Task 3: Implement Operator Overloading (Point Class in C++)

Objective:
The objective of this task is to understand and implement operator overloading in C++. Specifically, the + operator is overloaded to enable addition of two Point objects, demonstrating how operators can be customized for user-defined data types.

Problem Statement

Create a Point class with the following requirements:
- Two private data members: x and y coordinates
- Overload the + operator to add two Point objects
- Display the resulting Point after addition

Concepts Used

- Operator Overloading
- Class and Object
- Encapsulation
- Function Overloading

Source Code (C++)

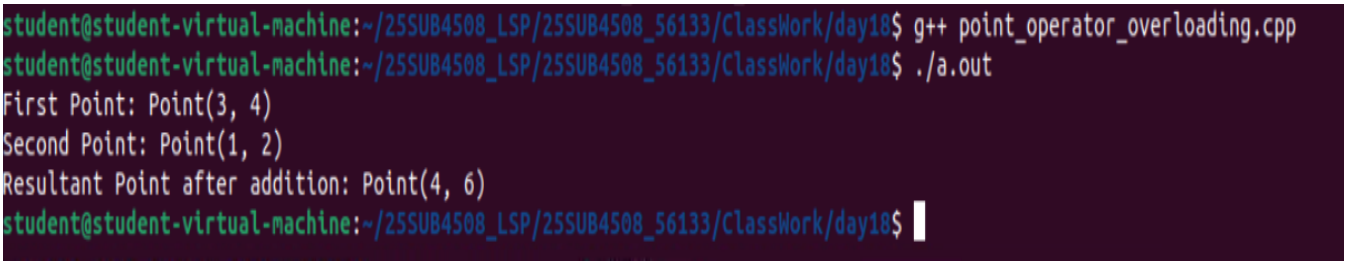
```
#include <iostream>
using namespace std;

class Point {
private:
	int x;
	int y;

public:
	// Constructor
	Point(int xVal = 0, int yVal = 0) {
	x = xVal;
	y = yVal;
```

```
        }

        // Overload + operator
        Point operator+(const Point& p) {
        Point temp;
        temp.x = x + p.x;
        temp.y = y + p.y;
        return temp;
        }

        // Display function
        void display() {
        cout << "Point(" << x << ", " << y << ")" << endl;
        }
};

int main() {
        Point p1(3, 4);
    Point p2(1, 2);

        Point p3 = p1 + p2;

        cout << "First Point: ";
        p1.display();

        cout << "Second Point: ";
        p2.display();

        cout << "Resultant Point after addition: ";
        p3.display();

        return 0;
}
```

Explanation of the Code

1. A class named Point is created with private integer data members x and y.
2. A constructor initializes the x and y values. Default values are provided.
3. The + operator is overloaded using the operator+ function.
4. The operator+ function adds the x and y coordinates of two Point objects.
5. A temporary Point object stores the result and is returned.
6. The display() function prints the Point in coordinate format.
7. In main(), two Point objects are created and added using the overloaded + operator.

Program Output



Conclusion

This task demonstrates operator overloading in C++, allowing intuitive use of operators with user-defined classes. Overloading the + operator for the Point class makes object addition simple, readable, and similar to built-in data types.