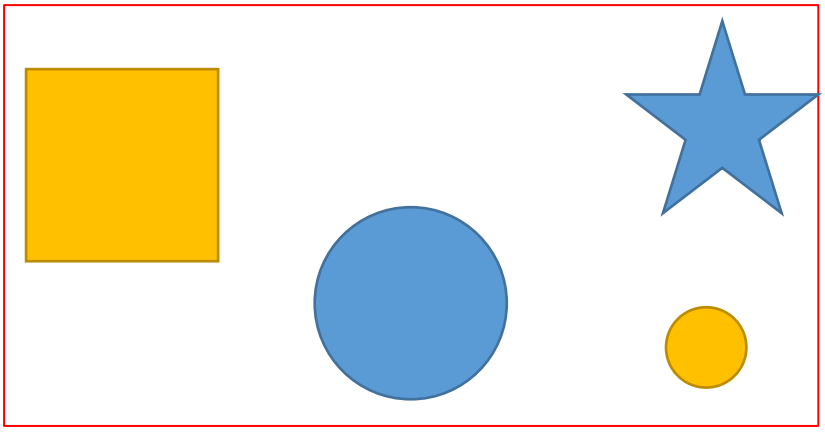
Several of the Microsoft Office products share the feature of a canvas. It allows the users to draw on it, like the following diagram (the border lines belong to the canvas):



Consider a simplified program, where we can **ONLY** draw circles and rectangles on the canvas (no need to consider other shapes). Choose one of these languages: C++/C#/Java, and write down some pseudo-codes (i.e. doesn't need to be grammatically accurate) for the following questions.

1. Define two classes for a **Circle** and a **Rectangle**. For both classes, we would also like to know where the center of the circle/rectangle is, and be able to get the area of the circle/rectangle. (10pts)

2. Obviously, these two classes share some common "characteristics". Define a **"Shape"** class that encapsulates these common characteristics. What is the name of the relationship between the **Rectangle**/**Circle** class and **Shape** class, and what do you need to change about your codes in Q1? Also, why do we want/need the **"Shape"** class at all? (10pts)

3. Should you be able to draw an "unspecified" Shape on the canvas? Or does it have to be a rectangular or circle? What does this mean about the Shape class? Hint: I'm looking for the word "abstract" in your answer. (5pts)

4. Also note that, the **Shapes** above all have colors, fills, shades, outlines, etc. To simplify it, let's combine all of these aesthetic elements of the **Shape** into one thing, and call it the "**Style**" of the **Shape**. If we want to include these information, what changes would you need to make to your previous codes? (10pts)

5. Should Style be its own class? (Hint: yes). What is the relationship between **Shape** and **Style** in your code/design? Hint: your answer should be one or more of the following: association, aggregation, composition, inheritance. (5pts)

6. Now, the user wants to draw something on the canvas. There are many ways to do this, but let's follow this method: there is a **Canvas** class, with a "draw" function. **The user decides upon the shape first**, then clicks on the canvas, and the "draw" function is called, and the shape shows up on the canvas. Write the sketch codes for the Canvas class, and the "draw" function in this scenario. (10pts)

7. What's the relationship between **Canvas** and **Shape**: association, aggregation, composition, inheritance? Compare this relationship to the relationship between **Shape** and **Style**, what are the differences, if there are any?

   Explain what would happen to the **Shape instances** on a **Canvas instance**, when you delete a **Canvas instance**?

   What would happen to the **"Style instances"** when you delete a **Shape instance**? (20pts)