

DHAKA UNIVERSITY OF ENGINEERING & TECHNOLOGY, GAZIPUR



Department of Computer Science and Engineering

Course No.: **CSE-2112**

Course Title: **Object Oriented Programming Language Sessional**

Lab No: 05

Lab Name: Inheritances Program Problem and Solution.

Experiment Date: **20-04-2021**

Submission Date: **22-04-2021**

Submitted To:

Mr.Md. Omor Farque

Associative Professor, Department of CSE

Mss. Sabah Binte Noor

Associative Professor, Department of CSE

Submitted By-

Name: **Mehedi Hasan Shuvo**

Student Id.: 194016

Year: 2nd

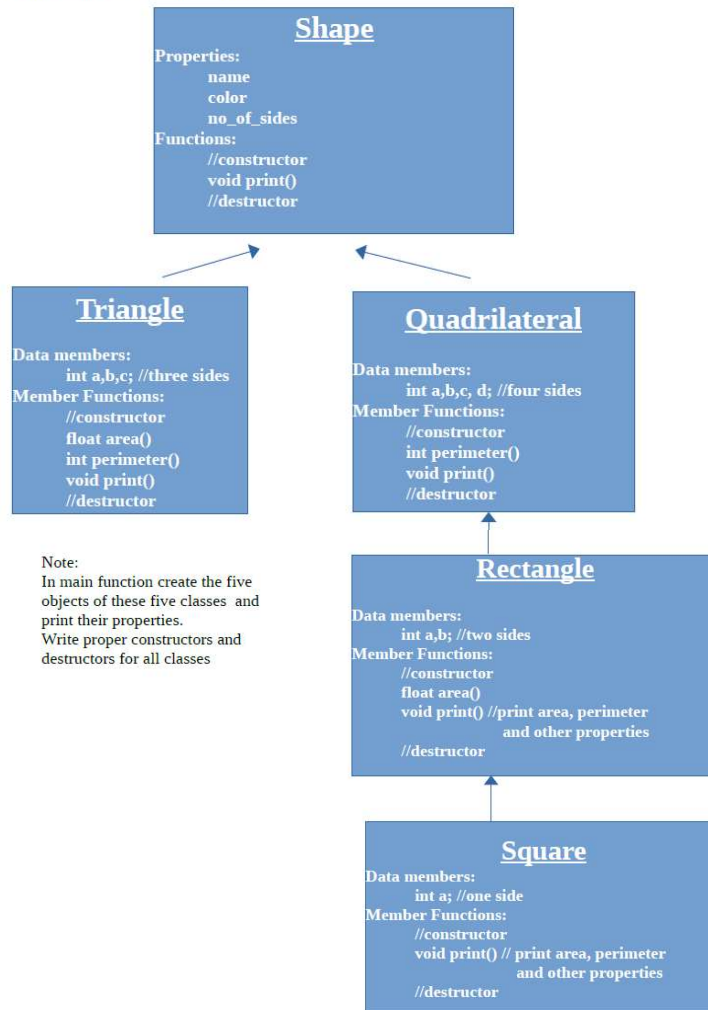
Semester: 1st

Session: 2019 - 2020

Problems No: 01

Problem Title:

1. Implement the following inheritance.



Solution:

```
#include<bits/stdc++.h>
using namespace std;

class Shape
{
private:
    string name;
    string color;
    int noOfSides;
public:
    Shape()
    {
        cout<<"Default Constructor Call of Shape Class";
    }
    Shape(string sName,string sColor,int sNoOfSides)
    {
        name=sName;
```

```

        color=sColor;
        noOfSides=sNoOfSides;
        cout<<"calling Default constructor of Shape class\n";
    }

    void display()
    {
        cout<<"Name is: "<<name<<" Color is: "<<color<<" No of Sides: "<<noOfSides<<endl;
    }
    ~Shape()
    {
        cout<<"Destructor Called of Shape Class\n";
    }

};

class Triangle:public Shape
{
private:
    int a;
    int b;
    int c;
public:
    Triangle()
    {
        cout<<"calling Default constructor of Triangle class\n";
    }
    Triangle(int tA,int tB,int tC,string name,string color,int noOfSides):Shape(name,color,noOfSides)
    {
        a=tA;
        b=tB;
        c=tC;
        cout<<"calling constructor of Triangle class\n";
    }

    float area()
    {
        if (a < 0 || b < 0 || c < 0 ||(a + b <= c) || a + c <= b ||b + c <= a)
        {
            cout << "Not a valid trianglen";
            exit(0);
        }
        float s = (a + b + c) / 2;
        return sqrt(s * (s - a) *(s - b) * (s - c));
    }

    int perimeter()
    {
        return (a + b + c) / 2;
    }

    void display()
    {
        Shape::display(); //scope resolution function
        cout<<"Area of Triangle: "<<area()<<endl;
        cout<<"Perimeter of Triangle: "<<perimeter()<<endl;
    }

