EXPERIMENT NO: - 4 **DATE:-**

1.1 Aim: Write a program in JAVA to demonstrate the method and constructor overloading

1.2 Objective:

- To learn Method overloading.
- To learn Constructor Overloading

1.3 Hardware used: Personal Computer capable of running Window XP or above OS, java version 5.0 or greater 1GB free ram , Free disk space 1GB, Processor speed 1GHz or faster**.**

1.4 Software used (if applicable) / Programming Languages Used:

1. Eclipse IDE/Net Beans for Java Developers.
2. Java Development Kit 8 (jdk).
3. Language Used: Java.

1.5 Theory:

Method Overloading

Method overloading results when two or more methods in the same class have the same name but different parameters. Methods with the same name must differ in their types or number of parameters. This allows the compiler to match parameters and choose the correct method when a number of choices exist. Changing just the return type is not enough to overload a method, and will be a compile-time error. They must have a different signature. When no method matching the input parameters is found, the compiler attempts to convert the input parameters to types of greater precision. A match may then be found without error. At compile time, the right implementation is chosen based on the signature of the method call

Below is **an example of a class demonstrating Method Overloading**

```
public class MethodOverloadDemo {
        void sumOfParams() { // First Version
                System.out.println("No parameters");
        }
        void sumOfParams(int a) { // Second Version
                System.out.println("One parameter: " + a);
        }
        int sumOfParams(int a, int b) { // Third Version
                System.out.println("Two parameters: " + a + " , " + b);
                return a + b;
        }
```

```
        double sumOfParams(double a, double b) { // Fourth Version
                System.out.println("Two double parameters: " + a + " , " + b);
                return a + b;
        }
        public static void main(String args[]) {
                MethodOverloadDemo moDemo = new MethodOverloadDemo();
                intResult;
                double doubleResult;
                moDemo.sumOfParams();
                System.out.println();
                moDemo.sumOfParams(2);
                System.out.println();
                intResult = moDemo.sumOfParams(10, 20);
                System.out.println("Sum is  " + intResult);
                System.out.println();
                doubleResult = moDemo.sumOfParams(1.1, 2.2);
                System.out.println("Sum is  " + doubleResult);
                System.out.println();
        }
}
```

In Java, a constructor is just like a method but without return type. It can also be overloaded like Java methods.

Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

```
1.  //Java program to overload constructors
2.  class Student5{
3.      int id;
4.      String name;
5.      int age;
6.      //creating two arg constructor
7.      Student5(int i,String n){
8.      id = i;
9.      name = n;
10.     }
11.     //creating three arg constructor
12.     Student5(int i,String n,int a){
```

```
13.    id = i;
14.    name = n;
15.    age=a;
16.    }
17.    void display(){System.out.println(id+" "+name+" "+age);}
18.
19.    public static void main(String args[]){
20.    Student5 s1 = new Student5(111,"Karan");
21.    Student5 s2 = new Student5(222,"Aryan",25);
22.    s1.display();
23.    s2.display();
24.   }
25. }
```

1.6 Procedure:-

1.7 Observation Table/ Waveforms: --

1.8 Result: Print Out of Program Attached.

1.9 Conclusion: Program in JAVA to demonstrate the method and constructor overloading is implemented successfully.

Constructor Overloading

1. Writing more than 1 constructor in a class with a unique set of arguments is called as Constructor Overloading
2. All constructors will have the name of the class
3. Overloaded constructor will be executed at the time of instantiating an object
4. An overloaded constructor cannot be static as a constructor relates to the creation of an object
5. An overloaded constructor cannot be final as constructor is not derived by subclasses it won't make sense
6. An overloaded constructor can be private to prevent using it for instantiating from outside of the class

Method Overloading

1. Writing more than one method within a class with a unique set of arguments is called as method overloading
2. All methods must share the same name
3. An overloaded method if not static can only be called after instantiating the object as per the requirement
4. Overloaded method can be static, and it can be accessed without creating an object
5. An overloaded method can be final to prevent subclasses to override the method
6. An overloaded method can be private to prevent access to call that method outside of the class

1.10 Specification of Equipment's used (Attached as annexure separately).

1.11 Question:

1. Write a Java Program to Copy the values from one object to another Object using concept of Constructor.

Signature of Staff with Date