    }
}
```

Output:

Values of a = 50

Inside function x = 70

Values of a = 70

In above example, we can see the values of a variable changes outside the functions also.

What is Function Overloading (Method Overloading)?

The process of defining multiple functions with same name inside a class but difference in number of parameters and their data types is called as Function overloading. Function overloading uses the concepts of polymorphism of OOPs in Java. It uses the same message for different work or process.

e.g.,

```
void area ( int r );
void area ( int l, int b );
void area ( int a, int b, int c );
```

```
void area ( double x, double y );
```

```
void area ( double m );
```

In above example, we can see all the functions having the same name but all having different number of parameters or their data types.

Example Program for Function Overloading:

Question 1 : Write a Java Program to define overloaded functions to calculate the following:

(i) Area of circle.

(area = $3.14 * r * r$)

(ii) Area of rectangle.

(area = length * breadth)

(iii) Area of triangle.

($s = (a+b+c)/2$, area = $\sqrt{s(s-a)(s-b)(s-c)}$)

Answer: Java program with overloaded functions for calculations of area.

```
class Overloaded
```

```
{  
    void area ( int r )  
    {  
        double ar = 3.14 * r * r ;  
        System.out.println("Area of Circle = " + ar);  
    }  
    void area (int l, int b )  
    {  
        int ar = l * b ;  
        System.out.println("Area of Rectangle = " + ar );  
    }  
    void area ( int a, int b, int c )  
    {  
        double s = (a+b+c)/2;  
        double ar = Math.sqrt(s*(s-a)*(s-b)*(s-c));  
        System.out.println("Area of Triangle = " + ar );  
    }  
    public static void main (String args [ ] )  
    {  
        Overloaded ob = new Overloaded( );  
        ob.area( 6 );  
        ob.area( 4, 3 );  
        ob.area( 5, 3, 4 );  
    }  
}
```

Solved the following questions based of functions overloading:

1. Write a Java Program to define overloaded function **volume()** to calculate the volume for the following:
 - a) Volume of Cube [vc = s * s * s]
 - b) Volume of Cuboid [vcd = l * b * h]
 - c) Volume of Sphere [vs = 4/3 * π * r * r * r, where π = 3.14]

Ans: Please refer the link : <https://bit.ly/3geFOqi>

2. Design a class to overload a function **num_calc()** as follows:
 - a) **void num_calc(int num, char ch)** with one integer argument and one character argument, computes the square of integer argument if choice ch is 's' otherwise finds its cube.
 - b) **void num_calc(int a, int b, char ch)** with two integer argument and one character argument. It computes the product of integer arguments if ch is 'p' else adds the integers.
 - c) **void num_calc(Sting s1, String s2)** with two string arguments, which prints whether the strings are equal or not.

Ans: Please refer the link : <https://bit.ly/2zcr62N>

3. Design a class to overload a function **compare()** as follows:
 - a) **void compare(int, int)** - to compare two integer values and print the greater of the two integers.
 - b) **void compare(char, char)** - to compare the numeric value of two characters and print the character with higher numeric value.
 - c) **void compare(String, String)** - to compare the length of the two strings and print the longer of the two.

Ans: Please refer the link : <https://bit.ly/2M2eBtr>

4. Design a class to overload a function **volume()** as follows:
 - a) **double volume(double R)** - with radius (R) as an argument, returns the volume of sphere using the formula.
$$V = \frac{4}{3} \times \frac{22}{7} \times R^3$$
 - b) **double volume(double H, double R)** - with height (H) and radius (R) as the arguments, returns the volume of a cylinder using the formula.
$$V = \frac{22}{7} \times R^2 \times H$$
 - c) **double volume(double L, double B, double H)** - with length (L), breadth (B) and height (H) as the arguments, returns the volume of a cuboid using the formula.
$$V = L \times B \times H$$

Ans: Please refer the link : <https://bit.ly/3d5Sk9C>

5. Design a class to overload a function **area()** as follows:

- a) **double area(double *a*, double *b*, double *c*)** with three double arguments, returns the area of a scalene triangle using the formula:

$$area = \sqrt{s(s-a)(s-b)(s-c)} \text{ where } s = \frac{a+b+c}{2}$$

- b) **double area(int *a*, int *b*, int *height*)** with three integer arguments, returns the area of a trapezium using the formula:

$$area = \frac{1}{2} \times height (a + b)$$

- c) **double area(double *diagonal1*, double *diagonal2*)** with two arguments, returns the area of a rhombus using the formula:

$$area = \frac{1}{2} (diagonal1 \times diagonal2)$$

Ans: Please refer the link : <https://bit.ly/3e7V9qX>