```

```

~Triangle()
{
    cout<<"Triangle Destractor Callled"<<endl;
}
};

class Quadrilateral:public Shape
{
private:
    int a;
    int b;
    int c;
    int d;
public:
    Quadrilateral()
    {
        cout<<"calling Default constructor of Quadrilateral class\n";
    }
    Quadrilateral(int tA,int tB,int tC,int tD,string name,string color,int
noOfSides):Shape(name,color,noOfSides)
    {
        a=tA;
        b=tB;
        c=tC;
        d=tD;
        cout<<"calling constructor of Quadrilateral class\n";
    }

    int perimeter()
    {

        return (a+b+c+d);
    }

    void display()
    {
        Shape::display(); //scope resolution function
        cout<<"Primeter of Quadrilateral: "<<perimeter()<<endl;
    }

    ~Quadrilateral()
    {
        cout<<"Quadrilateral Destractor Callled"<<endl;
    }
};

class Rectangle:public Shape
{
private:
    int a;
    int b;
public:
    Rectangle()
    {
        cout<<"calling Default constructor of Rectangle class\n";
    }
    Rectangle(int tA,int tB,string name,string color,int noOfSides):Shape(name,color,noOfSides)
    {

```

```

    a=tA;
    b=tB;
    cout<<"calling constructor of Rectangle class\n";
}

float area()
{
    return (a*b);
}

int perimeter()
{
    return 2*(a+b);
}

float diagonal(){
    return sqrt((a*a)+(b*b));
}

void display()
{
    Shape::display(); //scope resolution function
    cout<<"Area of Rectangle: "<<area()<<endl;
    cout<<"Perimeter of Rectangle: "<<perimeter()<<endl;
    cout<<"Diagonal of Rectangle: "<<diagonal()<<endl;
}

~Rectangle()
{
    cout<<"Rectangle Destractor Callled"<<endl;
}
};

class Square:public Shape
{
private:
    int a;
public:
    Square()
    {
        cout<<"calling Default constructor of Square class\n";
    }
    Square(int tA,string name,string color,int noOfSides):Shape(name,color,noOfSides)
    {
        a=tA;
        cout<<"calling constructor of Square class\n";
    }

    float area()
    {
        return (a*a);
    }
    int perimeter()
    {
        return 4*a;
    }
}

```

```

float diagonal(){
    return a*sqrt(2);
}

void display()
{
    Shape::display(); //scope resolution function
    cout<<"Area of Square: "<<area()<<endl;
    cout<<"Perimeter of Square: "<<perimeter()<<endl;
    cout<<"Diagonal of Square: "<<diagonal()<<endl;
}

~Square()
{
    cout<<"Square Destractor Callled"<<endl;
}
};

int main()
{
    Shape ss1=Shape("Shape","None",0);
    ss1.display();
    cout<<endl;

    Triangle t1=Triangle(3,4,5,"Triangle","Red",3);
    t1.display();
    cout<<endl;

    Quadrilateral q1=Quadrilateral(3,4,5,6,"Quadrilateral","Green",4);
    q1.display();
    cout<<endl;

    Rectangle r1=Rectangle(3,4,"Rectangle","Blue",2);
    r1.display();
    cout<<endl;

    Square s1=Square(3,"Square","Black",4);
    s1.display();
    cout<<endl;

    return 0;
}

```

Output:

