



Vidyavardhini's College Of Engineering & Technology

NAME : CHETAN BHUYAL

DATE : 04/08/2025

ROLL NO. : 16

SUBJECT : FULL STACK JAVA

AIM :

Programs on method and constructor overloading

DESCRIPTION :

Method Overloading allows a class to have more than one method with the same name but different **parameter lists** (number, type, or order of parameters). It helps in increasing the readability of the program.

CODE :

```
class Person {  
    String name;  
    int age;  
  
    // Constructor Overloading  
  
    // Default constructor  
    Person() {  
        name = "Unknown";  
        age = 0;  
    }  
  
    // Constructor with one parameter  
    Person(String name) {  
        this.name = name;  
        age = 18;  
    }  
  
    // Constructor with two parameters  
    Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
}
```

```
void greet() {  
    System.out.println("Hello!");  
}
```

```
void greet(String message) {  
    System.out.println("Hello, " + message);  
}
```

```
void greet(String message, int times) {  
    for (int i = 0; i < times; i++) {  
        System.out.println("Hello, " + message);  
    }  
}
```

```
void showDetails() {  
    System.out.println("Name: " + name + ", Age: " + age);  
}
```

```
public class OverloadingExample {  
    public static void main(String[] args) {  
        // Constructor Overloading  
        Person p1 = new Person();  
    }  
}
```

```
Person p1 = new Person();
Person p2 = new Person("Rahul");
Person p3 = new Person("Sneha", 25);

p1.showDetails();
p2.showDetails();
p3.showDetails();

// Method Overloading
p3.greet();
p3.greet("Sneha");
p3.greet("Sneha", 3);
}
```

OUTPUT :

```
Calculator calc = new Calculator();
System.out.println("Add 2 integers: " + calc.add(5, 10));
System.out.println("Add 3 integers: " + calc.add(2, 3, 4));
System.out.println("Add 2 doubles: " + calc.add(4.5, 3.2));
```

CONCLUSION :

Method Overloading and **Constructor Overloading** allow using the **same name** with **different parameters** to improve flexibility and readability.

- **Method Overloading** → Same method name, different parameters (used for performing similar tasks).

- **Constructor Overloading** → Same constructor name, different parameters (used for creating objects in different ways).