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. class Loading :
 {
 public static void main(String[] args)
 {
 System.out.println(Overloading.area(5));
 }

}
2. class Loading :
 {
 public static void main(String[] args)
 {
 System.out.println(Overloading.area(5));
 }
}
public class Overloading
 {    static double
area(int r)
 {    return 3.14
* r * r;
}
}
```

Conclusion: function and constructor overloading are powerful features that allow us to define multiple methods or constructors with the same name but different parameters. This enables us to create more flexible and versatile code by providing different ways to interact with our classes and objects. Function overloading allows us to define multiple methods with the same name but different parameter types or numbers, while constructor overloading allows us to define multiple constructors with different parameter lists. By using function and constructor overloading, we can enhance code readability, improve code reusability, and provide more options for users to interact with our Java programs.