Lab 3 - Shapes Shapes Shapes

Due: end of class

TASKS

For this problem, we wish to draw a number of shapes, using "character graphics". For each part of the problem, your program should input all parameters from the keyboard (after prompting the user). Moreover, the program should ensure that inputs are legal for the shape in question. Do not worry about the case where the parameters are too big to be displayed properly in the xterm.

1. Draw a w-wide h-high rectangular frame, using asterisks. Ex (w=5, h=4):
   ```
   *****
   *   *
   *   *
   *****
   ```
2. Draw the lower triangular part of a square, given the side length. Ex (side=5):
   ```
   *
   **
   ***
   ****
   *****
   ```
3. Similar to 2, but draw the trapezoid with the bottom *height* elements of the triangle. Ex (side=5, height=3):
   ```
   ***
   ****
   *****
   ```
4. Similar to 2, but draw the upper triangular part of the square (i.e., the other half of the square).
5. Given a radius, draw a circle with that radius.
   Hint: Recall that a circle of radius r is defined as the set of points (x,y) where $x^2+y^2<r^2$.

Although we used character graphics in this lab, the same approaches can be used with individual pixels (or squinting hard enough at the asterisks).

HAND IN

Your 136 instructor will tell you what to hand in and how.

GENERAL COMMENTS FOR ALL PROGRAMS THIS SEMESTER

You should have the following header on all programs:

```
/*
  Author: <name>
  Course: {135,136}
  Instructor: <name>
  Assignment: <title, e.g., "Lab 1">

  This program does ...
*/
```

GRADING

All 135 and 136 programs this semester will be graded on:

- Correctness: Does your program work?
- Testing: Have you generated sufficient and good test data to give reasonable confidence that your program works?
- Structure: Have you structured your code to follow proper software engineering guidelines? This includes readability and maintainability.
- Documentation: How well documented is your code? Good documentation does not repeat the code in English, but explains the point of each code block, highlighting any design decisions and/or tricky implementation details.