First Part: (10 Marks)

Implement a Circle3D class whose surface is parallel to xy plane. They have a center and a radius.

Implement your class so that the main function given at the end works!

Your Circle3D class should provide following functionalities: area, volume, and contains(Point3D).

Contains(Point3d): determine if point is on the surface of the Circle.

Look at the following main function to implement your Circle3D class correctly.

Second part: (5 Marks)

Suppose, we want to manipulate square objects in our program. Our square objects wil be axis-parallel as the rectangle objects were in your offline.

Since a square is always a rectangle, we want to implement it by inheriting from the rectangle class.

Now, implement a Square class by inheriting from the Rectangle class.

Your square class should not have any member variables. It sould provide all functionalities of a square through rectangle's member variables and functions. However, it should have a constructor (see below) and other functions if necessary.

Your square class should allow users to create square objects as follows:

Square s(p, 50);

where p is the bottom-left point of the square and 50 is the length of the square's sides.

Look at the following main function to implement your square class correctly.

```
int main()

{

        Point3D p(5,5,5);

        point3D p3(7,7,7);

        Shape3D * s3d = new Circle3D(p, 10);

        cout << s3d->area();

        cout << s3d->volume(); //should print 0

        cout << s3d->contains(p3);


        Point2D p(10,10);

        Point2D p2(7,7);

        Shape * s = new Square(p, 20);

        cout << s->print(); //should print bottom-left point and side-length

        cout << s->area();

        cout << s->contains(p2);

}
```