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: Oct 08, 2025**

LAB 05: Passing Objects as Variables to Functions

Task # 1:

Objective: Understand how object data behaves when passed by value, reference, and pointer.

Steps:

1. Create a class Student with:

```
class Student {
    string name;
    int rollNumber;
    float marks[3];
public:
    Student(string n, int r, float m1, float m2, float m3);
    float totalMarks();
    void display();
    void updateMarks(float m1, float m2, float m3);
};
```

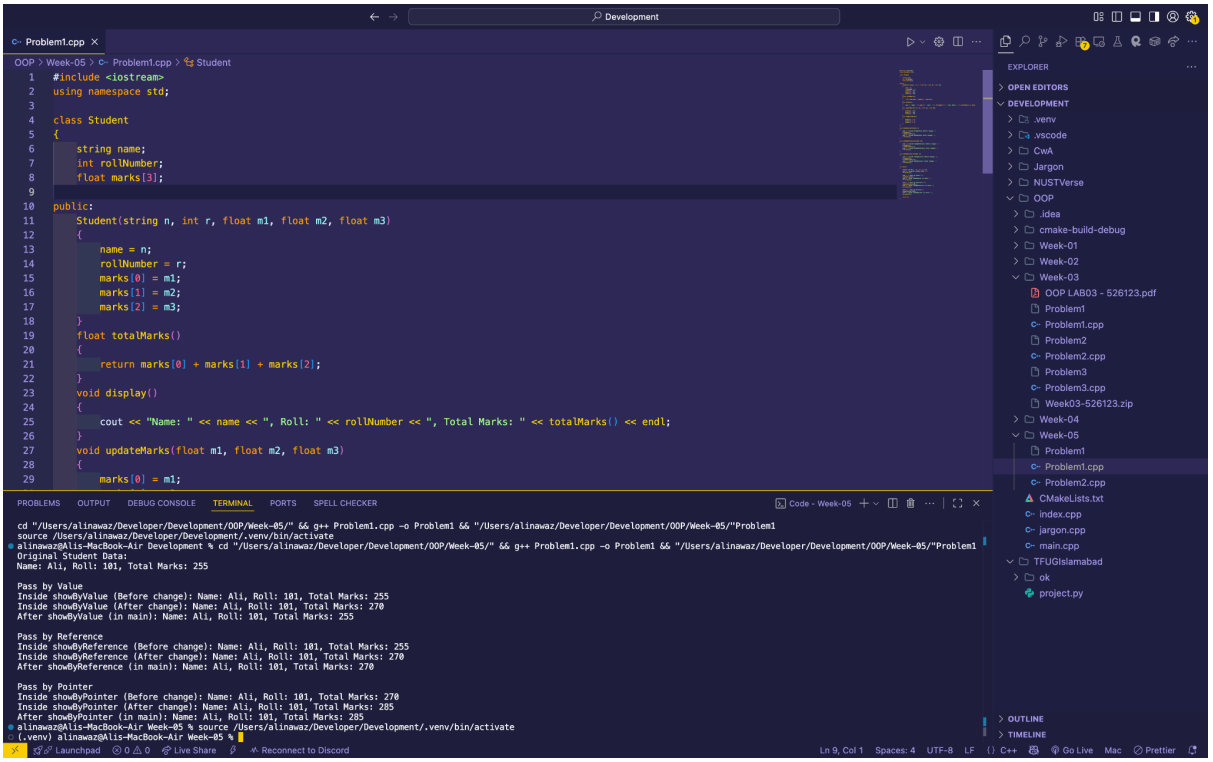
2. Write three functions (outside the class):

```
void showByValue(Student s);
void showByReference(Student &s);
void showByPointer(Student *s);
```

3. In each function, display the student's name and total marks, modify the marks (e.g., increase each by 5), and observe if the changes affect the original object.

4. In main(), create a Student object, call all three functions, and display results after each call.

Screenshot:



Question No 1:

Why did only some functions modify the original marks?

Answer:

Because when we pass the object by value, a copy of the object is made, and changes happen only in that copy, not in the real one.

But when we pass the object by reference or by pointer, the function works on the real object, so changes stay even after the function ends.

Task # 2:

Objective: Extend Task 1 to include more attributes and observe passing behavior for larger objects.

