

24csu350

Assingment 2:-

```
#include <iostream>

#include <string>

using namespace std;

// Base class

class Person {

protected:

string name;

int age;

public:

Person(string n, int a) {

setName(n);

setAge(a);

}

void setName(string n) {

if (!n.empty()) name = n;

}

void setAge(int a) {

if (a > 0 && a < 120) age = a;

}

virtual void displayDetails() {

cout << "Name: " << name << ", Age: " << age << endl; }

virtual double calculatePayment() {return 0;
```

```

}

virtual ~Person() {}

};

// Derived class

class Student : public Person {

double GPA;

public:

Student(string n, int a, double g) : Person(n, a) { setGPA(g);

}

void setGPA(double g) {

if (g >= 0.0 && g <= 4.0) GPA = g;

}

void displayDetails() override {

Person::displayDetails();

cout << "Role: Student, GPA: " << GPA << endl; }

double calculatePayment() override { return 20000;

}

};

// Derived class

class Professor : public Person { public:Professor(string n, int a) : Person(n, a) {}

void displayDetails() override {

Person::displayDetails();

cout << "Role: Professor" << endl;

}

double calculatePayment() override { return 50000;

```

```
}  
  
};  
  
// Test of polymorphism  
  
int main() {  
  
    Person* people[2];  
  
    people[0] = new Student("Alice", 20, 3.8); people[1] = new Professor("Dr. Bob", 45);  
  
    for (int i = 0; i < 2; ++i) { people[i]->displayDetails();  
  
        cout << "Payment: " << people[i]->calculatePayment() << endl << endl; delete people[i];  
  
    }  
  
    return 0;  
  
}
```