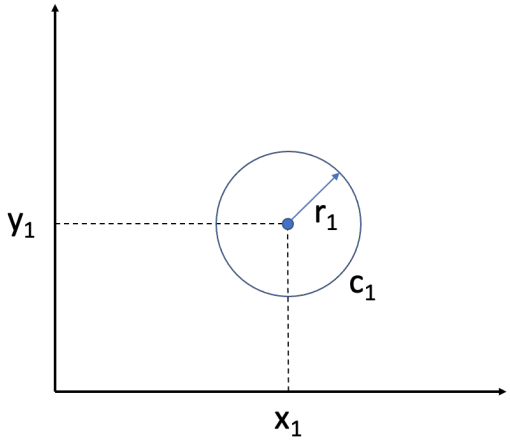
1: Point Class (with Tutor)

Create a new C# project for defining and testing the class Point described during the lecture. The project should include one file for the class Point and another file for a class called Program. Program should include the *Main* program entry-point. The *Main* will contain the instructions to create two points (p1 = (5, 1) and p2 = (7, 2)) and to print their coordinates on the screen.

2: Circle class (with Tutor)

Think about the possible design of a class to represent circles, as it was briefly discussed in the lecture.



For instance:

```
Point p1 = new Point(6, 4);
int r1 = 1;

Circle c1 = new Circle(p1, r1)
```

A Circle object should include a Display() method that prints the coordinates of its centre (hint: invoke the Display method of the Point object representing the centre) and the value of the radius. It should also include two additional methods for calculating and printing respectively the *circumference* and the *area*.
The Circle class should be added to the project of exercise 1 and a couple of Circle objects should be instantiated in the *Main* entry-point that was created (in Program). Read the coordinates of the centre and the radius of each circle as input from the keyboard. Show the properties of the circles, i.e., their circumference and area via invoking the above mentioned methods.

3: Distance between two Points (independent work)

Modify the definition of the class Point (of exercise 1) to include the following method:

```
public void DistanceFrom(Point p2)
```

The method should print on the screen the distance between the point on which it is invoked and the Point object passed as argument.
The formula to calculate the distance between two points (x_1, y_1) and (x_2, y_2) in a two-dimensional plane is:

$$\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

To implement the formula use the methods `Math.Sqrt` and `Math.Pow`. To access an attribute of a class, use the same dot notation we have been using for the methods (dot notation). Are you able to access the x and y coordinates of the p2 object using *p2.x* and *p2.y* from the body of the `DistanceFrom` method? Test the new `DistanceFrom (Point p2)` method from the Main entry-point of the ProgramTest class as in the previous exercises.

4: Segment Class (independent work)

Try to design a class that model Segment objects, e.g.:

```
Point p1 = new Point(2, 3);
Point p2 = new Point(3, 4);

Segment s1 = new Segment(p1, p2)
```

The class should include a method `Length()` that prints the length of the Segment. As in the previous exercise, use the same dot notation we have been using for the methods to access an attribute of the Point class. Are you still able to access the x and y coordinates of the p2 object using *p2.x* and *p2.y*?