



# Institute of Geographical Information Systems

## CS-212 - Object Oriented Programming LAB

Semester: Fall 2025

Class: SCEE-IGIS - 2024

Name: Ali Nawaz

CMS ID : 00000526123

Submitted to: Ma'am Alvina Anjum

Due Date: Nov 26, 2025

### LAB 11: Friends Function

#### Lab Task 1:

Create a class Complex with private real & imaginary parts.

Make a friend function addComplex(Complex, Complex) that returns sum as a new object.

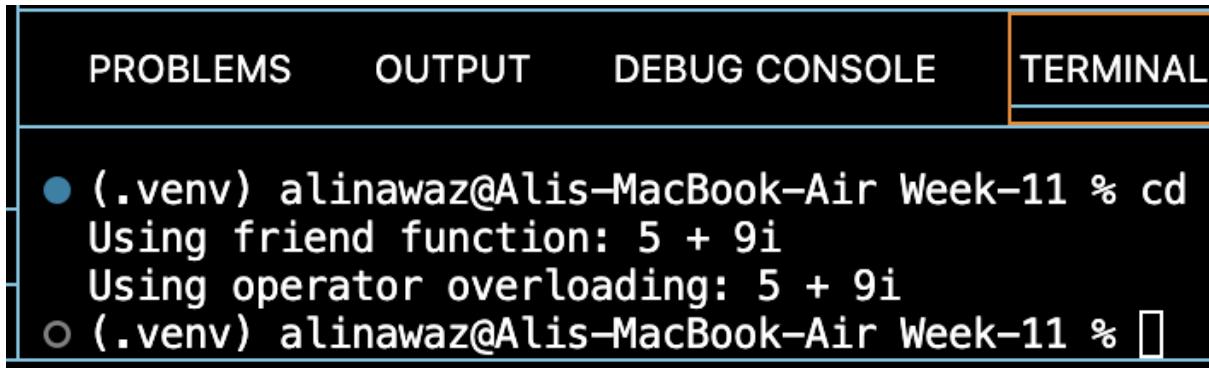
Display result using show().

Extension: overload friend Complex operator+(Complex, Complex);

#### Screenshot:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
# (venv) alinawaz@Aliis-MacBook-Air Week-11 % cd "/Users/alinawaz/Developer/OOP/Week-11/" && g++ Problem1.cpp -o Problem1 && ./Problem1
Using friend function: 5 + 9i
Using operator overloading: 5 + 9i
Using friend function: 5 + 9i
Using operator overloading: 5 + 9i
Code Week...
Code Week...
Ln 40, Col 39 Spaces: 4 UTF-8 LF ⌘ C++ ⌘ Go Live Mac ⌘ Prettier
```

**Output:**



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, displaying the following output:

```
● (.venv) alinawaz@Alis-MacBook-Air Week-11 % cd
Using friend function: 5 + 9i
Using operator overloading: 5 + 9i
○ (.venv) alinawaz@Alis-MacBook-Air Week-11 %
```

**Lab Task 2:**

The company SkyRoute Logistics uses autonomous drones to deliver parcels. Each drone has a battery level (in %) and unique Drone ID. Sometimes, a drone low on battery may request power transfer from another drone mid-route.

Create a class Drone having private variables droneID, batteryLevel

- constructor to initialize
- functions: setBattery(), setDroneID(), getBattery(), displayStatus()

Write a friend function:

```
void transferPower(Drone &donor, Drone &receiver, double percent);
```

This function will reduce battery from donor and add to receiver.

**Demonstration:**

Create two drones, charge one, drain one, then use the friend function to transfer energy.

**Screenshot:**

The screenshot shows a code editor interface with the following details:

- Explorer View:** Shows a project structure with files like `Day-01` through `Day-29`, `calculator.py`, and `notes.py`.
- Editor View:** The current file is `Problem2.cpp`. The code defines a `Drone` class with methods for setting battery level, displaying status, and transferring power between two drones.
- Terminal View:** The terminal output shows the execution of the program. It starts with environment setup (`venv alinawaz`), compilation (`g++ Problem2.cpp -o Problem2`), and running the executable (`./Problem2`). The terminal then displays the initial battery levels of both drones (88% and 102%), followed by a message indicating power was transferred successfully from the donor drone (102%) to the receiver drone (35%).

