METHOD OVERLOADING

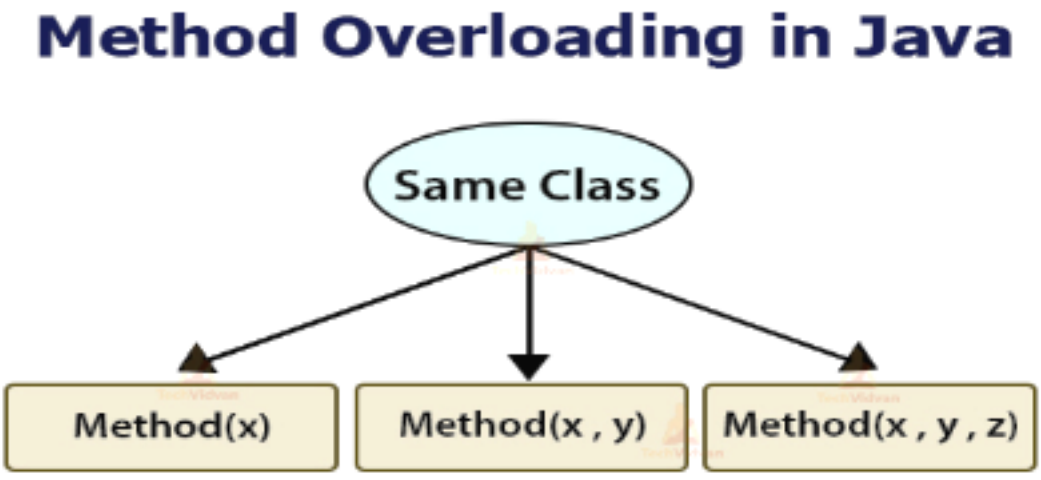
Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different.

NEED OF METHOD OVERLOADING :

Overloading in java is the ability to create multiple methods of the same name, but with different parameters.The main advantage of this is cleanliness of code. This means that if we have any type of variable, we can get a String representation of it by using String. valueOf(variable)

➔ Method overloading is achieved by either:

- ❖ changing the number of arguments.
- ❖ changing the datatype of arguments
- ❖ Function name should be the same
- ❖ **Checking the return type**



syntax:

void func() { ... }
void func(int a) { ... }
float func(double a) { ... }
float func(int a, float b) { ... }
Here, the func() method is overloaded.

Simple Example :

```
Import java.util.Scanner;
class A
{
   void disp(char c)
    {
         System.out.println(c);
    }
   void disp(char c, int num)
    {
         System.out.println(c + " "+num);
    }
}
class Main
{
   public static void main(String args[])
   {
       A a1 = new A();
       a1.disp('a');
       a1.disp('a',10);
   }
}
```

OUTPUT :

```
a
a 10
```

METHODS OF OVERLOADING :

1.Overloading by changing no of argm

```
import java.util.*;
class A
{
```

```
    void display(int a)
     {
       System.out.println("Arg " + a);
     }

    void display(int a, int b)
     {
       System.out.println("Arg "+a+" "+b);
     }
}
class Main
{
   public static void main(String[] args)
    {
        A a1=new A();
        a1.display(1);
        a1.display(1, 4);
    }
}
```

Output:

```
Arg : 1
Arg : 1 4
```

2.overloading by changing datatype of parameters

```
import java.util.*;
class A
{
    // this method accepts int
   void display(int a)
    {
        System.out.println("Got Integer data.");
    }
    // this method  accepts String object
    void display(String a)
    {
        System.out.println("Got String object.");
    }
}
class Main
```

```
{
    public static void main(String[] args)
    {
        A a1=new A();
        a1.display(1);
        a1.display("Hello");
    }
}
```

Output:

```
Got Integer data.
Got String object.
```

Java program to find the sum of two integer number and two float numbers by getting inputs from the user using method overloading.

```
import java.util.Scanner;

class Add

{

   int c;

   float s;

   void sum(int x,int y)

   {

   c=x+y;

   System.out.println("the sum of two integers"+c);

   }

   void sum(float q,float r)

   {

      s=q+r;
```

```
        System.out.println("the sum of two float integers"+s);
    }
}
public class Main
{
  public static void main(String[] args)
 {
    int a,b;
    float a1,b1;
    Add p=new Add();
    Scanner s=new Scanner(System.in);
    System.out.println("enter your two integer numbers");
    a=s.nextInt();
    b=s.nextInt();
    System.out.println("enter your two float numbers");
    a1=s.nextFloat();
    b1=s.nextFloat();
    p.sum(a,b);
    p.sum(a1,b1);
   }
}
```

Program to find the power of a number.Use method overloading.First method have two integer argument and second method have one integer argument and one double argument.

```
class  Power
{
        static int pow(int n,int m)
        {
        int i;
        for(i=1;i<m;i++)
        {
        n=n*n;
        }
        return n;
        }
  void pow(double n,int m)
        {
        int i;
        for(i=1;i<m;i++)
        {
        n=n*n;
        }
        System.out.println(n);
        }
}
class Main
{
public static void main (String[] args) {
        Power p=new Power();
System.out.println(Power.pow(4,2));
        p.pow(11.1,2);
   }
}
```