Experiment Number : 07 **Date: 05.02.2026**

METHOD OVERLOADING AND METHOD OVERRIDING

Aim:

To understand and implement method overloading and method overriding.

PRE LAB EXERCISE

QUESTIONS

✓ What is method overloading?

Ans: Method overloading means same method name with different parameters in the same class.

✓ What is method overriding?

Ans: Method overriding means child class provides its own version of a parent class method.

✓ Difference between overloading and overriding.

Ans: Overloading: Same method name, different parameters

Overriding: Same method name, same parameters, different class

IN LAB EXERCISE

Objective:

To demonstrate compile-time and runtime polymorphism.

PROGRAMS:

1.Student Result System (Method Overriding)

Description:

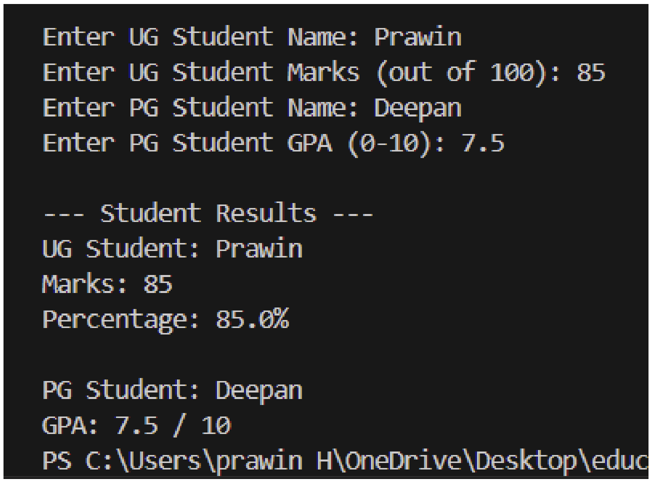
- Base class Student has method displayResult().
- Subclasses UGStudent and PGStudent override the method to show different grading systems.

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");
    }
}

// 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();

        sc.close();
    }
}
```

OUTPUT:



```
Enter UG Student Name: Prawin
Enter UG Student Marks (out of 100): 85
Enter PG Student Name: Deepan
Enter PG Student GPA (0-10): 7.5

--- Student Results ---
UG Student: Prawin
Marks: 85
Percentage: 85.0%

PG Student: Deepan
GPA: 7.5 / 10
PS C:\Users\prawin H\OneDrive\Desktop\educ
```

2. Calculator Program (Method Overloading)

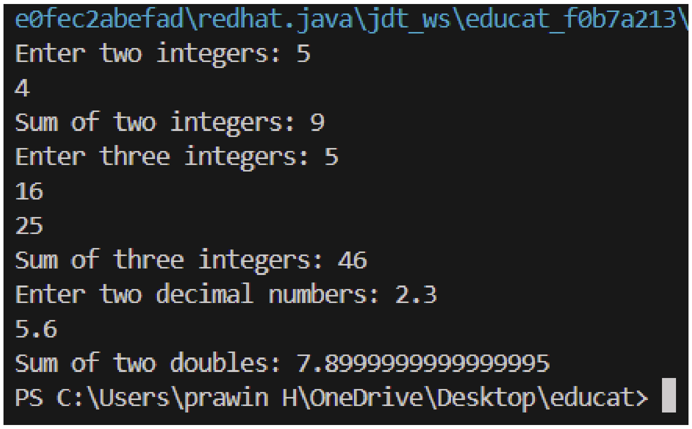
Description:
Create a Calculator class with multiple add() methods to calculate:

- 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:



```
e0fec2abefad\redhat.java\jdt_ws\educat_f0b7a213\
Enter two integers: 5
4
Sum of two integers: 9
Enter three integers: 5
16
25
Sum of three integers: 46
Enter two decimal numbers: 2.3
5.6
Sum of two doubles: 7.8999999999999995
PS C:\Users\prawin H\OneDrive\Desktop\educat>
```

POST LAB EXERCISE

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

Ans: Overloading: Return type alone is not important

Overriding: Return type must be same or compatible

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

Ans: No. Changing only return type does not create overloading.

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

Ans: Static methods cannot be overridden.Static methods can be overloaded

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

Ans: No. To override, the method must have the same parameters.

Result:

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

ASSESSMENT

Description	**Max Marks**	**Marks Awarded**
Pre Lab Exercise	**5**	
In Lab Exercise	**10**	
Post Lab Exercise	**5**	
Viva	**10**	
Total	**30**	
	Faculty Signature	