Class Diagrams and Pseudocode

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

1. Name (*first last*): Angie Diaz
2. Name (*first last*): Dylan Harvey
3. Name (*first last*): Janaki Bhosale

**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(r: double): void  +getRadius(): double  +calcArea(): double |

## Square Class

| Square |
| --- |
| -side: double |
| +setSide(s: double): void  +getSide(): double  +calcArea(): double |

## Rectangle Class

| Rectangle |
| --- |
| -length: double  -width: double |
| +setLength(l: double): void  +getLength(): double  +setWidth(w: double): void  +getWidth(): double  +calcArea(): double |

## Trapezoid Class

| Trapezoid |
| --- |
| - base1: double  - base2: double  - height: double |
| + setBase1(b1: double): void  + getBase1(): double  + setBase2(b2: int): void  + getBase2(): double  + setHeight(h: double): void  + getHeight(): double  + calcArea(): double |

# Pseudocode for main program

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

**In areaCalc.cpp (main())**

* Include header files
* Declare integer variable for menu choice, double variables for l, w, b1, b2, h, r, s
* In cout statements, print out menu options, 1 for circle, 2, for square, 3 for rectangle, 4 for trapezoid and 5 to quit
* Use cin statement to store user input for menu choice
* while menu choice variable is less than 1 or greater than 5:
  + Ask user to re-enter a valid menu choice
* if menu choice equals 5
  + quit program
* else if menu choice equals 1
  + create Circle object named circle1
  + Ask user for radius and store in r variable
  + Send r as argument in circle1.setRadius()
  + In a cout statement display area by calling circle1.calcArea()
* else if menu choice equals 2
  + create Square object named square1
  + Ask user for side and store in s variable
  + Send s as an argument in square1.setSide()
  + In a cout statement display area by calling square1.calcArea()
* else if menu choice equals 3
  + create Rectangle object named rectangle1
  + Ask user for length and width, store in l and w variables
  + Send l as argument in rectangle1.setLength()
  + Send w as an argument in rectangle1.setWidth()
  + In a cout statement, display area by calling rectangle1.calcArea()
* Else if menu choice equals 4
  + create Trapezoid object named trapezoid1
  + Ask user for base 1, base 2, and height; store in b1, b2, h variables
  + Send b1 as argument in trapezoid1.setBase1()
  + Send b2 as argument in trapezoid1.setBase2()
  + Send h as argument in trapezoid1.setHeight()
  + In a cout statement, display area by calling trapezoid1.calcArea()
* Exit program

**In Circle.h**

* Header Guards
* Include cmath
* Declare const double for PI = 3.14159
* Define class Circle (from UML diagram)
  + -radius: double
  + +setRadius(r: double): void
  + +getRadius(): double
  + +calcArea(): double
* End Header Guards

**In Circle.cpp**

* Include Circle.h
* Define void Circle::setRadius(double r)
  + radius = r
* Define double Circle::getRadius()
  + return radius
* Define double Circle::calcArea()
  + return (PI \* pow(radius, 2))

**In Rectangle.h**

* Header Guards
* Define class Rectangle (from UML diagram)
  + -length: double
  + -width: double
  + +setLength(l: double): void
  + +getLength(): double
  + +setWidth(w: double): void
  + +getWidth(): double
  + +calcArea(): double
* End Header Guards

**In Rectangle.cpp**

* Include Rectangle.h
* Define void Rectangle::setLength(double l)
  + length = l
* Define double Rectangle::getLength()
  + return length
* Define void Rectangle::setWidth(double w)
  + width = w
* Define double Rectangle::getWidth()
  + return width
* Define double Rectangle::calcArea()
  + return (length \* width)

**In Square.h**

* Header Guards
* Define class Square (from UML diagram)
  + -side: double
  + +setSide(s:double): void
  + +getSide(): double
  + +calcArea(): double
* End Header Guards

**In Square.cpp**

* Include Square.h
* Define void Square::setSide(double s)
  + side = s
* Define double Square::getSide()
  + return side
* Define double Square::calcArea()
  + return (side \* side)

**In Trapezoid.h**

* Header Guards
* Define class Trapezoid (from UML diagram)
  + -base1: double
  + -base2: double
  + -height: double
  + + setBase1(b1: double): void
  + + getBase1(): double
  + + setBase2(b2: int): void
  + + getBase2(): double
  + + setHeight(h: double): void
  + + getHeight(): double
  + + calcArea(): double
* End Header Guards

**In Trapezoid.cpp**

* Include Trapezoid.h
* Define void Trapezoid::setBase1(double b1)
  + base1 = b1
* Define double Trapezoid::getBase1()
  + return base1
* Define void Trapezoid::setBase2()
  + base2 = b2
* Define double Trapezoid::getBase2()
  + return base2
* Define void Trapezoid::setheight(double h)
  + height = h
* Define double Trapezoid::getheight()
  + return h
* Define double Trapezoid::calcArea()
  + return ((base1 + base2)\*height/2)