CCPS506 Lab 6 – Custom Types in Haskell

Preamble

In this lab you'll create a custom type in Haskell and make it an instance of several built-in Haskell type-classes.

Lab Description

Create a custom Haskell type for representing regular polygons. Regular polygons are equilateral and equiangular (all sides have the same length, all angles between sides are the same). Common examples are square, pentagon, hexagon, octagon, etc.

This custom type must be called **`RegularPoly`** and will have two fields – one field for the number of sides (`Int`), and another for the length of each side (`Float`). Your **`RegularPoly`** type must support the following functionality:

i) Implement a function called **`polyArea`** that accepts a **`RegularPoly`** as input and returns the area of that polygon. Use Google to find a general formula for computing the area for any regular polygon.

ii) Your **`RegularPoly`** should be an instance of the **`Eq`** type class. Equality between two regular polygons is defined as having the same area. Note that two regular polygons need not have the same number of sides or side length to have the same area!

   As we know at this point in our computer science careers, comparing equality between floating point values is fraught. Thus, your (==) function should check equality to within some small value (i.e. 1e-15). If the areas are within this tiny value of each other, we call them equal ("close enough")

iii) Your **`RegularPoly`** should also be an instance of the **`Show`** type class. Come up with a visually pleasing way of displaying instances of your regular polygon type. How exactly you display this is up to you, but the number of sides and side length should be included somehow.

Submission

Everything related to your custom type should be contained in a single Haskell module called Lab6. This module should be stored in a single Haskell file called **lab6.hs**.

Labs are to be submitted ***individually***! Submit lab6.hs on D2L.