**LAB # 10**

**OBJECT:**

To understand the concept of an abstract base class.

**SOURCE CODE:**

#include <iostream>

using namespace std;

class CPolygon{

protected:

int width, height;

public:

void set\_values (int a, int b)

{ width=a; height=b; }

virtual int area (void) =0;

};

class CRectangle: public CPolygon

{

public:

int area (void)

{ return (width \* height); }

};

class CTriangle: public CPolygon{

public:

int area (void)

{ return (width \* height / 2); }

};

int main (){

CRectangle rect;

CTriangle trgl;

CPolygon \* ppoly1 = &rect;

CPolygon \* ppoly2 = &trgl;

ppoly1->set\_values (4,5);

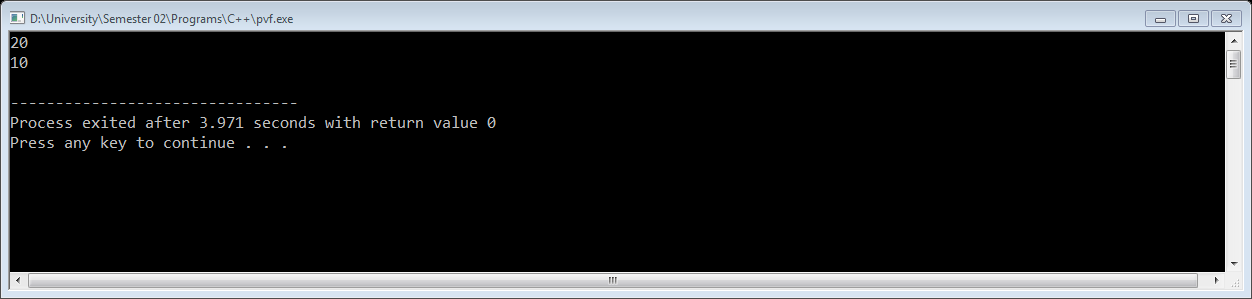
ppoly2->set\_values (4,5);

cout << ppoly1->area() << endl;

cout << ppoly2->area() << endl;

return 0;

}



**EXERCISE:** Create a base class called shape. Use this class to store two double type values that could be used to computer the area of the 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 function

display\_area () to computer and display the area of figures. Make display\_area ()

as virtual and redefine this function in derived class to suit their requirements.

Using these three classes, design a program that will accept the dimensions of a

triangle or a rectangle interactively and displays the area.

**SOURCE CODE:**

#include<iostream>

#include<conio.h>

using namespace std;

class Shape{

protected:

double l,b;

public:

void getdata(double l1,double b1){

l=l1;

b=b1;

}

void virtual display\_area()=0;

};

class triangle:public Shape{

public:

void display\_area(){

cout<<"\nArea of Triangle:"<<(0.5\*l\*b);

}

};

class Rectangle:public Shape{

public:

void display\_area(){

cout<<"\nArea of Rectangle:"<<l\*b;

}

};

int main(){

Shape \* s1;

triangle t1;

s1=&t1;

s1->getdata(20,10);

s1->display\_area();

Rectangle r1;

s1=&r1;

s1->getdata(5,4);

s1->display\_area();

return 0;

}