

## **(MULTIPLE INHERITANCE+AGGREGATION)**

**(MADE BY ALI AKBER)**

**(BSCS 2ND SS1)**

### **//ASSIGNMENT QUESTION.**

### **//MULTIPLE INHERITANCE.**

```
#include<iostream>

#include<string.h>

using namespace std;

class Empolyee{

    private:

    string name;

    int id;

    public:

    Empolyee():name(""),id(0){}          //no arg constructor.

    Empolyee(int i,string na):name(na),id(i){}    //two arg constructor.

    void getdata(){

        cout<<"Enter Empolyee name:"<<endl;

        getline(cin,name);

        cout<<"Enter Empolyee id:"<<endl;

        cin>>id;

        cin.ignore();

    }

    void showdata(){

        cout<<"Empolyee name is:"<<name<<endl;

        cout<<"Empolyee id is:"<<id<<endl;

    }

}
```

```

};

class Student{
    private:
        string university;
        string degree;
    public:
        Student():university(""),degree({});
        Student(string uni,string deg):university(uni),degree(deg){};
        void getedu(){
            cout<<"Enter the University in which student read:"<<endl;
            getline(cin,university);
            cin.ignore();
            cout<<"Enter the Degree earned by the student:"<<endl;
            getline(cin,degree);
            cin.ignore();
        }
        void showedu(){
            cout<<"University In which student read is:"<<university<<endl;
            cout<<"Degree which the student earned is:"<<degree<<endl;
        }
};

```

```

class Manager: private Empolyee,private Student{
    private:
        string title;
        double dues;
    public:

```

```

    Manager():Empolyee(),Student(),title(""),dues(0){}    //no arg constructor.
Manager(int i,string na,string uni,string deg,string tit,double d)

:Empolyee(i,na),Student(uni,deg),title(tit),dues(d){}    //multiple arg constructor.

void getdata(){

    Empolyee::getdata();

    Student::getedu();

    cout<<"Enter Manager title: "<<endl;

    getline(cin,title);

    cin.ignore();

    cout<<"Enter Manager dues:"<<endl;

    cin>>dues;

    cin.ignore();

}

void showdata(){

    Empolyee::showdata();

    Student::showedu();

    cout<<" Manager title is:"<<title<<endl;

    cout<<"Manager dues is:"<<dues<<endl;

}

};

class Scientist: private Empolyee,private Student{

private:

int publications;

public:

    Scientist():Empolyee(),Student(),publications(0){}    //no arg constructor.

Scientist(int i,string na,string uni,string deg,int pub)

```

```

:Empolyee(i,na),Student(uni,deg),publications(pub){}    //multiple arg constructor.

void getdata(){

    Empolyee::getdata();

    Student::getedu();

    cout<<"Enter Scientist Publications : "<<endl;

    cin>>publications;

    cin.ignore();

}

void showdata(){

    Empolyee::showdata();

    Student::showedu();

    cout<<"Scientist Publications are:"<<publications<<endl;

}

};

class Laborer: public Empolyee{

};

int main(){

    Manager m;

    Scientist s1,s2;

    Laborer l;

    cout<<"Enter data of the manager:"<<endl;

    m.getdata();

    cout<<endl;

    cout<<"Manager data is as follows:"<<endl;

    m.showdata();

    cout<<endl;

```

```
cout<<"Enter data of the 1st scientist : "<<endl;
```

```
s1.getdata();
```

```
cout<<endl;
```

```
cout<<"1st scientist data is as follows:"<<endl;
```

```
s1.showdata();
```

```
cout<<endl;
```

```
cout<<"Enter data of the 2nd scientist : "<<endl;
```

```
s2.getdata();
```

```
cout<<endl;
```

```
cout<<"2nd scientist data is as follows:"<<endl;
```

```
s2.showdata();
```

```
cout<<endl;
```

```
cout<<"Enter data of the laborer:"<<endl;
```

```
l.getdata();
```

```
cout<<endl;
```

```
cout<<"Laborer data is as follows:"<<endl;
```

```
l.showdata();
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

## **//ASSIGNMENT QUESTION 02**

### **//MULTIPLE INHERITANCE**

```
#include<iostream>

#include<string.h>

using namespace std;

class Type{

    private:

    char dimensions[50];

    char grade[50];

    public:

    Type():dimensions(""),grade(""){}    //no arg constructor.

    Type(char d[],char g[]){

        strcpy(dimensions,d);

        strcpy(grade,g);    }    //two arg constructor.

    void gettype(){

        cout<<"Enter Dimensions of the lumber:"<<endl;

        cin.getline(dimensions,50);

        cin.ignore();

        cout<<"Enter Grade of the lumber:"<<endl;

        cin.getline(grade,50);

        cin.ignore();

    }

    void showtype()const{

        cout<<"Dimensions of the lumber is:"<<dimensions<<endl;

        cout<<"Grade of the lumber is:"<<grade<<endl;

