CSc 115 Assignment 5 – Object-Orientation & Heaps

> Note: Rather than complicate this last assignment by creating a single application that incorporates both object-orientation and heap objectives, this assignment has two separate components: a) hardcopy hand-in questions that exercise heap objectives and b) an object-orientation programming exercise. LJ

Objectives: When you have completed this assignment you will be able to:

- Implement a class heirarchy
- Use and define abstract classes and methods
- Design polymorphic methods
- Define a Heap
- Insert nodes into a heap
- Remove nodes from a heap

Part 1 (Hardcopy Hand-in in Class): Due November 23, 2012

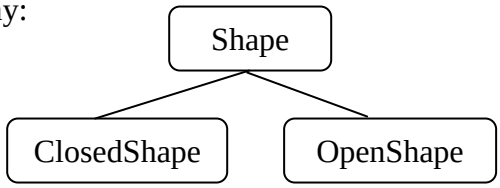
1. Draw a 'MaxHeap', called h, created by using the array: 12, 17, 3, 9, 2, 13, 7, 19.
2. Show what the heap would look like after each of the following pseudocode operations:
   a) `h.heapInsert(18)`
   b) `h.heapInsert(16)`
   c) `h.heapDelete()`
   d) Indicate and prove the Big-O running time of the heapInsert operation
   e) Indicate and prove the Big-O running time of the heapDelete operation

Hand in: Write your answers on paper and hand in at the be class on Friday November 23, 2012.

Part 2 (Programming Milestone): Due November 24, 2012 before 3:30 pm

Files to be downloaded: `Shape.java`, `OpenShape.java`, `ClosedShape.java`, `DrawingFrame.java`, `Main.java`.

1. Examine the Shape, SimpleShape and Compound classes provided. They implement the following class heirarchy:



   Consider issues like the following: In the constructor of the ClosedShape class, a complete copy of the x and y arrays are made. Why might the designer of this class have done that?
2. Complete the `setWidth()` and `setHeight()` methods in the ClosedShape class.

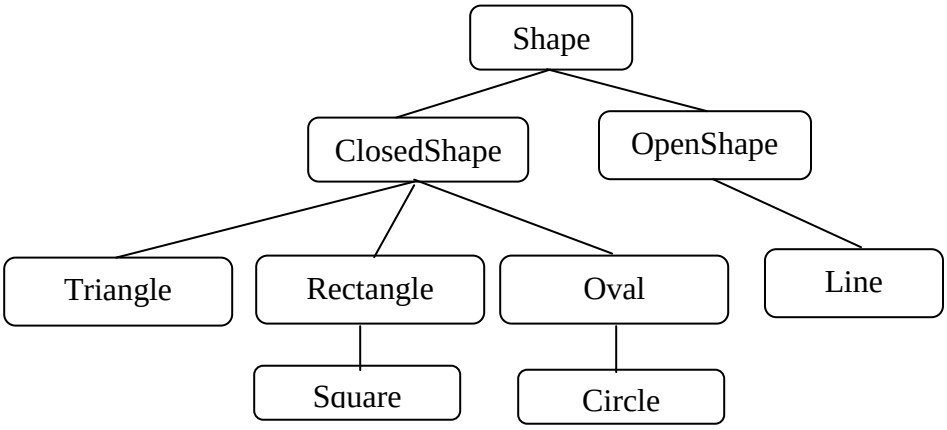
Observation: Since these are all *abstract classes*, no object of type Shape, ClosedShape nor OpenShape can be instantiated. (But feel free to try!)

This milestone is a formative exercise: It will not be graded but its completion will inform you and the instructor of your progress through this assignment.

Part 3 (Class Heirarchy): Due November 30, 2012 before 3:30 pm

(End of Term Option: Can be submitted late until December 3 at 3:30, will lose 10%)

Using the specifications below add Triangle, Rectangle, Square, Oval, and Circle classes below the ClosedShape class such that the heirarchy below is implemented. You will have to write methods for the three abstract methods: `Draw()`, `Area()` and `Perimeter()`.



Each of the non-abstract classes can now be instantiated. A very simple environmment to *draw* objects of these classes has been provided in the files `DrawingFrame.java` and `Main.java`. There is no need to alter the Main.java file (but you can if you want). The method paintComponent in `DrawingFrame.java` must be altered to paint your objects.

