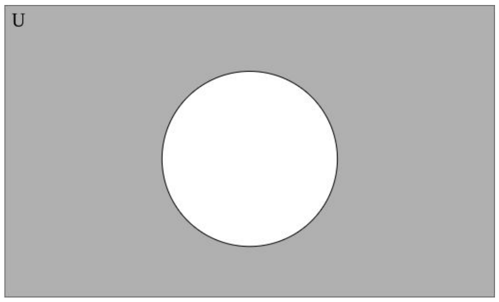
CS253 Lab 06

Mr Bean has always been a curious guy, fascinated by shapes. Sometime back, he thought of creating a software library of a few interesting regions. The main operations he wants to do on a region are:

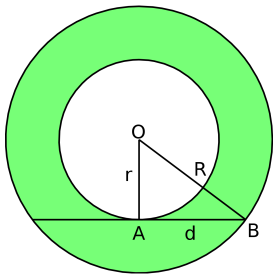
1. Check if a point belongs to the region (including its boundary/perimeter)
2. translate the region by (x,y) units: x unit on X-axis and y units on Y-axis
3. rotate the region by an angle θ (in radians).

He tried multiple times but couldn't do it alone. He recently got to know about you - a budding OOP and Design Patterns expert who has already implemented a Shapes package. He knows that you are a very helpful and intelligent, so he would like you to help him in creating the following region objects in an Object Oriented way, so as to minimise the effort and make code readable and maintainable:

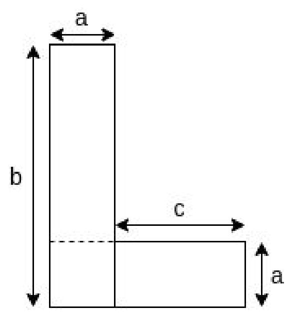
1. **Triangle**
2. **Rectangle**
3. **Square**
4. **Circle**
5. **Complement of a Circle:** Region that is ***not*** part of a circle (lies outside a given circle, Example, shaded region below)



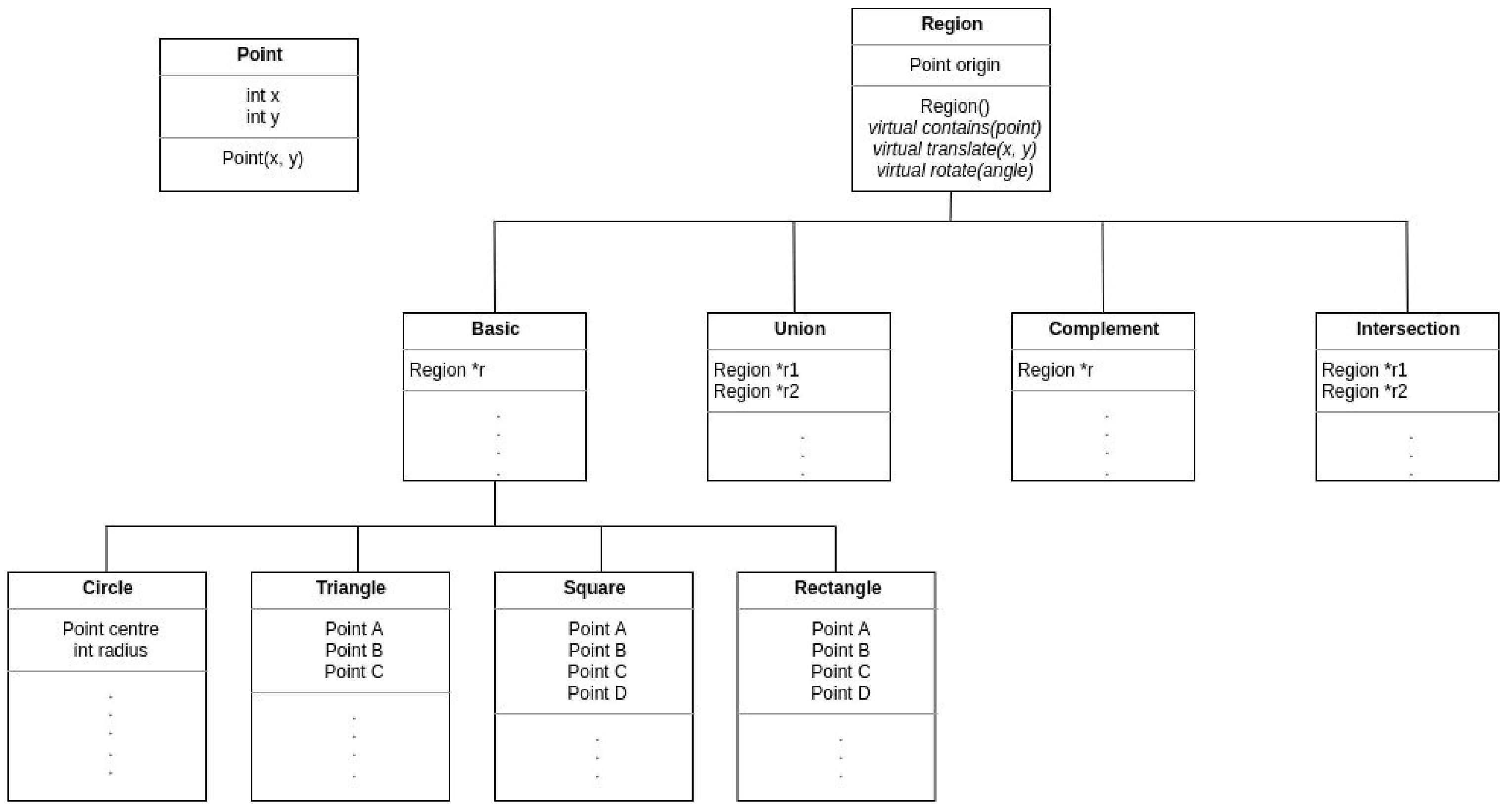
6. **Annulus:** A region bounded by two concentric circles.



7. **L-shaped region:** A region made using two rectangles.



Mr. Bean has tried his best to provide all the necessary details that might be helpful for you. He has scribbled everything on a paper. The picture is attached here:



Your task is to ***implement the regions package*** for Mr Bean.For every region, you may need an **'origin'** point which could be treated as the point of reference for the whole region while translating or rotating the region. You are free to choose any point as the point of origin. For example, for a circle, it could be the centre, for a rectangle it could be the left-bottom corner.

The diagram above just gives the rough idea, and is not complete. Mr Bean depends on your knowledge to come up with the best design. Thus, **apart from the given functions in the class diagram, you should add other necessary (but sufficient) functions for each class and also appropriate declarations for the given functions. [Don't rely too much on Mr Bean. :)]**

You will need to add a *Driver* routine to test your regions. There are 7 regions above. For each region R, you need to provide examples to test the following:

i. Point A inside R
ii. Point B outside R
iii. R1 = translate R to (x, y)
iv. Point A1 inside R but outside R1
v. Point B1 outside R but inside R1
vi. R2 = rotate R by angle θ
vii. Point A2 inside R but outside R2
viii. Point B2 outside R but inside R2

However, note that Scenario (vii) and (viii) may not be applicable for all regions.

Lab logistics:

Unlike earlier labs, this will be a *take home* lab.

b. On Thursday, 27th Feb 2020, we shall discuss the requirements during lab hours. Then you have to implement the functionality during the week 27th Feb - 5th March. The deadline for submission will be 5th March, end of the day.
c. You can use a language of your choice among Python, C++, Java. Please ask the instructor if you plan to use any other language.