Inheritance(14-08-24)

Author: ThanhTH10

Date: 14/08/2024

1.Design 3 classes:Student ,Exam and Result.
The student class has data members such as representing id,name,email and address.
create class exam by inheriting from student class and it has data members representing marks scored in six subjects.
Derive the result class from class exam and it should have data member totalmarks.
write a program to model this releationship. what type of inheritance this model belongs to?

```
#include <iostream>
using namespace std;
class Student
{
protected:
    string id;
    string name;
    string email;
    string address;

public:
    Student(string _id, string _name, string _email, string _address) : id(_id), name(_name),
email(_email), address(_address) {}
    void display()
    {
        cout << "Student" << endl;
        cout << "ID: " << id << "\t|Name: " << name << "\t|Email: " << email << "\t|Address: "
<< address << endl;
    }
};
class Exam : public Student
{
protected:
    double math;
    double science;
    double english;
    double hindi;
    double social;
    double computer;
public:
    Exam(string _id, string _name, string _email, string _address, double _math, double
_science, double _english, double _hindi, double _social, double _computer)
        : Student(_id, _name, _email, _address), math(_math), science(_science),
english(_english), hindi(_hindi), social(_social), computer(_computer) {}
    void display()
    {
        Student::display();
        cout << "\nMark report" << endl;
        cout << "Math: " << math << "\t|Science: " << science << "\t|Englist: " << english <<
"\t|Hindi: " << hindi << "\t|Social: " << social << "\t|Computer: " << computer << endl;
    }
};
class Result : public Exam
```

```
{
    double totalMarks;
    double avg;
public:
    Result(string _id, string _name, string _email, string _address, double _math, double
_science, double _english, double _hindi, double _social, double _computer)
        : Exam(_id, _name, _email, _address, _math, _science, _english, _hindi, _social,
_computer)
    {
        totalMarks = math + science + english + hindi + social + computer;
        avg = totalMarks / 6;
    }
    void display()
    {
        Exam::display();
        cout << "\nTotal mark: " << totalMarks << "\t|AVG mark: " << avg << endl;
    }
};
int main()
{
    Result result("CE160136", "John Doe", "john@example.com", "123 Main St", 90, 85, 95, 80,
75, 90);
    result.display();
    return 0;
}
```

Output:

```
Student
ID: CE160136    |Name: John Doe |Email: john@example.com        |Address: 123 Main St

Mark report
Math: 90        |Science: 85    |Englist: 95    |Hindi: 80      |Social: 75     |Computer: 90

Total mark: 515 |AVG mark: 85.8333
```

Type of inheritance: Mutil-level inheritance

---

2. extend the inheritance disucssed in the above pgm such that **result** class also inherits properties of **sport** class.**sport** class consist of data member grade. **sport** class is a derived class of **student** class.write a program to model this releationship such that members of the students class are not inherited twice. what type of inheritance this model belongs to?

```
#include <iostream>
using namespace std;
class Student
{
protected:
    string id;
    string name;
    string email;
    string address;

public:
    Student(string _id, string _name, string _email, string _address)
        : id(_id), name(_name), email(_email), address(_address) {}
    void display()
```

```
    {
        cout << "Student" << endl;
        cout << "ID: " << id << "\t|Name: " << name << "\t|Email: " << email << "\t|Address: "
<< address << endl;
    }
};
class Exam : virtual public Student
{
protected:
    double math;
    double science;
    double english;
    double hindi;
    double social;
    double computer;
public:
    Exam(string _id, string _name, string _email, string _address,
         double _math, double _science, double _english, double _hindi, double _social, double
_computer)
        : Student(_id, _name, _email, _address),
          math(_math), science(_science), english(_english), hindi(_hindi), social(_social),
computer(_computer) {}
    void display()
    {
        cout << "\nMark report" << endl;
        cout << "Math: " << math << "\t|Science: " << science << "\t|Englist: " << english <<
"\t|Hindi: " << hindi << "\t|Social: " << social << "\t|Computer: " << computer << endl;
    }
};
// Sport class
class Sport : virtual public Student
{
    string grade;
public:
    Sport(string _id, string _name, string _email, string _address, string _grade)
        : Student(_id, _name, _email, _address), grade(_grade) {}
    void display()
    {
        cout << "\nGrade: " << grade << endl;
    }
};
class Result : public Exam, public Sport
{
    double totalMarks;
    double avg;
public:
    Result(string _id, string _name, string _email, string _address,
           double _math, double _science, double _english, double _hindi, double _social,
double _computer,
           string _grade)
        : Student(_id, _name, _email, _address),
          Exam(_id, _name, _email, _address, _math, _science, _english, _hindi, _social,
_computer),
          Sport(_id, _name, _email, _address, _grade)
    {
        totalMarks = math + science + english + hindi + social + computer;
        avg = totalMarks / 6;
    }
    void display()
    {
        Student::display();
```

```
        Exam::display();
        Sport::display();
        cout << "\nTotal mark: " << totalMarks << "\t|AVG mark: " << avg << endl;
    }
};
int main()
{
    Result result("CE160136", "John Doe", "john@example.com", "123 Main St", 90, 85, 95, 80,
75, 90, "A");
    result.display();
    return 0;
}
```

