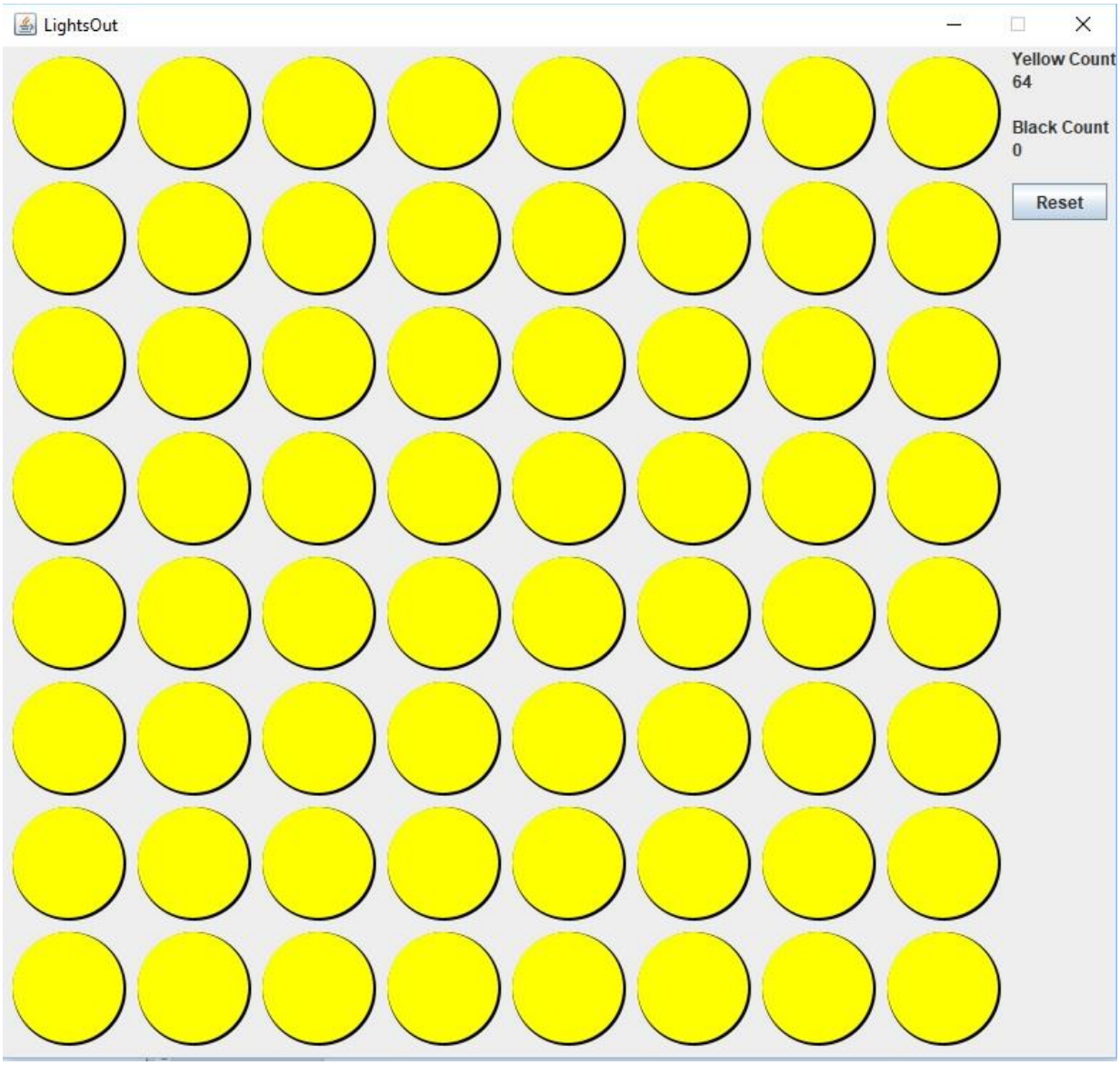
CompSci 251

Assignment 9

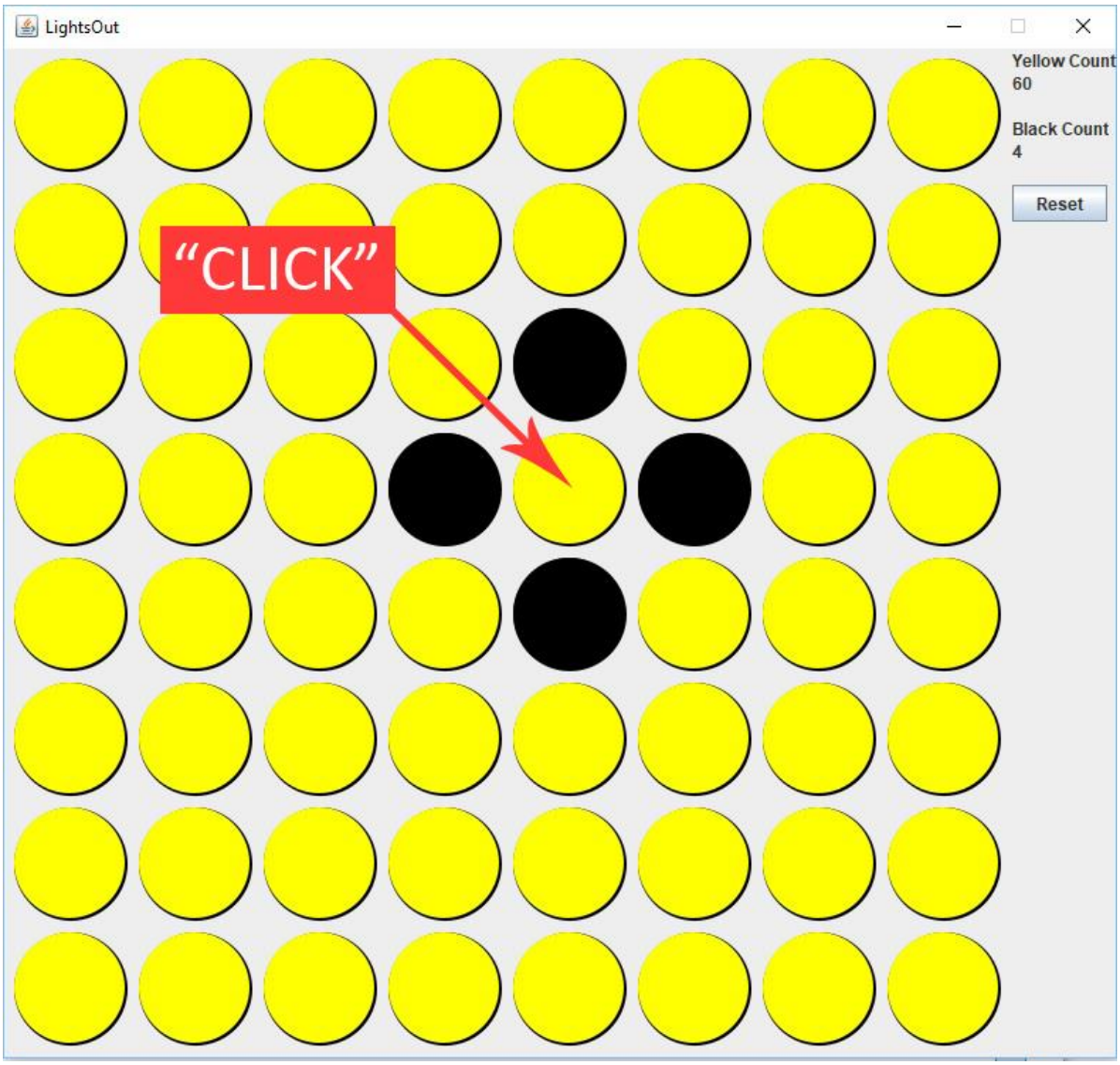
Due Dec. 13th, 2018

Introduction

