

# **Jaypee University of Engineering and Technology, Guna**

**Department of Computer Science and Engineering**

**Object Oriented Programming Lab  
(18B17CI271)**

## **Lab Exercise-8** **(Run time polymorphism)**

**Name : Palash Mishra**

**Enroll no. : 201B172**

## Question 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.

## A:

Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area using the concept of dynamic binding. 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 triangles and used as follows:

**Area of rectangle =  $x * y$**

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

## Solution:

```
#include <iostream>
using namespace std;
class Shape
{
protected:
    double x, y;
public:
    void get_data()
    {
        cout << "Enter the value for x and y" << endl;
        cin >> x >> y;
    }
    virtual void display_area()
    {
        double area;
        area = x * y;
        cout << "Area : = " << area << " sq.unit" << endl;
    }
};
class Triangle : public Shape
{

```

```

public:
    get_data();
    void display_area()
    {
        double area;
        area = 0.5 * x * y; //Area of Triangle
        cout << "Area of Triangle = " << area << " sq.unit" << endl;
    }
};

class Rectangle : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        area = x * y; //Area of Rectangle
        cout << "Area of Rectangle= " << area << " sq.unit" << endl;
    }
};

int main()
{
    Shape *B1 = new Triangle;
    B1->get_data();
    B1->display_area();
    Shape *B2 = new Rectangle;
    B2->get_data();
    B2->display_area();
    return 0;
}

```

## Output:

```

PS C:\Users\hp\Desktop\tut9> cd "c:\Users\hp\Desktop\tut9\" ; if ($?) { g++ lab8A.cpp -o lab8A } ; if ($?) { .\lab8A }
Enter the value for x and y
5
4
Area of Triangle = 10 sq.unit
Enter the value for x and y
5
4
Area of Rectangle= 20 sq.unit
PS C:\Users\hp\Desktop\tut9> █

```

## **B:**

Extend the Program-1 to display the area of circle. This requires 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.)

## **Solution:**

```
#include <iostream>
using namespace std;
class Shape
{
protected:
    double x, y;

public:
    void get_data()
    {
        cout << "Enter the value for x and y" << endl;
        cin >> x >> y;
    }
    virtual void display_area()
    {
        double area;
        area = x * y;
        cout << "Area : = " << area << " sq.unit" << endl;
    }
};

class Triangle : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        area = 0.5 * x * y; //Area of Triangle
        cout << "Area of Triangle = " << area << " sq.unit" << endl;
    }
};

class Rectangle : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        area = x * y; //Area of Rectangle
    }
};
```

```

        cout << "Area of Rectangle =" << area << " sq.unit" << endl;
    }
};
class Cirlce : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        cout<<"You can put any value of y because Area is independent on y\n";
        y = x;
        area = 3.14 * x * y;//Area of Circle
        cout << "Area = " << area << " sq.unit" << endl;
    }
};
int main()
{
    Shape *B1 = new Triangle;
    B1->get_data();
    B1->display_area();
    Shape *B2 = new Rectangle;
    B2->get_data();
    B2->display_area();
    Shape *B3 = new Cirlce;
    B3->get_data();
    B3->display_area();
    return 0;
}

```

## Output:

```

PS C:\Users\hp\Desktop\tut9> cd "c:\Users\hp\Desktop\tut9\" ; if ($?) { g++ Lab8B.cpp -o Lab8B } ; if ($?) { .\Lab8B }
Enter the value for x and y
5
6
Area of Triangle = 15 sq.unit
Enter the value for x and y
5
6
Area of Rectangle =30 sq.unit
Enter the value for x and y
5
6
You can put any value of y because Area is independent on y
Area = 78.5 sq.unit
PS C:\Users\hp\Desktop\tut9> █

```

## C:

Run the above program with the following modifications:

- Remove the definition of `display_area()` from one of the derived classes.
- In addition to the above change, declare the `display_area()` as pure virtual in the base class `Shape`.

Comment on the output in each case.

## Solution :

```
#include <iostream>
using namespace std;
class Shape
{
protected:
    double x, y;
public:
    void get_data()
    {
        cout << "Enter the value for x and y" << endl;
        cin >> x >> y;
    }
    virtual void display_area() = 0;
};
class Triangle : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        area = 0.5 * x * y; //Area of Triangle
        cout << "Area of Triangle = " << area << " sq.unit" << endl;
    }
};
class Rectangle : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        area = x * y; //Area of Rectangle
        cout << "Area of Rectangle = " << area << " sq.unit" << endl;
    }
}
```

```

};
class Cirlce : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        cout<<"You can put any value of y because Area is independent on y\n";
        y = x;
        area = 3.14 * x * y; //Area of Circle
        cout << "Area = " << area << " sq.unit" << endl;
    }
};
int main()
{
    Shape *B1 = new Triangle;
    B1->get_data();
    B1->display_area();
    Shape *B2 = new Rectangle;
    B2->get_data();
    B2->display_area();
    Shape *B3 = new Cirlce;
    B3->get_data();
    B3->display_area();
    return 0;
}

```

## Output :

```

PS C:\Users\hp\Desktop\tut9> cd "c:\Users\hp\Desktop\tut9\" ; if ($?) { g++ lab8C.cpp -o lab8C } ; if ($?) { .\lab8C }
Enter the value for x and y
6
7
Area of Triangle = 21 sq.unit
Enter the value for x and y
6
7
Area of Rectangle = 42 sq.unit
Enter the value for x and y
6
78
You can put any value of y because Area is independent on y
Area = 113.04 sq.unit
PS C:\Users\hp\Desktop\tut9> █

```