Output:

```
Student
ID: CE160136    |Name: John Doe |Email: john@example.com        |Address: 123 Main St

Mark report
Math: 90        |Science: 85    |Englist: 95    |Hindi: 80      |Social: 75     |Computer: 90

Grade: A

Total mark: 515 |AVG mark: 85.8333
```

Type of inheritance: Mutilpath inheritance

3.It is required to find out the cost of constructing a house.Create a base class called **House**.There are two class called **door** and **window**.The house class has members which provide information releated to the area of construction,Door,Windows.It gives the responsibility of computing the cost of doors and windows construction to Door and Window classes respectively.write an interactive program to model the above releationship.

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

class House
{
protected:
    int numberOfUnit;
    double unitPrice;
public:
    House(int nou, double up) : numberOfUnit(nou), unitPrice(up) {}
    virtual double totalPrice() = 0;
};
class Door : public House
{
public:
    Door(int nod, double pod) : House(nod, pod) {}
    double totalPrice() override
    {
        return numberOfUnit * unitPrice;
    }
};
class Window : public House
{
```

```
public:
    Window(int now, double pow) : House(now, pow) {}
    double totalPrice() override
    {
        return numberOfUnit * unitPrice;
    }
};
int main()
{
    Door door(2, 100.0);
    Window window(4, 50.0);
    cout << "Total price of door: " << door.totalPrice() << endl;
    cout << "Total price of window: " << window.totalPrice() << endl;
    cout << "Sum: " << door.totalPrice() + window.totalPrice() << endl;
    return 0;
}
```

Option 2 (I don't clear about the question, so that I do 2 version here)

```
#include <iostream>
using namespace std;
class ConstructionElement
{
protected:
    int numberOfUnit;
    double costPerOne;

public:
    ConstructionElement(int nou, double cpo) : numberOfUnit(nou), costPerOne(cpo) {}
    virtual int calculateCost()
    {
        return numberOfUnit * costPerOne;
    }
};
class Door : public ConstructionElement
{
public:
    Door(int door_num, double door_cost) : ConstructionElement(door_num, door_cost) {}
};
class Window : public ConstructionElement
{
public:
    Window(int win_num, double win_cost) : ConstructionElement(win_num, win_cost) {}
};
class House
{
private:
    int area;
    Door door;
    Window window;
public:
    House(int a, int door_num, double door_cost, int win_num, double win_cost)
        : area(a), door(door_num, door_cost), window(win_num, win_cost) {}
    int calculateTotalCost()
    {
        return area * 100 + door.calculateCost() + window.calculateCost();
    }
    void displayCost()
    {
        cout << "Total cost of constructing the house: $" << calculateTotalCost() << endl;
    }
```

```
};
int main()
{
    int area, door_num, door_cost, win_num, win_cost;
    cout << "Enter the area of the house: ";
    cin >> area;
    cout << "Enter the number and price per unit of the door: ";
    cin >> door_num >> door_cost;
    cout << "Enter the number and price per unit of the window: ";
    cin >> win_num >> win_cost;
    House house(area, door_num, door_cost, win_num, win_cost);
    house.displayCost();
    return 0;
}
```