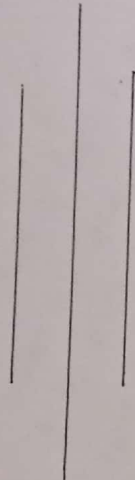


KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE



A

Lab Report On
Object Oriented Programming {C++}

Lab Report No: 5

Submitted by:
Ashraya Kadel
CE 2022
Rollno: 25

Submitted to:
Rajan Chuladyo
Department of Computer
Science and Engineering

SUBMISSION DATE: 13/08/2023

Q7: Create a class called Polygon with two data members: number of sides and centroid and two member functions: display() that displays the values of member variables and move() that translates polygon object to a new location.

Create two other classes Triangle and Rectangle inheriting from Polygon class. Add relevant data members and member functions in these classes.

Ans:

* Source Codes

```
#include <iostream>
using std::cin;
using std::cout;
```

```
class Point {
private:
```

```
    int x, y;
```

```
public:
```

```
    Point() {}
```

```
    Point(int a, int b) : x(a), y(b) {}
```

```
    void display()
```

```
{ cout << "X: " << x << "Y: " << y << std::endl; }
```

```
    void move(int a, int b)
```

```
{ x = x + a; y = y + b; }
```

```
    Point operator+ (const Point &p)
```

```
{ int a, b;
```

```
    a = x + p.x; b = y + p.y; return Point(a, b); }
```

```

Point operator = (const Point &c)
{
    x = c.x; y = c.y;
}
};

```

```

class Polygon
{
protected:
    int nsides; float centroid;
public:
    Polygon() {}
    void display()
    { cout << "P: " << centroid; }
};

```

```

class Triangle : private Polygon
{
private:
    Point x, y, z;
public:
    Triangle() {}
    Triangle (Point a, Point b, Point c)
    {
        nsides = 3; x = a;
        y = b; z = c;
    }
    void move (Point a, Point b, Point c)
    {
        x = x + a;
        y = y + b;
        z = z + c;
    }
};

```

(2)

```

class Rectangle : private Polygon
{
private:
    Point x, y, z, l;

```

```

public:
    Rectangle() {}
    Rectangle (Point a, Point b, Point c, Point d)
    {
        nsides = 4; x = a;
        y = b; z = c; l = d;
    }
    void move (Point a, Point b, Point c, Point d)
    {
        x = x + a; y = y + b;
        z = z + c; l = l + d;
    }

```

```

    void Rdisplay()
    {
        cout << "Rectangle = " << endl;
        cout << "No of sides: \n" << nsides;
        cout << "Point 1\t"; (Rectangle::x.display());
        cout << "Point 2\t"; (Rectangle::y.display());
        cout << "Point 3\t"; (Rectangle::z.display());
        cout << "Point 4\t"; (Rectangle::l.display());
    }
};

```

```

int main()
{
    Rectangle A (Point (0,0), Point (4,0), Point (4,4), Point (0,4));
    A.Rdisplay();
    A.move (Point (1,1), Point (1,1), Point (1,1), Point (1,1));
    A.Rdisplay();
    return 0;
}

```

(3)