Steps:

1. Extend the Student class by adding:

- int age;
- char gender;
- string address;

2. Add a function:

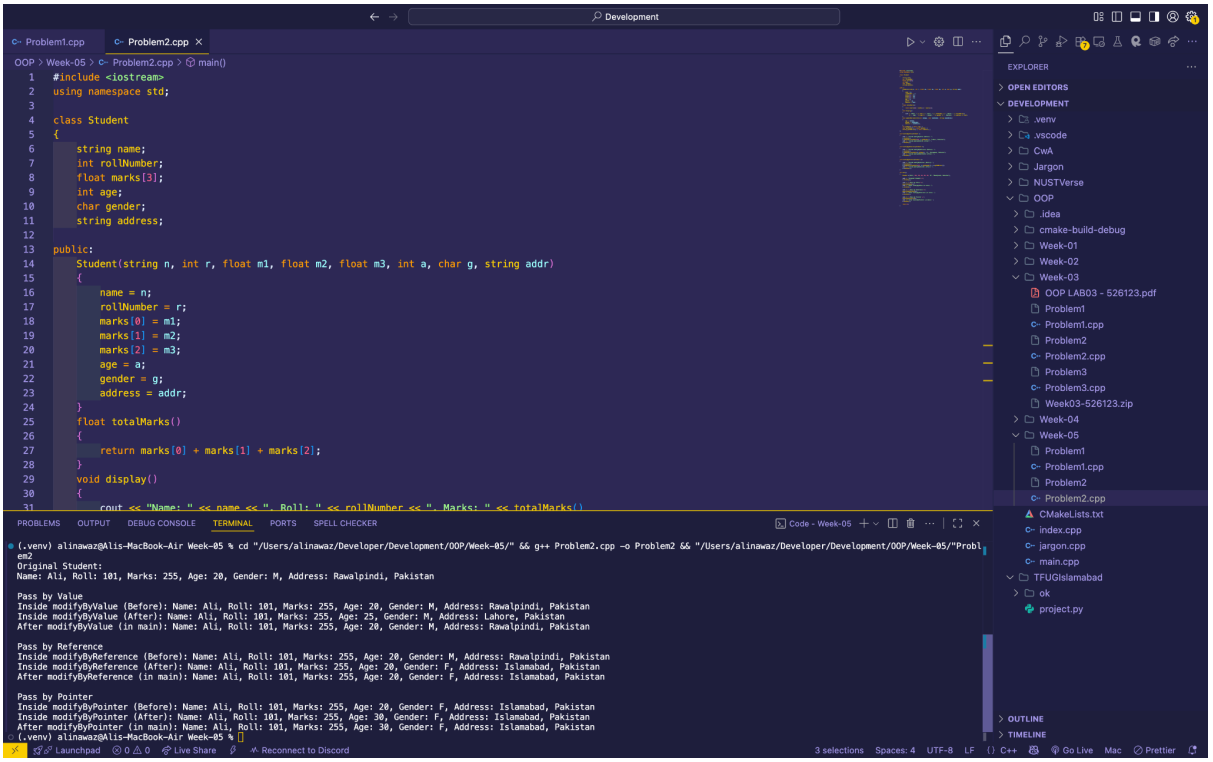
void updatePersonalInfo(int newAge, char newGender, string newAddress);

3. Pass the object:

- By value → modify age and address.
- By reference → change gender and address.
- By pointer → update age only.

4. Observe which changes persist in the original object.

Screenshot:



Question No 2:

Why did only some functions modify the original marks?

Answer:

Passing large objects by value is slow and uses more memory because it makes a full copy.
Passing them by reference or by pointer is faster and saves memory, since no copy is made, only a small link (or address) is shared.

Reflection Questions:

1. **What is the main conceptual difference between pass by reference and pass by pointer?**

Answer:

- Pass by reference directly connects the function to the real object (no need for * or & inside).
- Pass by pointer sends the address of the object, and we use -> or * to access it.

Both can change the real object, but they work a little differently in syntax.

2. **For large objects, which passing method is most memory-efficient and why?**

Answer:

Pass by reference is most memory-efficient because it doesn't make a copy, it only passes a link to the original object.
So, it saves both memory and time.

3. **How does object copying affect performance when using pass-by-value?**

Answer:

When you pass by value, a new copy of the object is made. Copying takes extra time and memory, especially if the object is large.
That's why pass-by-value is less efficient.