Department of Computer Science
The City College of CUNY

CSc 22100 1XD: Software Design Laboratory [Summer 2018]
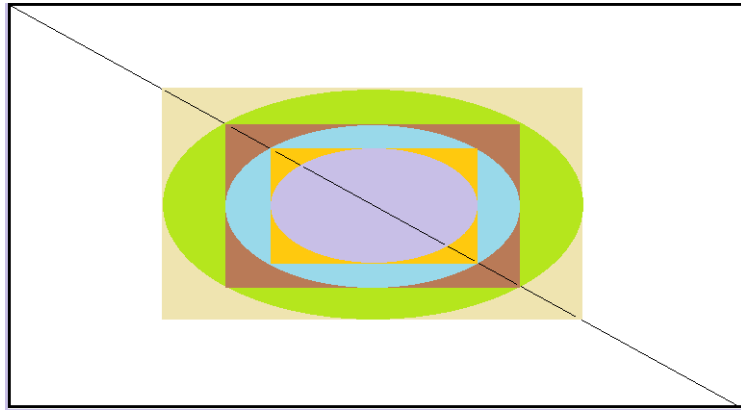
Exercise 2

A printout showing the problem, solution method, codes developed, and outputs produced for the tests indicated is due during and before the end of the class on Thursday, 12 July 2018. The deadline is strictly observed.

Consider the class hierarchy in Exercise 1.

1- Amend the hierarchy of Java classes in Exercise 1 as follows:

> Polygon *is_a* Shape;
> Rectangle *is_a* Shape
> Oval *is_a* Shape;
> Circle *is_a* Oval;

2- *Interface* ShapeInterface, *interface* PositionInterface, and *interface* ShapePositionInterface are specified in connection with the class hierarchy.

3- *Interface* ShapeInterface includes appropriate abstract, static, and/or default methods that describe the intrinsic functions and behaviors of the specific object types of the class hierarchy, including:
   a. *getArea* — describes the area of an object in the class hierarchy;
   b. *getPerimeter* — describes the perimeter of an object in the class hierarchy.

4- *Interface* PositionInterface includes appropriate abstract, static, and/or default methods that describe the positional functions and behaviors of the specific object types of the class hierarchy, including:
   a. *getPoint* – returns the point (x, y);
   b. *moveTo* – moves point (x, y) to point $(x + \Delta x, y + \Delta y)$;
   c. *distanceTo* – returns distance from point (x, y) to a point;

5- The *abstract* class Shape implements *interface* ShapePositionInterface which extends *interface* ShapeInterface and *interface* PositionInterface. *Interface* ShapePositionInterface includes appropriate abstract, static, and/or default methods that describe the functions and behaviors of the specific object types of the class hierarchy, including:
   a. *getBoundingBox* — returns the bounding rectangle of an object in the class hierarchy;
   b. *doOverlap* — returns true if two objects in the class hierarchy overlap.

6- Build a class Application that processes *polymorphically* the subclasses in the hierarchy to draw the geometric object shown:

Hesham A Auda
2 July 2018