    }

}
```

```

};

class Distance{
private:
    int feets;
    float inches;
public:
    Distance():feets(0),inches(0){}    //no arg constructor.
    Distance(int fe,float inc):feets(fe),inches(inc){}    //two arg constructor.
    void getdist(){
        cout<<"Enter feets :"<<endl;
        cin>>feets;
        cout<<"Enter inches :"<<endl;
        cin>>inches;
    }
    void showdist()const{
        cout<<"Feets are :"<<feets<<endl;
        cout<<"Inches are:"<<inches<<endl;
    }
};

class Lumber:public Type,public Distance{
private:
    int quantity;
    double price;
public:
    Lumber():Type(),Distance(),quantity(0),price(0){}    //no arg constructor.
    Lumber(char d[],char g[],int fe,float inc,int q,double p)

```

```

:Type(d,g),Distance(fe, inc),quantity(q),price(p){}    //multiple arg constructor.

void getLumber(){
    Type::gettype();
    Distance::getdist();
    cout<<"Enter the quantity of lumber required to be purchase:"<<endl;
    cin>>quantity;
    cout<<"Enter the price of lumber purchased:"<<endl;
    cin>>price;
}

void showLumber()const{
    Type::showtype();
    Distance::showdist();
    cout<<"Quantity of the lumber to be purchase is:"<<quantity<<endl;
    cout<<"Price of the lumber purchased is:"<<price*quantity<<endl;
}

};

int main(){
    Lumber construction;
    construction.getLumber();
    cout<<endl;
    cout<<"Details of The Construction is as follows"<<endl;
    construction.showLumber();
    cout<<endl;
    cout<<"Details of The Timber is as follows"<<endl;
    Lumber Timber( "6×2","rough" , 4, 0.0, 300, 4.45 );
    Timber.showLumber();
}

```



```
return 0;
```

```
}
```

### **//AGGREGATION QUESTION.**

```
#include<iostream>
```

```
#include<string.h>
```

```
using namespace std;
```

```
class Empolyee{
```

```
    private:
```

```
    string name;
```

```
    int id;
```

```
    public:
```

```
    Empolyee():name(""),id(0){}          //no arg constructor.
```

```
    Empolyee(int i,string na):name(na),id(i){}    //two arg constructor.
```

```
    void getdata(){
```

```
        cout<<"Enter Empolyee name:"<<endl;
```

```
        getline(cin,name);
```

```
        cout<<"Enter Empolyee id:"<<endl;
```

```
        cin>>id;
```

```
        cin.ignore();
```

```
    }
```

```
    void showdata(){
```

```
        cout<<"Empolyee name is:"<<name<<endl;
```

```
        cout<<"Empolyee id is:"<<id<<endl;
```

```
    }
```

```
};
```

```

class Student{

    private:

    string university;

    string degree;

    public:

    Student():university(""),degree({});

    Student(string uni,string deg):university(uni),degree(deg){};

    void getedu(){

        cout<<"Enter the University in which student read:"<<endl;

        getline(cin,university);

        cin.ignore();

        cout<<"Enter the Degree earned by the student:"<<endl;

        getline(cin,degree);

        cin.ignore();

    }

    void showedu(){

        cout<<"University In which student read is:"<<university<<endl;

        cout<<"Degree which the student earned is:"<<degree<<endl;

    }

};

class Manager{

    private:

    string title;

    double dues;

    Empolyee emp;

    Student stu;

```

```

public:
void getdata(){
    emp.getdata();
    stu.getedu();
    cout<<"Enter Manager title: "<<endl;
    getline(cin,title);
    cin.ignore();
    cout<<"Enter Manager dues:"<<endl;
    cin>>dues;
    cin.ignore();
}

void showdata(){
    emp.showdata();
    stu.showedu();
    cout<<" Manager title is:"<<title<<endl;
    cout<<"Manager dues is:"<<dues<<endl;
}

};

class Scientist: private Empolyee,private Student{
private:
    int publications;
Empolyee emp;
Student stu;
public:
void getdata(){
    emp.getdata();

```

```

        stu.getedu();

        cout<<"Enter Scientist Publications : "<<endl;

        cin>>publications;

        cin.ignore();
    }

    void showdata(){

        emp.showdata();

        stu.showedu();

        cout<<"Scientist Publications are:"<<publications<<endl;

    }

};

class Laborer{

    private:

        Empolyee emp;

    public:

    void getdata(){

        emp.getdata();

    }

    void showdata(){

        emp.showdata();

    }

};

int main(){

    Manager m;

    Scientist s1,s2;

    Laborer l;

```

```
cout<<"Enter data of the manager:"<<endl;

m.getdata();

cout<<endl;

cout<<"Manager data is as follows:"<<endl;

m.showdata();

cout<<endl;
```

```
cout<<"Enter data of the 1st scientist :"<<endl;

s1.getdata();

cout<<endl;

cout<<"1st scientist data is as follows:"<<endl;

s1.showdata();

cout<<endl;
```

```
cout<<"Enter data of the 2nd scientist :"<<endl;

s2.getdata();

cout<<endl;

cout<<"2nd scientist data is as follows:"<<endl;

s2.showdata();

cout<<endl;
```

```
cout<<"Enter data of the laborer:"<<endl;

l.getdata();

cout<<endl;

cout<<"Laborer data is as follows:"<<endl;

l.showdata();
```

```
cout<<endl;
```

```
return 0;
```

```
}
```