CS119 – Lab 6
Due Date: March 13

Purpose: We are not restricted to data types already defined for us in Haskell, but can also define our own types to suit are needs. In this lab we will define types that represent geometrical shapes and write functions that use these shapes.

Knowledge: This lab will help you become familiar with the following content knowledge:

- How to define your own data types
- How to write functions to manipulate your own defined data types

Task: Follow the steps in this lab carefully to complete the assignments. Copy the lab6 folder and write your functions in `Shape.hs`. You might want to start by creating a couple of shapes and computing their areas.

Assignment 1:
Add another case to the `Shape` data type which will create a Right Triangle by providing the two sides that meet at the right angle. Then modify the `area` function so that it computes the area of this new shape.

Criteria for Success: Calculate the area of a couple of right triangles and verify the results.

Assignment 2:
Write a function `scale :: Float -> Shape -> Shape`

The expression `scale factor s` will return a new shape that is scaled by the given factor.

Criteria for Success: Scale shapes of all three kinds and verify the results.

Assignment 3:
Define another data type `LocatableShape` that contains x and y coordinate values for the center for the center of the shape, as well as a given shape. You can do this by using the `type` keyword to define this type.

Then write a function:
`translate :: (Int,Int) -> LocatableShape- > LocatableShape`
which will move the shape. For example, if the first parameter is (10,10), this would increase both the x and y coordinates of the center of the shape by 10.

Criteria for Success: Translate a LocatableShape value and verify the the location has changed.

Submit your `Shape.hs` file in Canvas for grading.