```
4 class Drone {
5 private:
6     int droneID;
7     double batteryLevel;
8
9 public:
10    Drone(int id, double battery) {
11        droneID = id;
12        batteryLevel = battery;
13    }
14
15    void setBattery(double b) {
16        batteryLevel = b;
17    }
18
19    void setDroneID(int id) {
20        droneID = id;
21    }
22
23    double getBattery() {
24        return batteryLevel;
25    }
26
27    void displayStatus() {
28        cout << "Drone ID: " << droneID << endl;
29        cout << "Battery: " << batteryLevel << "%" << endl;
30    }
31
32    friend void transferPower(Drone &donor, Drone &receiver, double percent);
33 };
34
35 void transferPower(Drone &donor, Drone &receiver, double percent) {
36     if (donor.batteryLevel < percent) {
37         cout << "Not enough battery to transfer!" << endl;
38         return;
39     }
40     donor.batteryLevel -= percent;
41     receiver.batteryLevel += percent;
42 }
43
44 Power transferred successfully!
45
46 After transfer:
47 Drone Id: 101
48 Battery: 65%
49 Drone Id: 102
50 Battery: 35%
```

TERMINAL PORTS

```
[0 .venv alinawaz@Alis-MacBook-Air Week-11 % cd "/Users/alinawaz/Developer/OOP/Week-11/" && g++ Problem2.cpp -o Problem2 && ./Problem2
Drone ID: 101
Battery: 88%
Drone ID: 102
Battery: 102%
Power transferred successfully!
After transfer:
Drone Id: 101
Battery: 65%
Drone Id: 102
Battery: 35%
```

## Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● (.venv) alinawaz@Alis-MacBook-Air Week-11 % cd "/Users/alinawaz"
Before transfer:
Drone ID: 101
Battery: 80%
Drone ID: 102
Battery: 20%

Power transferred successfully!

After transfer:
Drone ID: 101
Battery: 65%
Drone ID: 102
Battery: 35%
○ (.venv) alinawaz@Alis-MacBook-Air Week-11 %
```

## Lab Task 3:

A content creator tracks engagement on two posts — each has likes and comments.

We need to calculate the total combined engagement across both posts.

Create class Post with likes and comments, a constructor and display function

Write a friend function:

Post totalEngagement(Post p1, Post p2);

Return a new Post with summed likes & comments.

Display the final combined reach.

### Screenshot:

```
OOP > Week-11 > Problem3.cpp > main()
4     class Post {
5         private:
6             int likes;
7             int comments;
8
9         public:
10            Post(int l = 0, int c = 0) {
11                likes = l;
12                comments = c;
13            }
14
15            void display() {
16                cout << "Likes: " << likes << endl;
17                cout << "Comments: " << comments << endl;
18            }
19
20            friend Post totalEngagement(Post p1, Post p2);
21        };
22
23    Post totalEngagement(Post p1, Post p2) {
24        int totalLikes = p1.likes + p2.likes;
25        int totalComments = p1.comments + p2.comments;
26
27        return Post(totalLikes, totalComments);
28    }
29
30
31    int main() {
32        Post p1(128, 40);
33        Post p2(300, 90);
34
35        cout << "Post 1:\n";
36        p1.display();
37
38        cout << "Post 2:\n";
39        p2.display();
40
41        Post combined = totalEngagement(p1, p2);
42
43        cout << "\nCombined Engagement:\n";
44        combined.display();
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

The screenshot shows a code editor with the file Problem3.cpp open. The code defines a class Post with private members likes and comments, and a public member function display(). It also contains a friend function totalEngagement() that returns a new Post object with the summed likes and comments of two input Post objects. The main() function creates two Post objects (p1 and p2) and prints their individual and combined engagement statistics using the display() function.

### Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
(.venv) alinawaz@Alis-MacBook-Air Week-11 % cd "/Users/alinawaz/OOP/Week-11/" && g++ Problem3.cpp -o Problem3 && ./Problem3
Post 1:
Likes: 120
Comments: 40
Post 2:
Likes: 300
Comments: 90

Combined Engagement:
Likes: 420
Comments: 130
(.venv) alinawaz@Alis-MacBook-Air Week-11 %
```

The screenshot shows a terminal window displaying the execution of the compiled program. The output shows the individual engagement statistics for two posts (Post 1 and Post 2) and the final combined engagement, which matches the output shown in the code editor's terminal tab.