



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

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



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

**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);  
    }  
}
```



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

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:**

```
import java.io.*;  
import java.util.*;  
  
class OverLoading2  
{  
    public static void main(String args[])  
    {  
        System.out.println(Adder.add(11,11));  
        System.out.println(Adder.add(12.3,12.6));  
    }  
}  
  
class Adder  
{  
    static int add(int a,int b)  
    {  
        return a+b;  
    }  
}
```



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

```
static double add(double a,double b)
{
    return a+b;
}

}
```

### Conclusion:

Function and constructor overloading are powerful features in Java that promote code efficiency and flexibility. By allowing the same method or constructor name to handle different parameter configurations, Java enables developers to write cleaner, more intuitive code.