**CSE208L Object Oriented Programming Lab**

**LAB # 5**

****

**2020**

**Submitted to:**

**Engr. Sumayya Salahuddin**

**Submitted by:**

**TAYYABA**

**Registration No :**

**19PWCSE1854**

**Semester: 3rd**

**Class Section:** **C**

“On my honor, as student of University of Engineering and Technology,

I have neither given nor received unauthorized assistance on this

academic work.”

December 31, 2020

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Objectives of the Lab:**

Objectives of the lab are to:

1. Understand the concept of inheritance and its different forms.
2. Write derived class from base class using inheritance.
3. Understand the constructor and destructor chaining in inheritance hierarchy.
4. Know how to use base class constructors within derived class.

**ACTIVITY # 01**

**Title:**

.Reuse Point class of Lab 1 (Activity 1.4.2) as base class. Make the member data protected. Write all class member functions outside Point class.

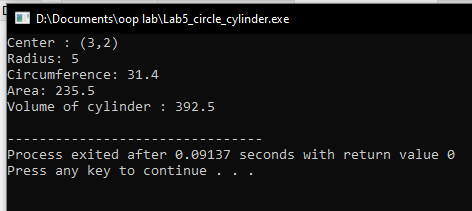
**Problem analysis:**

Create a class called point. This class maintains information about x and y coordinates. Make two sub classes circle and cylinder of point class and use the data member in sub classes as inheritance.

**In C++**

**Source code: Output:**

#include <iostream>

using namespace std;

class point{

protected:

double x, y;

public:

point();

point(double a, double b);

void setx(double a);

void sety(double b);

double getx();

double gety();

void show();

};

point::point(){

x=0; y=0;

}

point::point(double a, double b){

x=a; y=b;

}

void point::setx(double a){

x=a;

}

void point::sety(double b){

y=b;

}

double point::getx(){

return x;

}

double point::gety(){

return y;

}

void point::show(){

cout<<"("<<x<<","<<y<<")"<<endl;

}

class circle:public point{

private:

double radius;

public:

circle();

circle(double r, point p);

circle(double a, double b, double r);

void setter(double r);

double getter();

double display\_radius();

void calculate\_circum();

};

circle :: circle(){

x=0; y=0; radius=0;

}

circle :: circle(double r, point p){

x=p.getx();

y=p.gety();

radius=r;

}

circle :: circle(double a, double b, double r){

x=a; y=b; radius=r;

}

void circle :: setter(double r){

radius=r;

}

double circle :: getter(){

return radius;

}

double circle :: display\_radius(){

cout<<"Radius: "<<radius<<endl;

}

void circle :: calculate\_circum(){

cout<<"Circumference: "<<2\*3.14\*radius<<endl;

}

class cylinder : public point{

private:

double radius, height;

public:

cylinder();

cylinder(double h, circle c);

cylinder(double a, double b, double r, double h);

double area();

double volume();

};

cylinder::cylinder(){

height=0; radius=0; x=0; y=0;

}

cylinder :: cylinder(double h, circle c){

x=c.getx();

y=c.gety();

radius=c.getter();

height=h;

}

cylinder :: cylinder(double a, double b, double r, double h){

x=a;

y=b;

radius=r;

height=h;

}

double cylinder::area(){

cout<<"Area: "<<2\*3.14\*(radius + height)\*radius<<endl;

}

double cylinder::volume(){

cout<<"Volume of cylinder : "<<2\*3.14\*radius\*radius\*height<<endl;

}

int main(){

point p1(3,2);

circle c1(5,p1);

cout<<"Center : ";

c1.show();

c1.display\_radius();

c1.calculate\_circum();

cylinder cy1(2.5,c1);

cy1.area();

cy1.volume();

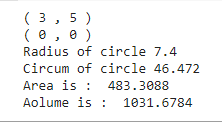
return 0;

}

**In Python**

**Source code: Output:**

class point:

    def \_\_init\_\_(self):

        self.x=0

        self.y=0

    def \_\_init\_\_(self,a,b):

        self.x=a

        self.y=b

    def setx(self,a):

        self.x=a

    def sety(self,b):

        self.y=b

    def getx(self):

        return self.x

    def gety(self):

        return self.y

    def display(self):

        print('(',self.x,',',self.y,')')

class circle(point):

    def \_\_init\_\_(self):

        self.radius=0

        self.x=0

        self.y=0

    def set\_values(self,r,p):

        self.radius=r

        self.x=p.getx()

        self.y=p.gety()

    def set\_values\_2(self,r,x,y):

        self.radius=r

        self.x=x

        self.y=y

    def setter(self,r):

        self.radius=r

    def getter(self):

        return self.radius

    def get\_radius(self):

        return self.radius

    def get\_circum(self):

        return (2\*3.14\*self.radius)

class cylinder(point):

    def \_\_init\_\_(self):

        self.radius=0

        self.height=0

        self.x=0

        self.y=0

    def set\_values(self,h,c):

        self.height=h

        self.radius=c.get\_radius()

        self.x=c1.getx()

        self.y=c1.gety()

    def setter(self,r):

        self.radius=r

    def getter(self):

        return self.radius

    def get\_area(self):

        return (2\*3.14\*self.radius\*(self.radius+self.height))

    def get\_volume(self):

        return (2\*3.14\*self.radius\*self.radius\*self.height)

p1=point(3,5)

p1.display()

c1=circle()

c1.display()

c1.set\_values(7.4,p1)

print('Radius of circle',c1.get\_radius())

print('Circum of circle',c1.get\_circum())

cy1=cylinder()

cy1.set\_values(3,c1)

print("Area is : ",cy1.get\_area())

print("Aolume is : ",cy1.get\_volume())

**Conclusion:**

This program helps us in understanding the basic concepts of classes and objects in different languages. It acts as a base for us and helps us in preparing ourselves for the higher level of programming. We get to know about inheritance in OOP with the help of this program.