**OOP Lab 8 Polymorphism**

**Q1)**

//This program is made by Shuchita Gupta 211B311

#include<iostream>

using namespace std;

class shape

{public:

double x,y;

void getdata(double x1,double y1=0){ x=x1;y=y1;}

virtual void display\_area()=0; //Pure virtual function

//virtual void display\_area()

{cout<<"Virtual function"<<endl;};

};

class triangle:public shape

{public:

virtual void display\_area(){

cout<<"area of triangle is "<<0.5\*x\*y<<endl;}

};

class rectangle:public shape

{public:

virtual void display\_area()

{

cout<<"area of rectangle is "<<x\*y<<endl;

}

};

class circle:public shape

{public:

virtual void display\_area(){

cout<<"area of circle is "<<3.14\*x\*x<<endl;}

};

int main()

{int chh;

double x,y;

cout<<"Enter 1 to get area of rectangle \n 2 to get area of triangle \n 3 to get area of circle ";

cin>>chh;

if(chh==1){cout<<"enter length and breadth of rectangle ";

cin>>x>>y;

rectangle r1;

r1.getdata(x,y);

r1.display\_area();

}

if( chh==3){cout<<"enter radius of circle";

cin>>x;

circle c1;

c1.getdata(x);

c1.display\_area();}

if(chh==2) {cout<<"enter base and height of triangle ";

cin>>x>>y;

triangle t1;

t1.getdata(x,y);

t1.display\_area();

}

return 0;

}

**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.**

1. When definition of display\_area() is removes from any of the derived class, then display\_area() of base class is called using object of the derived class.
2. When display\_area() is made pure virtual function, the base class becomes abstract class. Classes inheriting an abstract class must provide definition to the pure virtual functions otherwise thay also become abstract class.

When the definition is removed from any of the derived classes, it also becomes abstract and an error is raised because we cannot create an object of an abstract class.