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 12, 2025**

LAB 09: Inheritance with Constructors and Multiple Inheritance

Task # 1:

Example Code: Repeated + Hybrid Inheritance

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

// Base class
class Person {
protected:
    char name[50];
public:
    Person(const char* n = "Unknown") {
        strcpy(name, n);
    }
    void displayPerson() { cout << "Name: " << name << endl; }
};

// Derived classes with virtual inheritance to solve diamond problem
class Student : virtual public Person {
protected:
    int rollNo;
public:
    Student(const char* n = "Unknown", int r = 0) : Person(n), rollNo(r) {}
    void displayStudent() { cout << "Roll No: " << rollNo << endl; }
};
```

```
class Employee : virtual public Person {
protected:
   int empID;
public:
   Employee(const char* n = "Unknown", int e = 0) : Person(n), empID(e) {}
   void displayEmployee() { cout << "Employee ID: " << empID << endl; }
};

// Hybrid inheritance
class WorkingStudent : public Student, public Employee {
public:
   WorkingStudent(const char* n, int r, int e)
      : Person(n), Student(n, r), Employee(n, e) {}
   void display() {
      displayPerson();      // only one copy of Person
      displayStudent();
      displayEmployee();
   }
};

int main() {
   WorkingStudent ws("Ali Khan", 101, 5001);
   cout << "Working Student Details:" << endl;
   ws.display();
   return 0;
}
```

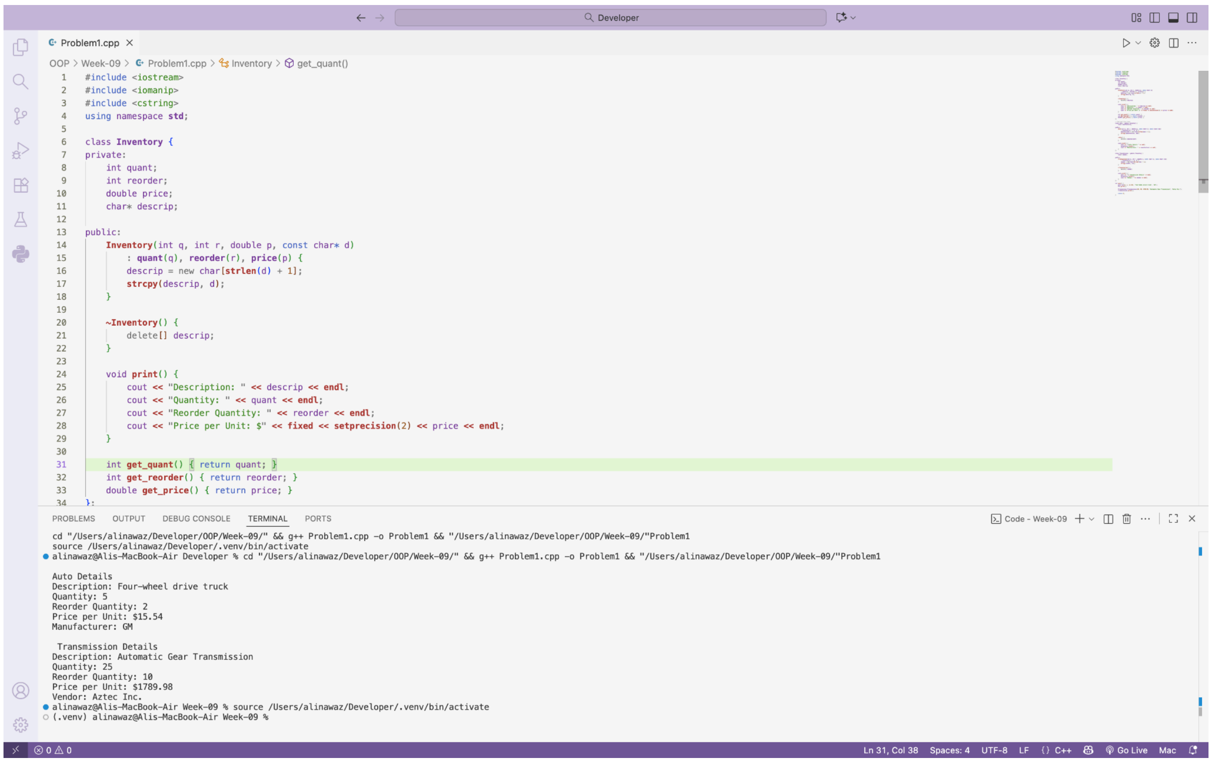
Output:

