/\*\*\*\*\*\*\*\*\*\* Online on Operator Overloading \*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Time: 30 minutes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Add the required functionalities so all the operations in main() works \*/

/\* You cannot add new member variables to any of the following classes. \*/

#include<iostream>

#include<cmath>

#include<string>

using namespace std;

const double pi = 3.1416;

void printVar(string s)

{

std::cout << endl << s << " = ";

}

class Point2D;

class Circle;

class RightCircularCone;

/\* The Point2D class \*/

class Point2D

{

double x,y;

public:

Point2D()

{

x = 0;

y = 0;

}

Point2D(double x, double y);

void setX(double x);

void setY(double y);

double getX();

double getY();

void print();

~Point2D();

};

Point2D::Point2D(double argx,double argy)

{

x = argx;

y = argy;

}

void Point2D::setX(double argx)

{

x=argx;

}

void Point2D::setY(double argy)

{

y = argy;

}

double Point2D::getX()

{

return x;

}

double Point2D::getY()

{

return y;

}

void Point2D::print()

{

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

}

Point2D::~Point2D()

{

x = 0;

y = 0;

}

/\* The Circle class \*/

class Circle

{

Point2D center;

double radius;

public:

Circle();

Circle(Point2D c, double r);

void setCenter(Point2D c);

void setRadius(double r);

Point2D getCenter();

double getRadius();

void print();

void reset() { center.setX(0); center.setY(0); radius=0; }

~Circle();

// your code

};

Circle:: Circle()

{

center.setX(0);

center.setY(0);

radius=0;

}

Circle::Circle(Point2D c, double r)

{

center.setX(c.getX());

center.setY(c.getY());

radius=r;

}

void Circle::setCenter(Point2D c)

{

center.setX(c.getX());

center.setY(c.getY());

}

void Circle::setRadius(double r)

{

radius=r;

}

Point2D Circle::getCenter()

{

return center;

}

double Circle::getRadius()

{

return radius;

}

void Circle::print()

{

cout << "[Center: ";

center.print();

cout << ", Radius: " << radius;

cout<<"]";

}

Circle::~Circle()

{

center.setX(0);

center.setY(0);

radius=0;

}

/\* The RightCircularCone class \*/

class RightCircularCone

{

Circle base;

double volume;

public:

RightCircularCone() {

base = Circle();

volume = -1;

}

~RightCircularCone() {

this->base.reset();

this->volume = -1;

}

void setBase(Circle c) { this->base = Circle(c.getCenter(), c.getRadius()); }

Circle getBase() { return this->base; }

void setVolume(double v) { this->volume = v; }

double getVolume() { return this->volume; }

void print() {

cout << "{Base = ";

this->base.print();

cout << ", Volume = " << this->volume << "}";

}

// your code

};

// your code

int main()

{

Point2D p1(5,10),p2(15,10);

printVar("Initial p1"); p1.print(); // Initial p1 = (5,10)

printVar("Initial p2"); p2.print(); // Initial p2 = (15,10)

Circle c1(p1,5),c2(p2,7);

printVar("Initial c1"); c1.print(); // Initial c1 = [Center: (5,10), Radius: 5]

printVar("Initial c2"); c2.print(); // Initial c2 = [Center: (15,10), Radius: 7]

RightCircularCone r1, r2;

printVar("Initial r1"); r1.print(); // Initial r1 = {Base = [Center: (0,0), Radius: 0], Volume = -1}

printVar("Initial r2"); r2.print(); // Initial r2 = {Base = [Center: (0,0), Radius: 0], Volume = -1}

// The following operation should set the base of r1 to be c1, and the implicit height of r1 to be 10.

// Note that there is no variable called height in the RightCircularCone class, and you cannot store such too.

// Volume of a right circular cone is defined as: (1/3) \* pi \* radius\_of\_base^2 \* height.

r1 = c1 \* 10;

printVar("Final r1"); r1.print(); // Final r1 = {Base = [Center: (5,10), Radius: 5], Volume = 261.8}

// The following operation should set the base of r2 to be c2, and the implicit height of r2 to be that of r1.

// e.g. here r2.volume = (1/3) \* pi \* r2.radius\_of\_base^2 \* r1.height.

r2 = c2 \* r1;

printVar("Final r2"); r2.print(); // Final r2 = {Base = [Center: (15,10), Radius: 7], Volume = 513.128}

cout << endl;

return 0;

}