24csu350

Assingment:-03

Inheritance and Relationship

```
#include <iostream>
#include <vector>
using namespace std;
// Base class
class Person {
protected:
string name;
int age;
public:
Person(string n, int a) : name(n), age(a) {}
virtual void displayDetails() {
cout << "Name: " << name << ", Age: " << age << endl; }
virtual double calculatePayment() { return 0.0; } };
// Student and subclasses
class Student : public Person {
public:
Student(string n, int a) : Person(n, a) {}
void displayDetails() override {
cout << "Student - "; Person::displayDetails(); }
};class UndergraduateStudent : public Student {
string major;
```

```
public:
UndergraduateStudent(string n, int a, string m) : Student(n, a), major(m) {} void displayDetails() override
{
Student::displayDetails();
cout << "Major: " << major << endl;
}
};
// Professor and subclasses
class Professor : public Person {
protected:
int yearsOfService;
public:
Professor(string n, int a, int y) : Person(n, a), yearsOfService(y) {} };
class AssistantProfessor : public Professor {
public:
AssistantProfessor(string n, int a, int y) : Professor(n, a, y) {} double calculatePayment() override {
return 40000 + 1000 * yearsOfService;
}
void displayDetails() override {
cout << "Assistant Professor - "; Person::displayDetails(); }
};
// Aggregation: Course has 1 Professor class Course {string title;
Professor* instructor;
public:
Course(string t, Professor* p) : title(t), instructor(p) {}
void showCourseInfo() {
```

```
cout << "Course: " << title << endl;

instructor->displayDetails();

}

};

int main() {

UndergraduateStudent u("Alice", 20, "Computer Science"); AssistantProfessor p("Dr. Bob", 45, 5);

Course c("Intro to C++", &p);

u.displayDetails();

cout << endl;

c.showCourseInfo();

cout << "Professor Salary: " << p.calculatePayment() << endl; return 0;
```