Devere A. Weaver
COP3014 Project 2

1. List of classes with members and functions

Class 1: Circle
Functions:
- area() - calculate area based on radius
- diameter() - calculate diameter based on radius
- circumference() - calculate circumference based on radius
- getRadius() - get the radius value
- setRadius() - set the radius

Data members:
- radius – double, radius of the Circle

Class 2: Triangle
Functions:
- getBase() - return the value of triangle base
- getHeight() - return the height of triangle
- setBase() - set the base of the triangle
- setHeight)() - set height of triangle
- area() - calculate area of triangle
- circle() - compare are of a Circle with the Triangle

Data Members:
- base – double , base of triangle
- height – double, height of triangle

Class 3: Rectangle
Functions:
- getWidth/getHeight() - return value of width and height respectively
- setWidth/setHeight() - set value of width and height respectively
- area() - calculate area of rectangle
- diagonal() - calculate the diagonal length of the rectangle
- perimeter() - calculate the perimeter of the rectangle
- isSquare() - compare length and width to see if the rectangle is a square

2. Steps to implement solution

1) Using pen and paper, think about the data for each class and how I want to manipulate it to get the desired state.
2) Using pen and paper, draw the desired flow of the program and any jumps in logic
3) Write out a more detailed description of each class member before implementing
4) Write out the control flow for main
5) Begin by implementing Circle class, test after each major change
6) Implement Triangle class, test with each major change
7) Implement Rectangle class, test with each major change
8) Implement main function to tie the classes together
9) Test each and every logical section of main, starting with Circle class, then Triangle, then Rectangle
10) After complete, continue to test each input/output to be sure that it is correct before turning in

3. Reflect

In the end I did indeed end up with exactly the same classes that I wanted to implement. I believe this was due to the fact that I did take so much time in thinking them over and exactly how I wanted them to fit into the core of the program. The hardest part of the project that required the most attention was actually the main function. I found it difficult to translate all the program requirements precisely how they were instructed, as a result, I had to interpret them whether the were correct or incorrect assumptions. The program in the end worked with I believe the majority of the functionality, but for sure leave a lot of room for improvement and refactoring later.