

## METHOD OVERLOADING AND METHOD OVERRIDING

### Aim:

To understand and implement method overloading and method overriding.

### PRE LAB EXERCISE

#### QUESTIONS

✓ **What is method overloading?**

Method overloading is having multiple methods in the same class with the same name but different parameters (number or type).

✓ **What is method overriding?**

Method overriding is when a subclass provides a new implementation of a method already defined in its parent class.

✓ **Difference between overloading and overriding.**

Overloading	OVERRIDING
Happens in same class	Happens in parent-child classes
Same method name, different parameters	Same method name, same parameters
Compile-time concept	Runtime concept
No inheritance required	Requires inheritance

## **IN LAB EXERCISE**

### **Objective:**

To demonstrate compile-time and runtime polymorphism.

### **PROGRAMS:**

#### **1.Student Result System (Method Overriding)**

##### **Description:**

- Base class Student has method displayResult().
- Subclasses UGStudent and PGStudent override the method to show different grading systems.

##### **Code :**

```
import java.util.Scanner;

// Base class
class Student {
    String name;

    void displayResult() {
        System.out.println("Student Result");
    }
}

// UG Student subclass
class UGStudent extends Student {
    int marks;

    UGStudent(String n, int m) {
        name = n;
    }
}
```

```

marks = m;
}

@Override
void displayResult() {
    double percentage = (marks / 100.0) * 100;
    System.out.println("UG Student: " + name);
    System.out.println("Marks: " + marks);
    System.out.println("Percentage: " + percentage + "%");
}
}

// PG Student subclass
class PGStudent extends Student {
    double gpa;

    PGStudent(String n, double g) {
        name = n;
        gpa = g;
    }

    @Override
    void displayResult() {
        System.out.println("PG Student: " + name);
        System.out.println("GPA: " + gpa + " / 10");
    }
}

```

**24BCS201**

**J PAVAN KRISHNA**

```
// Main class
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input for UG student
        System.out.print("Enter UG Student Name: ");
        String ugName = sc.nextLine();
        System.out.print("Enter UG Student Marks (out of 100): ");
        int ugMarks = sc.nextInt();
        sc.nextLine(); // consume newline

        // Input for PG student
        System.out.print("Enter PG Student Name: ");
        String pgName = sc.nextLine();
        System.out.print("Enter PG Student GPA (0-10): ");
        double pgGpa = sc.nextDouble();

        // Create objects
        Student s1 = new UGStudent(ugName, ugMarks);
        Student s2 = new PGStudent(pgName, pgGpa);

        System.out.println("\n--- Student Results ---");
        s1.displayResult();
        System.out.println();
        s2.displayResult();
```

**24BCS201**

**J PAVAN KRISHNA**

```
        sc.close();  
    }  
}
```

### **OUTPUT:**

Sample Input:

```
Enter UG Student Name: Ram  
Enter UG Student Marks (out of 100): 85  
Enter PG Student Name: Ravi  
Enter PG Student GPA (0-10): 9.2
```

Output:

```
--- Student Results ---
```

```
UG Student: Ram  
Marks: 85  
Percentage: 85.0%  
PG Student: Ravi  
GPA: 9.2 / 10
```

```
Enter UG Student Name: Rupak  
Enter UG Student Marks (out of 100): 89  
Enter PG Student Name: Krishna  
Enter PG Student GPA (0-10): 9  
  
--- Student Results ---  
UG Student: Rupak  
Marks: 89  
Percentage: 89.0%  
  
PG Student: Krishna  
GPA: 9.0 / 10
```

## **2. Calculator Program (Method Overloading)**

### **Description:**

Create a Calculator class with multiple add() methods to calculate:

- Addition of 2 integers
- Addition of 3 integers
- Addition of 2 double numbers

### **Code:**

```
import java.util.Scanner;

class Calculator {

    int add(int a, int b) {
        return a + b;
    }

    int add(int a, int b, int c) {
        return a + b + c;
    }

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

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Calculator calc = new Calculator();

        System.out.print("Enter two integers: ");
    }
}
```

```

int x = sc.nextInt();
int y = sc.nextInt();
System.out.println("Sum of two integers: " + calc.add(x, y));

System.out.print("Enter three integers: ");
int p = sc.nextInt();
int q = sc.nextInt();
int r = sc.nextInt();
System.out.println("Sum of three integers: " + calc.add(p, q, r));

System.out.print("Enter two decimal numbers: ");
double a = sc.nextDouble();
double b = sc.nextDouble();
System.out.println("Sum of two doubles: " + calc.add(a, b));

sc.close();
}
}

```

**Output:**

**Sample Input:**

```

Enter two integers: 10 20
Enter three integers: 5 10 15
Enter two decimal numbers: 2.5 3.5

```

**Output:**

```

Sum of two integers: 30
Sum of three integers: 30

```

Sum of two doubles: 6.0

```
Enter two integers: 4
4
Sum of two integers: 8
Enter three integers: 3
3
3
Sum of three integers: 9
Enter two decimal numbers: 2.4
43.8
Sum of two doubles: 46.19999999999996
```

## POST LAB EXERCISE

✓ **Is return type important in method overloading and method overriding?**

In overloading, return type alone is not enough to distinguish methods.

In overriding, the return type must be the same (or compatible/covariant).

✓ **Can you overload a method by changing only the return type?**

No. The parameter list must change. Return type alone cannot overload a method.

✓ **Can static methods be overridden? Can they be overloaded?**

Static methods cannot be truly overridden (they are hidden, not overridden).

Yes, static methods can be overloaded.

✓ **Can a method be overridden if the parameter list is different?**

No. If parameters change, it becomes overloading, not overriding.

**Result:**

**24BCS201**

**J PAVAN KRISHNA**

Thus the method overloading and overriding concepts were implemented and executed successfully.

### **ASSESSMENT**

Description	Max Marks	Marks Awarded
Pre Lab Exercise	<b>5</b>	
In Lab Exercise	<b>10</b>	
Post Lab Exercise	<b>5</b>	
Viva	<b>10</b>	
<b>Total</b>	<b>30</b>	
<b>Faculty Signature</b>		

**24BCS201**

**J PAVAN KRISHNA**