Assignment -33 **A Job Ready Bootcamp in C++, DSA and IOT** **MySirG**

Virtual Function, Abstract Class

1. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.

   Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

   Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of the triangles, and used as follows:

   Area of rectangle = x * y Area of triangle = 1/2 * x * y

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

class Shape
{

public:
    double x, y;
    void getData(double a, double b)
    {
        x = a;
        y = b;
    }
    virtual void displayArea() = 0;
};
class Triangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of triangle: " << ((x * y)/2) << " Unit" << endl;
    }
};
class Rectangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of rectangle: " << x * y << " Unit" << endl;
    }
};

int main()
{
    double b ,h;
    Shape *s;
    Triangle t;
    s = &t;
    cout << "Enter base and height of a triangle: ";
    cin >> b >> h;
    s->getData(b , h);
    s->displayArea();
    Rectangle r;
    s = &r;
    cout << "Enter length and breadth of a rectangle: ";
    cin >> b >> h;
    s->getData(b, h);
    s->displayArea();

    return 0;
}
=============================================================================
Output:
Enter base and height of a triangle: 4 6
Area of triangle: 12 Unit
Enter length and breadth of a rectangle: 8 9
Area of rectangle: 72 Unit
```

2. Extend the above program to display the area of circles. This requires the addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get_data() function as a default one with zero value.)

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

class Shape
{

public:
    double x, y;
    void getData(double a, double b = 0)
    {
        x = a;
        y = b;
    }
    virtual void displayArea() = 0;
};
class Triangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of triangle: " << ((x * y) / 2) << " Unit" << endl;
    }
};
class Rectangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of rectangle: " << x * y << " Unit" << endl;
    }
};
class Circle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of circle: " << (3.14159 * x * x) << " Unit" << endl;
    }
};
int main()
{
    double b, h;
    Shape *s;
    Triangle t;
    s = &t;
    cout << "Enter base and height of a triangle: ";
    cin >> b >> h;
    s->getData(b, h);
    s->displayArea();
    Rectangle r;
    s = &r;
    cout << "Enter length and breadth of a rectangle: ";
    cin >> b >> h;
    s->getData(b, h);
    s->displayArea();
    Circle c;
    s = &c;
    cout << "Enter radius of a circle: ";
    cin >> b;
    s->getData(b);
    s->displayArea();
    return 0;
}
=============================================================================
Output:
Enter base and height of a triangle: 5 8
Area of triangle: 20 Unit
Enter length and breadth of a rectangle: 12 16
Area of rectangle: 192 Unit
Enter radius of a circle: 2.5
Area of circle: 19.6349 Unit
```

3. Using the concept of pointers, write a function that swaps the private data values of two objects of the same class type.

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

class Swap
{
private:
    int x;

public:
    Swap(){};
    Swap(int a)
    {
        x = a;
    }
    friend void swap(Swap *X, Swap *Y);
    void display()
    {
        cout << x << endl;
    }
};
void swap(Swap *X, Swap *Y)
{
    int a;
    a = X->x;
    X->x = Y->x;
    Y->x = a;
}
int main()
{
    Swap s1(150), s2(29);
    cout << "Before Swapping" << endl;
    cout << "s1 = ";
    s1.display();
    cout << "s2 = ";
    s2.display();
    swap(&s1, &s2);
    cout << "After Swapping" << endl;
    cout << "s1 = ";
    s1.display();
    cout << "s2 = ";
    s2.display();
    return 0;
}
===============================================================================
Output:
Before Swapping
s1 = 150
s2 = 29
After Swapping
s1 = 29
s2 = 150
```

4. Create a base class called shape. Use this class to store 2 double type values that could be used to compute the area of figures. Derive 2 specific classes called triangle and rectangle from the base shape. Add to the base class a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in derived classes to suit their requirements. Using these 3 classes, design a program that will accept the dimensions of the shapes interactively and display area.

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

class Shape
{

public:
    double x, y;
    void getData(double a, double b)
    {
        x = a;
        y = b;
    }
    virtual void displayArea() = 0;
};
class Triangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of triangle: " << ((x * y)/2) << " Unit" << endl;
    }
};
class Rectangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of rectangle: " << x * y << " Unit" << endl;
    }
};

int main()
{
    double b ,h;
    Shape *s;
    Triangle t;
    s = &t;
    cout << "Enter base and height of a triangle: ";
    cin >> b >> h;
    s->getData(b , h);
    s->displayArea();
    Rectangle r;
    s = &r;
    cout << "Enter length and breadth of a rectangle: ";
    cin >> b >> h;
    s->getData(b, h);
    s->displayArea();

    return 0;
}
===============================================================================
Output:
Enter base and height of a triangle: 3 6
Area of triangle: 9 Unit
Enter length and breadth of a rectangle: 10 12
Area of rectangle: 120 Unit
```