```
Working Student Details:
Name: Ali Khan
Roll No: 101
Employee ID: 5001
```

Explanation

- Person is inherited virtually to avoid duplicate copies.
- WorkingStudent demonstrates hybrid inheritance:
- Multiple inheritance: Student + Employee
- Multilevel inheritance: Person -> Student -> WorkingStudent

Screenshot:



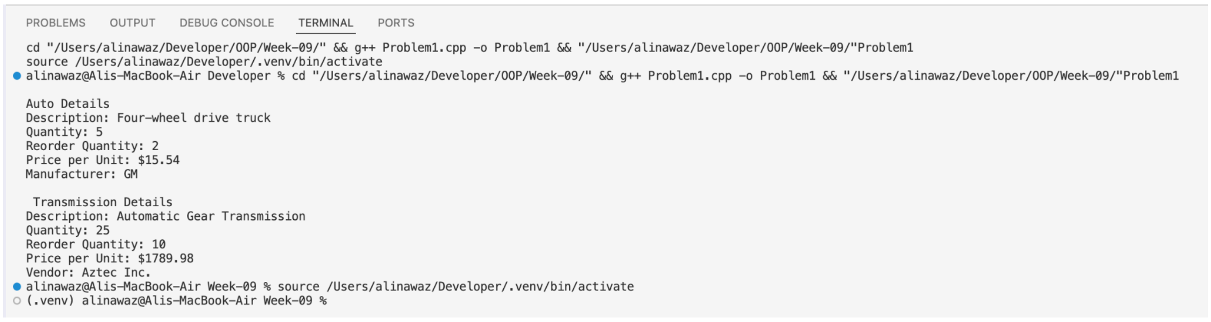
Problem1.cpp

OOP > Week-09 > Problem1.cpp > Inventory > get_quant()

```cpp
#include <iostream>
#include <iomanip>
#include <cstring>
using namespace std;

class Inventory {
private:
    int quant;
    int reorder;
    double price;
    char* descrip;

public:
    Inventory(int q, int r, double p, const char* d)
        : quant(q), reorder(r), price(p) {
        descrip = new char[strlen(d) + 1];
        strcpy(descrip, d);
    }

    ~Inventory() {
        delete[] descrip;
    }

    void print() {
        cout << "Description: " << descrip << endl;
        cout << "Quantity: " << quant << endl;
        cout << "Reorder Quantity: " << reorder << endl;
        cout << "Price per Unit: $" << fixed << setprecision(2) << price << endl;
    }

    int get_quant() { return quant; }
    int get_reorder() { return reorder; }
    double get_price() { return price; }
};
```

```
cd "/Users/alinawaz/Developer/OOP/Week-09/" && g++ Problem1.cpp -o Problem1 && "/Users/alinawaz/Developer/OOP/Week-09/"Problem1
source /Users/alinawaz/Developer/.venv/bin/activate
● alinawaz@Alis-MacBook-Air Developer % cd "/Users/alinawaz/Developer/OOP/Week-09/" && g++ Problem1.cpp -o Problem1 && "/Users/alinawaz/Developer/OOP/Week-09/"Problem1

Auto Details
Description: Four-wheel drive truck
Quantity: 5
Reorder Quantity: 2
Price per Unit: $15.54
Manufacturer: GM