1. The constructor for the Triangle class must have the following signature:

   ```
   public Triangle(int x, int y, int base, int height)
   ```

   You can assume that this represents an isosolese triangle with the unique side oriented horizontally and the two equal sides up, where (x,y) is lower left point.
2. The constructor for the Rectangle class must have the following signature:

   ```
   public Rectangle(int x, int y, int width, int height)
   ```

   (x,y) is upper left corner

3. The constructor for the Square class must have the following signature:

   ```
   public Square(int x, int y, int length)
   ```

   (x,y) is upper left corner.
4. The constructor for the Oval class must have the following signature:

   ```
   public Oval(int x, int y, int width, int height)
   ```

   (x,y) indicate the point at the centre of the oval and width and height specify a bounding rectangle.
5. The constructor for the Circle class must have the following signature:

   ```
   public Circle(int x, int y, int radius)
   ```

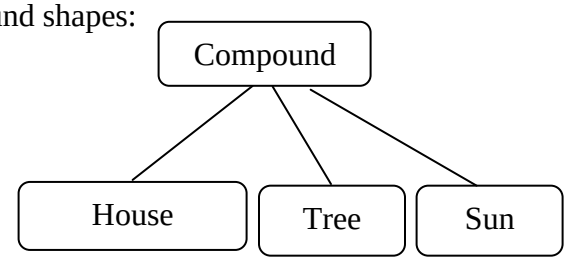
   (x,y) indicate the point at the centre of the oval and radius specifies the radius of the circle.
6. The constructor for the Line class must have the following method signature:

   ```
   public Line(int x, int y, double angle, double length)
   ```

   (x,y) indicate the leftmost point of the line; angle (in radians) indicates the angle the line makes with a right pointing horizontal line that starts at (x,y); and length indicates the length of the line.

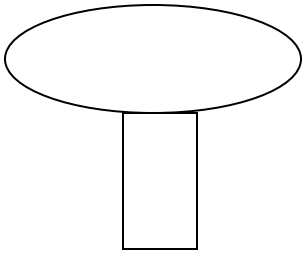
Part 4:

1. Now create the following class heirarch for Compound shapes:

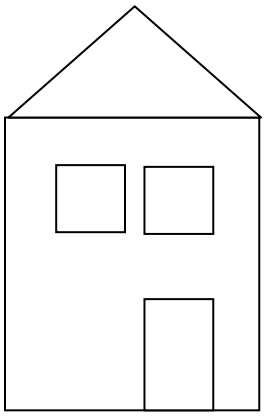


   Observe that there is a has-a relationship between components of the House, Sun, and Tree classes.

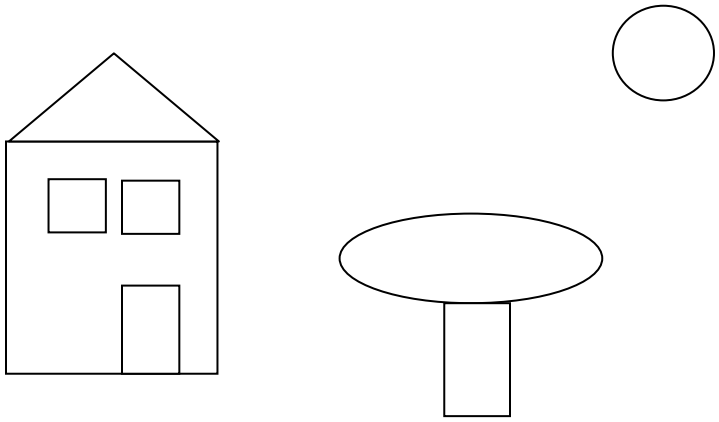
   A Tree should be composed of a rectangle and an oval, as follows:



A house is composed of a triangle, two rectangles and two squares, as follows:



2. Using objects from the Compound class, draw an outdoor scene, similar to the one below.
   (You can be creative here!)



Submitting your Solution:

- When complete submit All .java files that you used, modified, or created to complete this assignment to the CSc 115 Connex Site using the Assignments: Assignment 5 Submission link before 3:30 on Friday, November 30, 2012.
- Any file that you modified or created must contain a comment at the top that includes your name and student ID.
- If you adopted or adapted code from other sources, you must include an appropriate crediting reference to the original author or source.
- Be sure to include any testers that you wrote.