

/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

Design a class named PersonData with the following member variables:

FirstName

LastName

Address

City

State

Zip

PhoneNo.

Write the appropriate accessor (getter) and mutator (setter) functions for these member variables. Next, design a class named CustomerData, which is derived from the PersonData class. The CustomerData class should have the following member variables:

Customerid

MailingList

The Customerid variable will be used to hold a integer number for each customer. The MailingList variable should be a bool. It will be set to true if the customer wishes to be on or false if the customer does not wish to be on a mailing list.

Write appropriate Accessor and Mutator functions for these member variables. CustomerData class will have the InputCustomerData()member function which will Input all the data for customer. DisplayCustomerData()member function which will display all the data for customer.

Demonstrate an object of the CustomerData class in a simple program.

*/

```
#include<iostream>
```

```
using namespace std;
```

```
class PersonData
```

```
{
```

```
protected:
```

```
    string fn, ln, add, city, state;
```

```
    int zip;
```

```
    long long phno;
```

```
public:
```

```
    void accessor()
```

```
{
```

```
    cout<<"Enter first name: ";
```

```
    getline(cin, fn);
```

```

        cout<<"Enter last name: ";
        getline(cin, ln);
        cout<<"Enter address: ";
        getline(cin, add);
        cout<<"Enter city: ";
        getline(cin, city);
        cout<<"Enter state: ";
        getline(cin, state);
        cout<<"Enter zip: ";
        cin>>zip;
        cout<<"Enter phone number: ";
        cin>>phno;
    }
    void mutator()
    {
        cout<<"First name: "<<fn<<endl;
        cout<<"Last name: "<<ln<<endl;
        cout<<"Address: "<<add<<endl;
        cout<<"City: "<<city<<endl;
        cout<<"State: "<<state<<endl;
        cout<<"Zip: "<<zip<<endl;
        cout<<"Phone Number: "<<phno<<endl;
    }
};

class CustomerData: public PersonData
{
protected:
    int customerid;
    bool mailinglist;
public:
    void inputCustomerData()
    {
        accessor();
        cin.ignore();
        cout<<"Enter Customer ID: ";
    }
};

```

```

        cin>>customerid;

        cin.ignore();

        cout<<"Do you want Mailing List Service (Enter 1 for Yes or 0 for No): ";

        cin>>mailinglist;
    }

    void displayCustomerData()
    {
        mutator();

        cout<<"Customer ID: "<<customerid<<endl;

        cout<<"Mailing List Service: ";

        if(mailinglist)
            cout<<"Yes\n";

        else
            cout<<"No\n";

    }

};

int main()
{
    CustomerData obj;

    obj.inputCustomerData();

    obj.displayCustomerData();

    return 0;

}

```

*****OUTPUT*****

Enter first name: Deepanshu

Enter last name: Gupta

Enter address: P1-A

Enter city: Delhi

Enter state: Shahdara

Enter zip: 110032

Enter phone number: 8130819546

Enter Customer ID: 101

Do you want Mailing List Service (Enter 1 for Yes or 0 for No): 1

First name: Deepanshu

Last name: Gupta

Address: P1-A

City: Delhi

State: Shahdara

Zip: 110032

Phone Number: 8130819546

Customer ID: 101

Mailing List Service: Yes

/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

2. A retail store has a preferred customer plan where customers may earn discounts on all their purchases. The amount of a customer's discount is determined by the amount of the customer's purchases in the store.

- When a preferred customer spends 5000, he or she gets a 5% discount on all future purchases.
- When a preferred customer spends 10,000, he or she gets a 6% discount on all future purchases.
- When a preferred customer spends 15,000, he or she gets a 7% discount on all future purchases.
- When a preferred customer spends 20,000 or more, he or she gets a 10% discount on all future purchases.

Design a class named PreferredCustomer, which is derived from the CustomerData class which is having following attributes:

FirstName

LastName

Address

City

State

Zip

PhoneNo.

Customer_id

The PreferredCustomer class should have the following member variables:

purchasesAmount (a double)

discountLevel (a double)

The purchasesAmount variable holds the total of a customer's purchases. The discountLevel variable should be set to the correct discount percentage, according to the store's preferred customer plan. Write appropriate member functions for this class and demonstrate it in a simple program.

