#include<iostream>

using namespace std;

#ifndef POINT\_H

#define POINT\_H

class Point{

protected:

int x;

int y;

public:

Point();

Point(int valuex, int valuey);

void print();

void Setx(int setx);

void Sety(int sety);

int getx() const;

int getx() const;

};

#endif

#include"Point.h"

Point::Point()

{

x = 0;

y = 0;

}

Point::Point(int valuex, int valuey)

{

x=valuex;

y = valuey;

}

void Point::print()

{

cout << "X: " << x << " Y: " << y;

}

void Point::Setx(int setx)

{

x = setx;

}

void Point::Sety(int sety)

{

y = sety;

}

int Point::getx() const

{

return x;

}

int Point::getx() const

{

return y;

}

#include<iostream>

#include"Point.h"

using namespace std;

#ifndef CIRCLE\_H

#define CIRCLE\_H

class Circle: public Point

{

protected:

double radius;

public:

Circle();

Circle(int valuex, int valuey, int valuer);

void print();

void setRadius(double setr);

int getR() const;

double area()const;

};

#endif

#include"Circle.h"

Circle::Circle()

{

Point();

radius = 0;

}

Circle::Circle(int valuex, int valuey, int valuer)

{

Point(valuex, valuey);

radius = valuer;

}

void Circle::print()

{

cout << "X: " << x << " Y: " << y << " Radius: " << radius << endl;

}

void Circle::setRadius(double setr)

{

if (radius < 0)

{

radius = 0;

}

radius = setr;

}

int Circle::getR() const

{

return radius;

}

double Circle::area()const

{

return(3.14159\*radius\*radius);

}

#include<iostream>

#include "Circle.h"

using namespace std;

#ifndef CYLINDER\_H

#define CYLINDER\_H

class Cylinder:public Circle{

protected:

double height;

public:

Cylinder();

Cylinder(int valuex, int valuey, int valuer,int valuec);

void print();

void Setc(int setc);

int getc() const;

double volume(Circle r, Cylinder h)const;

};

#endif

#include "Cylinder.h"

Cylinder::Cylinder()

{

height = 0;

}

Cylinder::Cylinder(int valuex, int valuey, int valuer, int valuec)

{

Circle(valuex,valuey,valuer);

height = valuec;

}

void Cylinder::print()

{

cout << "X: " << x << " Y: " << y << " Radius: " << radius <<" Height: "<<height<< endl;

}

void Cylinder::Setc(int setc)

{

if (height < 0)

{

height = 0;

}

height = setc;

}

int Cylinder::getc() const

{

return height;

}

double Cylinder::volume(Circle r, Cylinder h)const

{

return(2 \* (3.14159\*radius\*radius) + 2 \* 3.14\*radius\*height);

}