CIS 330: Project #2H
Assigned: October 30th, 2018
Due Nov 5th, 2018
(which means submitted by 6am on Nov 6th, 2018)
Worth 4% of your grade

Assignment: You will implement 3 structs and 9 functions. All the files (starter code, correct output, grader program) needed for this project are available on the course website.

The three structs are Rectangle, Circle, and Triangle, and are described below.

The 3 structs refer to 3 different shapes: Triangle, Circle, and Rectangle.
For each shape, there are 3 functions: Initialize, GetArea, and GetBoundingBox.
You must implement 9 functions total (3*3).

The comments below clarify the format of the Rectangle, Circle, and Triangle, as well as the convention for GetBoundingBox, and an example of accessing data members for pointers to structs.

== Rectangle ==

The rectangle has corners (minX, minY), (maxX, minY), (minX, maxY), (maxX, maxY).
Its area is (maxX-minX)*(maxY-minY).
Its bounding box is from minX to maxX in X, and minY to maxY in Y.

== Circle ==

The circle has an origin (x and y) and a radius.

Its area is 3.14159*radius*radius.
Its bounding box is from (x-radius) to (x+radius) in X, and (y-radius) to (y+radius) in Y.

== Triangle ==

The triangle always has two points at the minimum Y-value. The third point's Y-value is at the maximum Y-value, and its X-value is at the average of the X's of the other two points. Saying it another way, the first two points form the "base", and the third point is "height" above it.

Thus, the area of the triangle is (pt2X-pt1X)*(maxY-minY)/2;
And the bounding box is from pt1X to pt2X in X, and from minY to maxY in Y.

== GetBoundingBox ==

The GetBoundingBox functions take a double * as an argument. If a shape has its minimum X at “a”, its maximum X at “b”, its minimum Y at “c”, and its maximum Y at “d”, then it should do something like:

```
void GetCircleBoundingBox(Circle *, double *bbox)
{
   bbox[0] = a;
   bbox[1] = b;
   bbox[2] = c;
   bbox[3] = d;
}
```

== What to modify ==

There is a line of code in proj2H.c that says “DO NOT MODIFY AFTER THIS POINT”. So, don’t.

Of course, if you want to temporarily modify main so you can add print statements, etc., then that is fine.

== Success ==

The grader program should walk you through what to do.

(It will compile, run it, and then compare the output to the correct output.)

== What to turn in ==

Only proj2H.c