"C:\Users\shuvo\OneDrive - duet.ac.bd\DUET\CSE2-1\Lab\Programming\Lab 05\problem 01.exe"

```
calling Default constructor of Shape class
Name is: Shape Color is: None No of Sides: 0

calling Default constructor of Shape class
calling constructor of Triangle class
Name is: Triangle Color is: Red No of Sides: 3
Area of Triangle: 6
Perimeter of Triangle: 6

calling Default constructor of Shape class
calling constructor of Quadrilateral class
Name is: Quadrilateral Color is: Green No of Sides: 4
Primeter of Quadrilateral: 18

calling Default constructor of Shape class
calling constructor of Rectangle class
Name is: Rectangle Color is: Blue No of Sides: 2
Area of Rectangle: 12
Perimeter of Rectangle: 14
Diagonal of Rectangle: 5

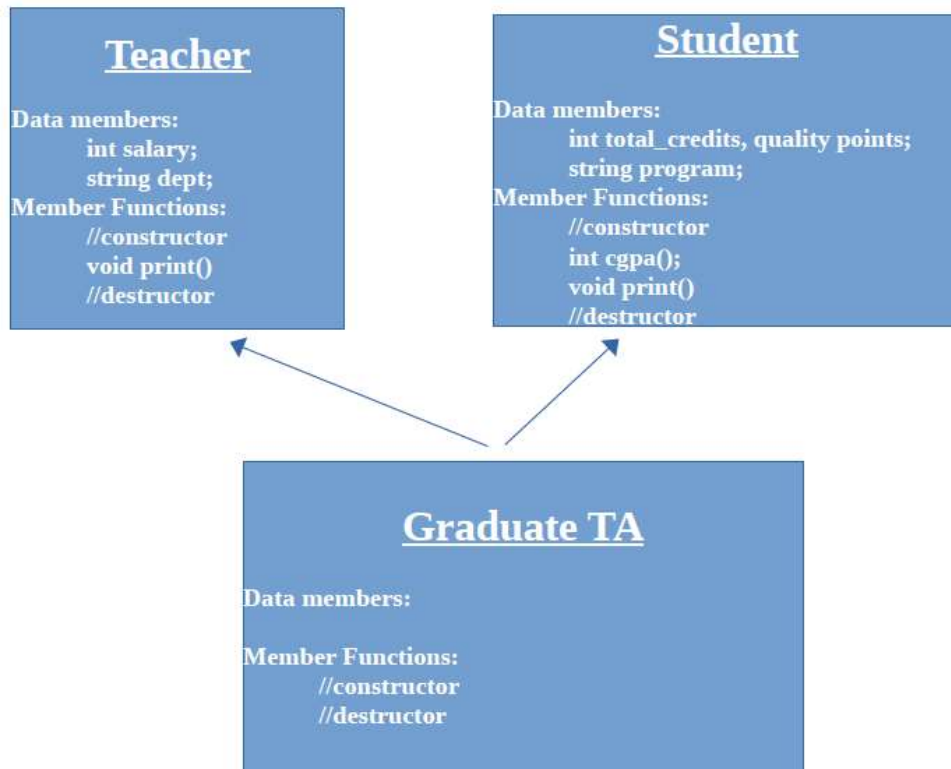
calling Default constructor of Shape class
calling constructor of Square class
Name is: Square Color is: Black No of Sides: 4
Area of Square: 9
Perimeter of Square: 12
Diagonal of Square: 4.24264

Square Destractor Callled
Destractor Called of Shape Class
Rectangle Destractor Callled
Destractor Called of Shape Class
Quadrilateral Destractor Callled
Destractor Called of Shape Class
Triangle Destractor Callled
Destractor Called of Shape Class
Destractor Called of Shape Class
```

Problem No :02

Problem Title:

2. Implement the following multilevel inheritance:



Solution:

```
#include<bits/stdc++.h>
using namespace std;

class Student
{
private:
    int totalCredit,earnTotalCreadit;
    float qualityPoint;
    string program;
public:
    Student()
    {
        cout<<"Default Constractor Called in Student Class\n";
    }
    Student(int sTotalCreadit,int sEarnTotalCreadit,string sProgram)
    {
        totalCredit=sTotalCreadit;
        earnTotalCreadit=sEarnTotalCreadit;
        program=sProgram;
        cout<<"Constractor Called in Student Class\n";
    }
}
```



```

float cgpa()
{
    return (earnTotalCreadit*1.0/totalCredit*1.0);
}

void display()
{
    cout<<"Program Name: "<<program<<endl;
    qualityPoint=cgpa()*100;
    if(qualityPoint==100 ||qualityPoint>=80)
    {
        cout<<"Cgpa is: "<<cgpa()*4<<" Grade is A+"<<endl;
    }
    else if(qualityPoint<80 && qualityPoint>=75)
    {
        cout<<"Cgpa is: "<<cgpa()*4<<" Grade is A"<<endl;
    }
    else if(qualityPoint<75 && qualityPoint>=70)
    {
        cout<<"Cgpa is: "<<cgpa()*4<<" Grade is A-"<<endl;
    }
    else if(qualityPoint<70 && qualityPoint>=65)
    {
        cout<<"Cgpa is: "<<cgpa()*4<<" Grade is B+"<<endl;
    }
    else if(qualityPoint<65 && qualityPoint>=60)
    {
        cout<<"Cgpa is: "<<cgpa()*4<<" Grade is B"<<endl;
    }
    else if(qualityPoint<60 && qualityPoint>=55)
    {
        cout<<"Cgpa is: "<<cgpa()*4<<" Grade is B-"<<endl;
    }
    else if(qualityPoint<55)
    {
        cout<<"Grade is F"<<endl;
    }
    else
    {
        cout<<"Result is Not Possible to calculate"<<endl;
    }
}

~Student()
{
    cout<<"Destructor Called in Student Class\n";
}

};

class Teacher:public Student
{
private:
    int salary;
    string department;