5. Create a base class called Photon. Use this class to store a double type value of wavelength that could be used to calculate photon energy. Create class calculate_photonEnergy which will photon energy. Using these classes, calculate photon energy.

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

class Photon
{
protected:
    double wavelength;

public:
    void setData(double X)
    {
        wavelength = X;
    }
};

class calculate_photonEnergy : public Photon
{
    double E;

public:
    void calculateEnergy()
    {
        E = (6.626 * pow(10, -34) * 3 * pow(10, 8)) / wavelength;
    }
    void displayData()
    {
        cout << "Photon energy: " << E << " J"<<  endl;
    }
};

int main()
{
    calculate_photonEnergy p;
    p.setData(400);
    p.calculateEnergy();
    p.displayData();
    return 0;
}
===============================================================================
Output:
```

```
Photon energy: 4.9695e-28 J
```

6. Extend above to display the area of circles. For a circle,only one value is needed i.e. radius but in get_data() function 2 values are passed.

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

class Shape
{

public:
    double x, y;
    void getData(double a, double b = 0)
    {
        x = a;
        y = b;
    }
    virtual void displayArea() = 0;
};
class Triangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of triangle: " << ((x * y) / 2) << " Unit" << endl;
    }
};
class Rectangle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of rectangle: " << x * y << " Unit" << endl;
    }
};
class Circle : public Shape
{
public:
    void displayArea()
    {
        cout << "Area of circle: " << (3.14159 * x * x) << " Unit" << endl;
    }
};
int main()
{
    double b, h;
    Shape *s;
    Triangle t;
    s = &t;
    cout << "Enter base and height of a triangle: ";
    cin >> b >> h;
    s->getData(b, h);
    s->displayArea();
    Rectangle r;
    s = &r;
    cout << "Enter length and breadth of a rectangle: ";
    cin >> b >> h;
    s->getData(b, h);
    s->displayArea();
    Circle c;
    s = &c;
    cout << "Enter radius of a circle: ";
    cin >> b;
    s->getData(b);
    s->displayArea();
    return 0;
}
=============================================================================
Output:
Enter base and height of a triangle: 5 8
Area of triangle: 20 Unit
Enter length and breadth of a rectangle: 12 16
Area of rectangle: 192 Unit
Enter radius of a circle: 2.5
Area of circle: 19.6349 Unit
```

7. Create a base class called Matrix. Use this class to store 4 int type values that could be used to calculate determinants and create matrices. Create class calculate_determinant which will calculate the determinant of a matrix. Using these classes, calculate the determinant of the matrix.

