Experiment No. 4
Implement a program on method and constructor overloading.
Date of Performance:
Date of Submission:

Aim: Implement a program on method and constructor overloading.

Objective: To use concept of method overloading in a java program to create a class with same function name with different number of parameters.

Theory:

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. It is similar to constructor overloading in Java, that allows a class to have more than one constructor having different argument lists.

Example: This example to show how method overloading is done by having different number of parameters for the same method name.

```
Class DisplayOverloading
{
   public void disp(char c)
   {
       System.out.println(c);
   }
   public void disp(char c, int num)
   {
       System.out.println(c + " "+num);
   }
}
Class Sample
{
  Public static void main(String args[])
  {
     DisplayOverloading obj = new DisplayOverloading();
     Obj.disp('a');
     Obj.disp('a',10);
  }
}
```

Output:

A

A 10

Java supports Constructor Overloading in addition to overloading methods. In Java, overloaded constructor is called based on the parameters specified when a new is executed.

Sometimes there is a need of initializing an object in different ways. This can be done using constructor overloading.

For example, the Thread class has 8 types of constructors. If we do not want to specify anything about a thread then we can simply use the default constructor of the Thread class, however, if we need to specify the thread name, then we may call the parameterized constructor of the Thread class with a String args like this:

Thread t= new Thread (" MyThread ");

Code:

1) **CONSTRUCTOR OVERLOADING**

```
class Overload1
{
public static void main(String[] args)
{ System.out.println(Multiplier.mult(2,3));
 System.out.println(Multiplier.mult(2,2,2));
}
}
class Multiplier
{
 static int mult(int a,int b)
{
return a*b;
}
static int mult(int a,int b,int c)
{
return a*b*c;
}
}
```

OUTPUT:

```
C:\Users\Dnyanesh\Desktop\java>java Overload1.java
6
8
```

2) METHOD OVERLOADING

```
class Overload2
{
public static void main(String args[])
{
System.out.println(Adder.add(2,3));
System.out.println(Adder.add(1.2,1.2));
}
}
class Adder
{
static int add(int a,int b)
{
return a+b;
}
static double add(double a,double b)
{return a+b;}
}
```

OUTPUT:

```
C:\Users\Dnyanesh\Desktop\java>javac Overload2.java

C:\Users\Dnyanesh\Desktop\java>java Overload2.java
5
2.4
```

Conclusion:

Comment on how function and constructor overloading used using java
Function and constructor overloading in Java involves creating multiple methods or constructors with the same name within a class but with different parameter lists.

Function Overloading:
Function overloading allows you to define multiple methods in a class with the same name but different parameter lists (number or types of parameters). This way, you can provide different behavior for the same operation depending on the inputs. The choice of which method to call is made at compile-time based on the provided arguments.

Constructor Overloading:

Constructor overloading is similar to function overloading but is used to define multiple constructors within a class with different parameter lists. Constructor overloading enables the creation of objects in various ways, depending on the arguments provided during object instantiation. Like function overloading, constructors can have different parameter types, but the number and types of parameters should differ to distinguish between them.