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.

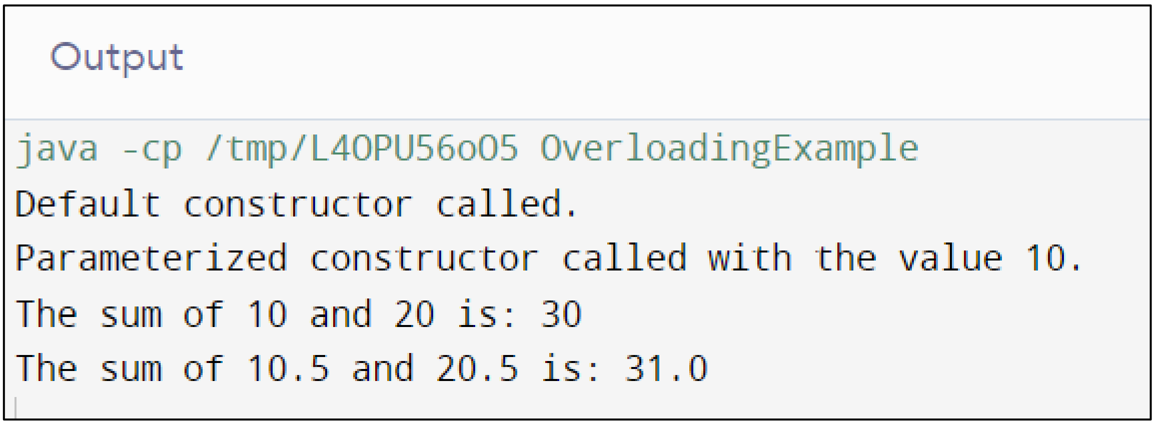
For example, the Thread class has 8 types of constructors. If we do not want to specify anything about a thread then we can simply use the default constructor of the Thread class, however, if we need to specify the thread name, then we may call the parameterized constructor of the Thread class with a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

Thread t= new Thread (" MyThread ");

Code:

```
public class OverloadingExample {

    public OverloadingExample() {
        System.out.println("Default constructor called.");
    }

    public OverloadingExample(int number) {
        this.number = number;
        System.out.println("Parameterized constructor called with the value " + number
+ ".");
    }

    private int number;

    public int add(int number1, int number2) {
        return number1 + number2;
    }

    public double add(double number1, double number2) {
        return number1 + number2;
    }

    public static void main(String[] args) {
        OverloadingExample object = new OverloadingExample();
    OverloadingExample anotherObject = new OverloadingExample(10);

        int sum = object.add(10, 20); // 30
        double doubleSum = object.add(10.5, 20.5); // 31.0

        System.out.println("The sum of 10 and 20 is: " + sum);
        System.out.println("The sum of 10.5 and 20.5 is: " + doubleSum);
      }
    }
```

Output:



```
java -cp /tmp/L4OPU56oO5 OverloadingExample
Default constructor called.
Parameterized constructor called with the value 10.
The sum of 10 and 20 is: 30
The sum of 10.5 and 20.5 is: 31.0
```

Conclusion:

Function and constructor overloading are two powerful features of Java that allow you to write more flexible and reusable code.

Function overloading allows you to define multiple functions with the same name, but with different parameter lists. This allows you to write a single function that can perform multiple tasks, depending on the types and number of arguments that are passed to it.

Constructor overloading allows you to define multiple constructors for a class, each with a different parameter list. This allows you to create objects of the class in different ways, depending on the data that you need to initialize them with.

Both function and constructor overloading can be used to improve the readability, maintainability, and reusability of your code.