```

```

    string name;
public:
    Teacher()
    {
        cout<<"Default Constractor Called in Teacher Class\n";
    }
    Teacher(string tName,string tDepartment,int tSalary,int sTotalCredit,int sEarnCreadit,string
SPProgram):Student(sTotalCredit,sEarnCreadit,SPProgram)
    {
        salary=tSalary;
        department=tDepartment;
        name=tName;
        cout<<"Constractor Called in Teacher Class\n";
    }

    void display()
    {
        cout<<"Teacher Info:\n";
        cout<<"Name: "<<name<<endl;
        cout<<"Department: "<<department<<endl;
        cout<<"Salary: "<<salary<<endl;

        cout<<"\nStudent Information:"<<endl;
        Student::display();
    }

    void displayStudent()
    {
        cout<<"Student Information:"<<endl;
        Student::display();
    }
    void displayTeacher()
    {
        cout<<"Teacher Info:\n";
        cout<<"Name: "<<name<<endl;
        cout<<"Department: "<<department<<endl;
        cout<<"Salary: "<<salary<<endl;
    }

    ~Teacher()
    {
        cout<<"Destractor Called in Teacher Class"<<endl;
    }
};

// Use In Multilevel Inheritance:

class Graduate:public Teacher
{
public:

    Graduate()
    {
        cout<<"Default Constructor Called in Graduate CLass"<<endl;
    }

    Graduate(string tName,string tDepartment,int tSalary,int totalCredit,int earnCreadit,string program):

```

```

    Teacher(tName,tDepartment,tSalary,totalCredit,earnCreadit,program)
    {
        cout<<"Constructor Called in Graduate CClass"<<endl;
    }

void displayTeacherInfo()
{
    Teacher::displayTeacher();
}

void displayStudentInfo()
{
    Teacher::displayStudent();
}
void displayTeacherAndStudentInfo()
{
    Teacher::display();
}

~Graduate()
{
    cout<<"Destructor Called in Graduate Class"<<endl;
}

};

int main()
{
    Graduate g1=Graduate("Sabah Binte Nur","Cse",50000,20,19,"cse 1st year");
    cout<<"*****Display Student Information*****"<<endl<<endl;
    g1.displayStudentInfo();
    cout<<"\n\n*****Display Teacher Information*****"<<endl<<endl;
    g1.displayTeacherInfo();
    cout<<"\n\n*****Display Teacher & Student Information*****"<<endl<<endl;
    g1.displayTeacherAndStudentInfo();
    cout<<endl<<endl;

    return 0;
}

```

Output:

```
"C:\Users\shuvo\OneDrive - duet.ac.bd\DUET\CSE2-1\Lab\Programming\Lab 05\problem 02.exe"
Constructor Called in Student Class
Constructor Called in Teacher Class
Constructor Called in Graduate Class
*****Display Student Information*****

Student Information:
Program Name: cse 1st year
Cgpa is: 3.8 Grade is A+

*****Display Teacher Information*****

Teacher Info:
Name: Sabah Binte Nur
Department: Cse
Salary: 50000

*****Display Teacher & Student Information*****

Teacher Info:
Name: Sabah Binte Nur
Department: Cse
Salary: 50000

Student Information:
Program Name: cse 1st year
Cgpa is: 3.8 Grade is A+

Destructor Called in Graduate Class
Destructor Called in Teacher Class
Destructor Called in Student Class

Process returned 0 (0x0)   execution time : 0.100 s
Press any key to continue.
```