CIS 212: Project #3A
Assigned: Thursday, November 1st, 2018
Due Sunday, November 11th, 2018
(which means submitted by 6am on November 2nd, 2018)
Worth 3% of your grade

Please read this entire prompt!

Assignment: You will build off your work from project 2H. In this project, you will build an abstract type (using only C) called Shape, complete with a dispatch table. You will also build 3 concrete types: Circle, Triangle, and Rectangle. The methods and structs for the concrete types will come from your project 2H.

1) Copy your 2H code: cp proj2H.c proj3A.c. All of the following instructions should be changes to the proj3A.c file you have just created.
2) Make an abstract type called Shape. It should have a data member called "self" of type "void *" and two data members that are pointers to functions – GetBoundingBox and GetArea. The signatures of these functions can be inferred from the starter code.
    a. Hint: define Shape immediately after the other three structs.
3) Modify the three "get area" functions and the three "get bounding box" functions to take void * arguments.
    a. Hint: You will need to do a cast inside the function. For example, if GetCircleArea has a "void *" function argument, then you will need to cast this void * to be a "Circle *" inside the function to be able to access its data members.
4) Rename the three Initialize functions to be Create functions. They should no longer take a circle/rectangle/triangle as an argument. Instead, they will create the correct shape in their function definition (with a malloc). They will also create a shape, and connect the shape's dispatch table to circle/rectangle/triangle. The Create function should return a shape.
    a. Hint: move the three initialize functions to go after the 6 get area / bounding box functions. You need to do this because the shape's dispatch table will want to point at functions like "GetCircleBoundingBox." If GetCircleBoundingBox appears later in the file, then the compiler won't have seen it yet, and it will complain.
5) Add functions called "GetBoundingBox" and "GetArea". The signatures for these functions are in the new starter code (proj3A_starter.c).
6) Replace the main function that was copied over from proj2H.c with the new main function in the new starter code (proj3A_starter.c).

== WARNING ==

Do not modify the main function that you get from proj3A_starter.c. Or, rather, if you modify it for testing purposes, make sure to switch it back. It is critical you call the functions the way we intend. If you do modify the way the functions are called from main, you should expect less than half credit.

== Success ==

Before you submit, make sure to test your code using the provided grader program script (grader.sh) on the Virtual Box.

Upload your new file, proj3A.c

== ADMIRE ==

You have implemented class inheritance in C. No small feat.