**Polymorphism and Virtual Functions**

**LAB #** **08**

**Fall 2019**

**CSE208L Object Oriented Programming Lab**

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Sumayyea Salahuddin**

January 6, 2020

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Objectives of the Lab:**

Objectives of the lab are to:

# Understand the difference between static and dynamic binding.

# Demonstrate the polymorphism with the help of virtual function and base class pointer.

* Develop abstract class using pure virtual functions.

# Activity # 01

**Title:**

Define an Abstract base class Shape, inherit Point from shape, Circle from Point and Cylinder from Circle.

**In C++**

**Source code:**

#include <iostream>

using namespace std;

class Shape

{

protected:

int area, vol;

public:

virtual void Area(){area=0;}

virtual void Volume(){vol=0;}

virtual void Display()const=0;

};

class Point: public Shape

{

protected:

int x, y;

public:

Point():x(0),y(0){}

Point(int a, int b){x=a,y=b;}

void Area(){area=0;}

void Volume(){vol=0;}

virtual void Display()const{cout<<"Area of Point: "<<area<<endl<<"Volume of Point: "<<vol<<endl;}

};

class Circle: public Point

{

protected:

int rad;

public:

Circle():rad(0),Point(0,0){}

Circle(int a, int b, int r){x=a,y=b,rad=r;}

void Area(){area=3.14\*rad\*rad;}

void Volume(){vol=0;}

void Display()const{cout<<"Area of Circle: "<<area<<endl<<"Volume of Circle: "<<vol<<endl;}

};

class Cylinder:public Circle

{

protected:

int height;

public:

Cylinder():height(0),Circle(0,0,0){}

Cylinder(int a, int b, int r, int h){x=a,y=b,rad=r,height=h;}

void Area(){area=2\*3.14\*rad\*(rad+height);}

void Volume(){vol=3.14\*rad\*rad\*height;}

void Display()const{cout<<"Area of Cylinder: "<<area<<endl<<"Volume of Cylinder: "<<vol<<endl;}

};

int main()

{

Shape \*sh;

Point P(3,4);

Circle C(4,2,5);

Cylinder Cyl(3,7,4,8);

sh=&P;

sh->Area();

sh->Volume();

sh->Display();

sh=&C;

sh->Area();

sh->Volume();

sh->Display();

sh=&Cyl;

sh->Area();

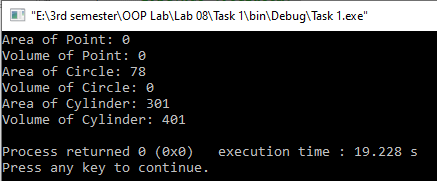
sh->Volume();

sh->Display();

return 0;

}

**Output:**



**Conclusion:**

This program helps us in understanding the basic concepts of Static and Dynamic Binding using Virtual Functions and Virtual Classes.