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

class Matrix
{
    protected:
        int m[2][2];
};

class calculate_determinant : public Matrix
{
    public:
        void inputMatrix()
        {
            cout << "Enter elemnts of 2 x 2 Matrix" << endl;
            cin >> m[0][0] >> m[0][1] >> m [1][0] >> m[1][1];
        }
        void displayMatrix()
        {
            cout << "Matrix" << endl;
            cout << m[0][0] << "   " << m[0][1] << endl;
            cout << m[1][0] << "   " << m[1][1] << endl;
        }
        void determinant_calculate()
        {
            int d;
            d = (m[0][0] * m[1][1]) - (m[0][1] * m[1][0]);
            cout << "Determinant: " << d << endl;
        }
};

int main()
{
    calculate_determinant c;
    c.inputMatrix();
    c.displayMatrix();
    c.determinant_calculate();
    return 0;
}
=============================================================================
Output:
Enter elemnts of 2 x 2 Matrix
-3
0
2
25
Matrix
-3  0
2  25
Determinant: -75
```

8. Create a base class called proof. Use this class to store two int type values that could be used to prove that triangle is a right angled triangle. Create a class compute which will determine whether a triangle is a right angled triangle.

   Using these classes, design a program that will accept dimensions of a triangle, and display the result.

(Summary: Prove that triangle is a right angled triangle using pythagoras theorem).

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

class Proof
{
protected:
    int a, b;
};

class compute : public Proof
{
    int c;

public:
    void acceptDimensions()
    {
        cout << "Enter Hypotenuse, Perpendicular and Base respectively: ";
        cin >> a >> b >> c;
    }
    void display()
    {
        if (a * a == (b * b + c * c))
            cout << "Right angle triangle" << endl;
        else
            cout << "Not right angle triangle" << endl;
    }
};

int main()
{
    compute c;
    c.acceptDimensions();
    c.display();
    return 0;
}
```

```
==============================================================================
Output:
Enter Hypotenuse, Perpendicular and Base respectively: 13 5 12
Right angle triangle
```

9. Create a base class called volume. Use this class to store two double type values that could be used to compute the volume of figures. Derive two specific classes called cube and sphere from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_volume() to compute and display the volume of figures. Make display_volume() as a virtual function and redefine this function in the derived classes to suit their requirements.

   Using these three classes, design a program that will accept dimensions of a cube or a sphere interactively, and display the volume.

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

class volume
{
protected:
    double side;

public:
    void get_data(double x)
    {
        side = x;
    }
    virtual void display_volume() {}
};
class cube : public volume
{
public:
    void display_volume()
    {
        cout << "Side of cube is " << side << endl;
         cout << "Volume of a cube is " << side * side * side << " Unit" <<
endl;
    }
};
class sphere : public volume
{
public:
    void display_volume()
    {
        cout << "Radius of sphere is " << side << endl;
          cout << "Volume of a sphere is " << (4 * 3.14159 * side * side *
side) / 3 << " Unit" << endl;
    }
};

int main()
{
    volume *v;
    cube c;
    v = &c;
    v->get_data(5);
    v->display_volume();
    sphere s;
    v = &s;
    v->get_data(10);
    v->display_volume();
    return 0;
}
==========================================================================
Output:
Side of cube is 5
Volume of a cube is 125 Unit
Radius of sphere is 10
Volume of a sphere is 4188.79 Unit
```

10. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called square and parallelogram from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.

    Using these three classes, design a program that will accept dimensions of a square or a parallelogram interactively, and display the area.

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

class shape
{
protected:
    double base, height;

public:
    void get_data(double x, double y = 0)
    {
        base = x;
        height = y;
    }
    virtual void display_volume() {}
};
class square : public shape
{
public:
    void display_volume()
    {
        cout << "Side of square is " << base << endl;
        cout << "Area of square is " << base * base << " Unit" << endl;
    }
};
class parallelogram : public shape
{
public:
    void display_volume()
    {
          cout << "Base and height of parallelogram is " << base << " " <<
height << endl;
         cout << "Area of parallelogram is " << base * height << " Unit" <<
endl;
    }
};

int main()
{
    shape *v;
    square c;
    v = &c;
    v->get_data(5);
    v->display_volume();
    parallelogram s;
    v = &s;
    v->get_data(10, 15);
    v->display_volume();
    return 0;
}
=============================================================================
Output:
Side of square is 5
Area of square is 25 Unit
Base and height of parallelogram is 10 15
Area of parallelogram is 150 Unit
```