*/

```
#include<iostream>
```

```
using namespace std;
```

```
class CustomerData
```

```
{
```

```
protected:
```

```
    string fn, ln, add, city, state;
```

```
    int zip, customerid;
```

```
    long long phno;
```

```
public:
```

```
    void getCustomerData()
```

```
{
```

```
    cout<<"Enter first name: ";
```

```
    getline(cin, fn);
```

```

        cout<<"Enter last name: ";
        getline(cin, ln);
        cout<<"Enter address: ";
        getline(cin, add);
        cout<<"Enter city: ";
        getline(cin, city);
        cout<<"Enter state: ";
        getline(cin, state);
        cout<<"Enter zip: ";
        cin>>zip;
        cout<<"Enter phone number: ";
        cin>>phno;
        cout<<"Enter Customer ID: ";
        cin>>customerid;
    }
    void displayCustomerData()
    {
        cout<<"First name: "<<fn<<endl;
        cout<<"Last name: "<<ln<<endl;
        cout<<"Address: "<<add<<endl;
        cout<<"City: "<<city<<endl;
        cout<<"State: "<<state<<endl;
        cout<<"Zip: "<<zip<<endl;
        cout<<"Phone Number: "<<phno<<endl;
        cout<<"Customer ID: "<<customerid<<endl;
    }
};

class PreferredCustomer: public CustomerData
{
protected:
    int purchasesAmount, discountLevel;
public:
    void getData()
    {
        getCustomerData();
    }
};

```

```

        cout<<"Enter total amount of purchase: ";
        cin>>purchasesAmount;
        if (purchasesAmount>=20000)
            discountLevel=10;
        else if (purchasesAmount>=15000)
            discountLevel=7;
        else if (purchasesAmount>=10000)
            discountLevel=6;
        else if (purchasesAmount>=5000)
            discountLevel=5;
        else discountLevel=0;
    }
    void showData()
    {
        displayCustomerData();
        cout<<"Purchase Amount: "<<purchasesAmount<<endl;
        cout<<"Discount Percentage: "<<discountLevel<<"%\n";
    }
};

int main()
{
    PreferredCustomer obj;
    obj.getData();
    obj.showData();
    return 0;
}

```

*****OUTPUT*****

Enter first name: Deepanshu

Enter last name: Gupta

Enter address: P1-A

Enter city: Delhi

Enter state: Shahdara

Enter zip: 110032

Enter phone number: 8130819546

Enter Customer ID: 102

Enter total amount of purchase: 12500

First name: Deepanshu

Last name: Gupta

Address: P1-A

City: Delhi

State: Shahdara

Zip: 110032

Phone Number: 8130819546

Customer ID: 102

Purchase Amount: 12500

Discount Percentage: 6%

/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

3. What do you mean by diamond problem in C++ and how you will resolve this problem.

SOLUTION: When we inherit more than one base class in the same derived class and all these base classes also inherit another but same single class (super parent), multiple references of the super parent class become available to the derived class. So, it becomes unclear to the derived class, which version of the super parent class it should refer to. Virtual inheritance in C++ is a type of inheritance that ensures that only one copy or instance of the base class's members is inherited by the grandchild derived class. It is implemented by prefixing the virtual keyword in the inheritance statement.

*/

```
#include<iostream>

using namespace std;

class A
{
protected:
    int a;
public:
    void setA()
    {
        cout<<"Enter A = ";
        cin>>a;
    }
    void showA()
    {
        cout<<"A = "<<a<<endl;
    }
};

class B:public virtual A
{
protected:
    int b;
public:
    void setB()
    {
```

```

        cout<<"Enter B = ";
        cin>>b;
    }
    void showB()
    {
        cout<<"B = "<<b<<endl;
    }
};

class C:public virtual A
{
    protected:
        int c;
    public:
        void setC()
        {
            cout<<"Enter C = ";
            cin>>c;
        }
        void showC()
        {
            cout<<"C = "<<c<<endl;
        }
};

class D:public B, public C
{
    protected:
        int d;
    public:
        void setD()
        {
            cout<<"Enter D = ";
            cin>>d;
        }
        void showD()
        {

```

```
        cout<<"D = "<<d<<endl;
    }
};

int main()
{
    D obj;
    obj.setA();
    obj.setB();
    obj.setC();
    obj.setD();
    obj.showA();
    obj.showB();
    obj.showC();
    obj.showD();
    return 0;
}
```

*****OUTPUT*****

Enter A = 10

Enter B = 20

Enter C = 30

Enter D = 40

A = 10

B = 20

C = 30

D = 40

/*

Name : Deepanshu Gupta

Section : AI & ML

Roll Number : 10

4. With the help of a program explain the order of execution of constructor in multiple inheritance.

SOLUTION: The constructor of the derived class calls the constructors of the base classes in the same order in which they are specified in the header of the derived class. The syntax define constructors in multiple inheritance with arguments is also similar to the definition of single inheritance with arguments. The constructors of the base classes are connected with the constructor of the derived class by using colon (:) and separated by commas.

*/

```
#include<iostream>

using namespace std;

class Base
{
    public:
        Base()
        {
            cout<<"Base Default Constructor\n";
        }
};

class Derived: public Base
{
    public:
        Derived()
        {
            cout<<"Derived Default Constructor\n";
        }
        Derived(int x)
        {
            cout<<"Derived Parameterized Constructor\n";
        }
};

int main()
{
```

```
Base b;  
Derived d1;  
Derived d2(10);  
return 0;  
}
```

*****OUTPUT*****

Base Default Constructor

Base Default Constructor

Derived Default Constructor

Base Default Constructor

Derived Parameterized Constructor