Class Diagrams and Pseudocode

Name of team members who collaborated on the design and test plan:

1. Name (*first last*): Sam Ogle
2. Name (*first last*): Grayson Whitaker
3. Name (*first last*): Adam Haaf
4. Name : Armando Sallis

**Naming Convention for Member Variables and Member Functions**

* Use the variable names included below to name your member variables
* Use ‘set*Variable*’ and ‘get*Variable*’ to name setter-functions and getter-functions (e.g., setSide in the Square class)
* Name the member function that performs the area calculation ‘calcArea’

# Circle Class

|  |
| --- |
| Circle |
| - radius: double |
| + setRadius(): bool  + getRadius(): double  + areaCalc(): double |

## Square Class

|  |
| --- |
| Square |
| - side: double |
| + setSide(): bool  + getSide(): double  + areaCalc(): double |

## Rectangle Class

|  |
| --- |
| Rectangle |
| - length: double  - width: double |
| + setLength(): bool  + getLength(): double  + setWidth(): bool  + setWidth(): double  + areaCalc(): double |

## Trapezoid Class

|  |
| --- |
| Trapezoid |
| - base1: double  - base2: double  - height: double |
| + setBase1(): bool  + getBase1(): double  + setBase2(): bool  + getBase2(): double  + setHeight(): bool  + getHeight(): double  + areaCalc(): double |

# Pseudocode for main program

*(See Ch. 1.6 in our textbook for an example of how to write detailed pseudocode)*

1. Create areaCalc.ccp file (functions as the main file)

2. Use a do-while loop to prompt the user with a menu(1-5)

-quit if user chooses 5

-validate user input and prompt for valid input if false

3. Instantiate an object for the correctly chosen class, and declare the variables needed to take in the user input (use cin for input)

4. Next use the set functions to validate the user input and return a bool to indicate if the program should re-prompt the user for a correct input.

6. next use the areaCalc functions within the classes to properly calculate and display the area