 Transmission Details
Description: Automatic Gear Transmission
Quantity: 25
Reorder Quantity: 10
Price per Unit: $1789.98
Vendor: Aztec Inc.
● alinawaz@Alis-MacBook-Air Week-09 % source /Users/alinawaz/Developer/.venv/bin/activate
○ (.venv) alinawaz@Alis-MacBook-Air Week-09 %
```

Ln 31, Col 38   Spaces: 4   UTF-8   LF   {} C++   Go Live   Mac



```
cd "/Users/alinawaz/Developer/OOP/Week-09/" && g++ Problem1.cpp -o Problem1 && "/Users/alinawaz/Developer/OOP/Week-09/"Problem1
source /Users/alinawaz/Developer/.venv/bin/activate
● alinawaz@Alis-MacBook-Air Developer % cd "/Users/alinawaz/Developer/OOP/Week-09/" && g++ Problem1.cpp -o Problem1 && "/Users/alinawaz/Developer/OOP/Week-09/"Problem1

Auto Details
Description: Four-wheel drive truck
Quantity: 5
Reorder Quantity: 2
Price per Unit: $15.54
Manufacturer: GM

 Transmission Details
Description: Automatic Gear Transmission
Quantity: 25
Reorder Quantity: 10
Price per Unit: $1789.98
Vendor: Aztec Inc.
● alinawaz@Alis-MacBook-Air Week-09 % source /Users/alinawaz/Developer/.venv/bin/activate
○ (.venv) alinawaz@Alis-MacBook-Air Week-09 %
```

Explanation:

- **Base Class (Inventory)** handles item description, quantity, reorder level, and price.
- **Auto** and **Transmission** classes **inherit** Inventory and add their own members (manufacturer, vendor).
- Each derived class calls the base constructor using the **initializer list**.
- Memory is dynamically allocated and properly deallocated using new and delete[].
- Inventory::print() is explicitly called in derived classes to show base details.

Task # 2:

1. **Base Class:** `Person`
- **Data Member:** `name` (string or char array)
- **Function:** `displayPerson()` to display the name

2. **Derived Classes:**
- `Student` (virtually inherits from `Person`)
- **Data Member:** `rollNo`
- **Function:** `displayStudent()`
- `Employee` (virtually inherits from `Person`)
- **Data Member:** `empID`
- **Function:** `displayEmployee()`

3. **Hybrid Derived Class:** `WorkingStudent`
- Inherits from both `Student` and `Employee`
- **Function:** `display()` to display **all details**

Question:

1. Implement **constructors** for all classes to initialize their data members.
2. In `WorkingStudent`, ensure the base `Person` is inherited only once using **virtual inheritance**.
3. Implement `display()` in `WorkingStudent` to print the following:

```
Name: <name>
Roll No: <rollNo>
Employee ID: <empID>
```

4. In `main()`, create at least **one** **`WorkingStudent`** **object** and display its information.
5. Demonstrate **constructor calls** by printing messages in each constructor (optional but recommended for learning).

Example Input/Output

Input:

Name: Ali Khan

Roll No: 101

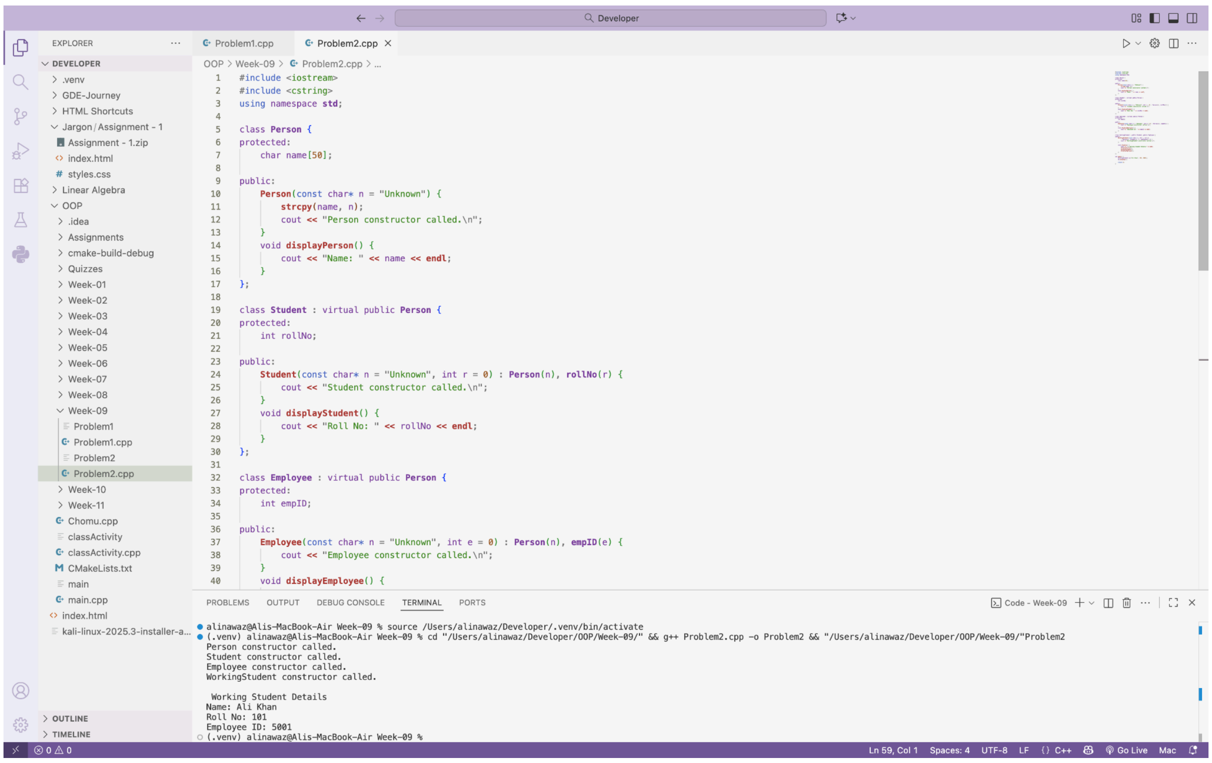
Employee ID: 5001

Output:

```
Working Student Details:
Name: Ali Khan
Roll No: 101
```

```
Employee ID: 5001
```

Screenshot:



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

class Person {
protected:
    char name[50];

public:
    Person(const char* n = "Unknown") {
        strcpy(name, n);
        cout << "Person constructor called.\n";
    }
    void displayPerson() {
        cout << "Name: " << name << endl;
    }
};

class Student : virtual public Person {
protected:
    int rollNo;

public:
    Student(const char* n = "Unknown", int r = 0) : Person(n), rollNo(r) {
        cout << "Student constructor called.\n";
    }
    void displayStudent() {
        cout << "Roll No: " << rollNo << endl;
    }
};

class Employee : virtual public Person {
protected:
    int empID;

public:
    Employee(const char* n = "Unknown", int e = 0) : Person(n), empID(e) {
        cout << "Employee constructor called.\n";
    }
    void displayEmployee() {
```

```
alinawaz@Alis-MacBook-Air Week-09 % source /Users/alinawaz/Developer/.venv/bin/activate
(.venv) alinawaz@Alis-MacBook-Air Week-09 % cd "/Users/alinawaz/Developer/OOP/Week-09/" && g++ Problem2.cpp -o Problem2 && "/Users/alinawaz/Developer/OOP/Week-09/"Problem2
Person constructor called.
Student constructor called.
Employee constructor called.
WorkingStudent constructor called.

 Working Student Details
Name: Ali Khan
Roll No: 101
Employee ID: 5001
(.venv) alinawaz@Alis-MacBook-Air Week-09 %
```

Explanation

- **Virtual inheritance** ensures that only one copy of Person is inherited (resolves diamond problem).
- Student and Employee both inherit Person, and WorkingStudent inherits both — showing **hybrid inheritance** (multiple + multilevel).
- Constructors print messages to visualize constructor call order.
- display() function combines data from all inherited classes.