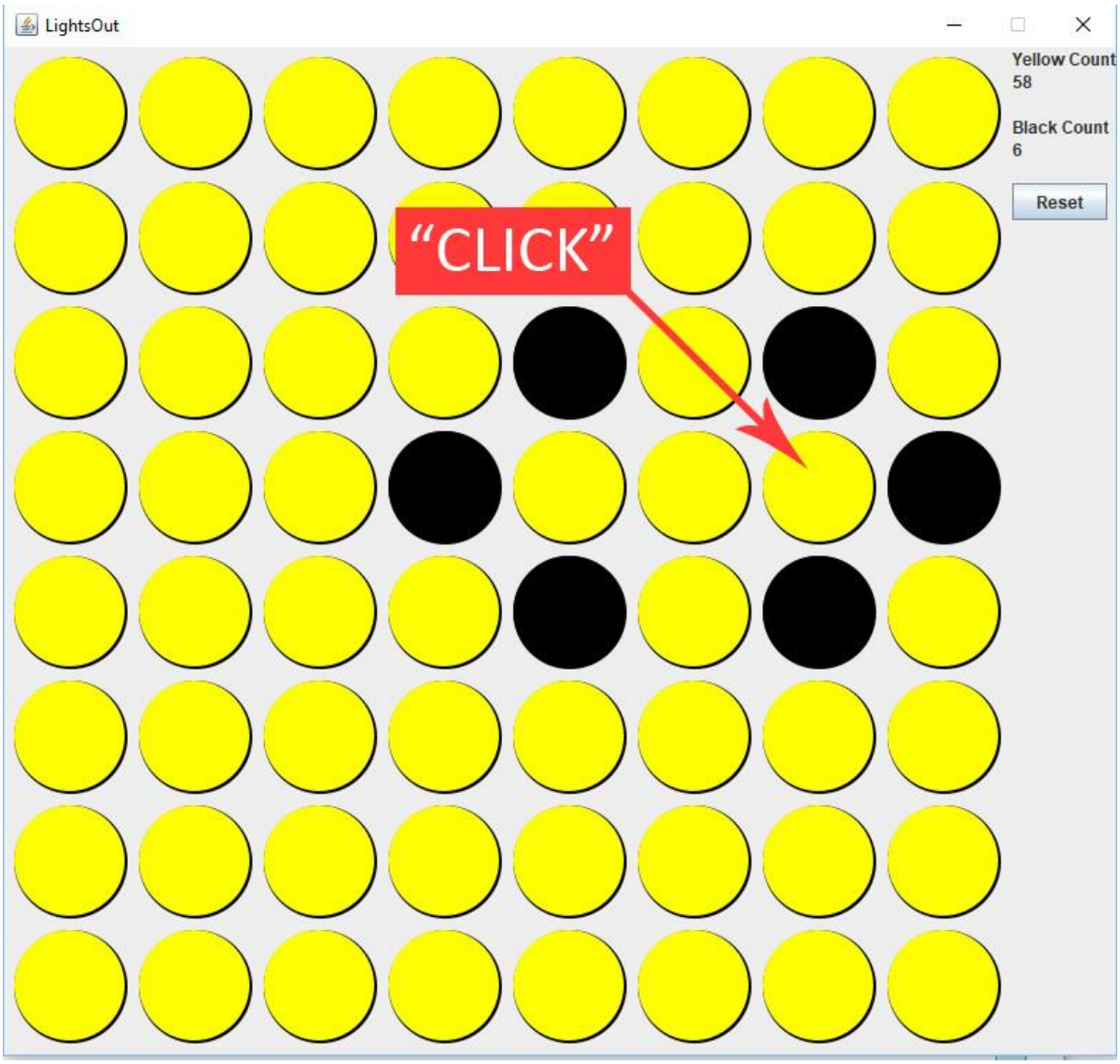
You will create a game called Lights Out using the Java Swing library. The basics of the game are simple, a square grid of yellow circles is displays to the screen. You goal is to turn them all black. Below is an image of what the game looks like during the start.



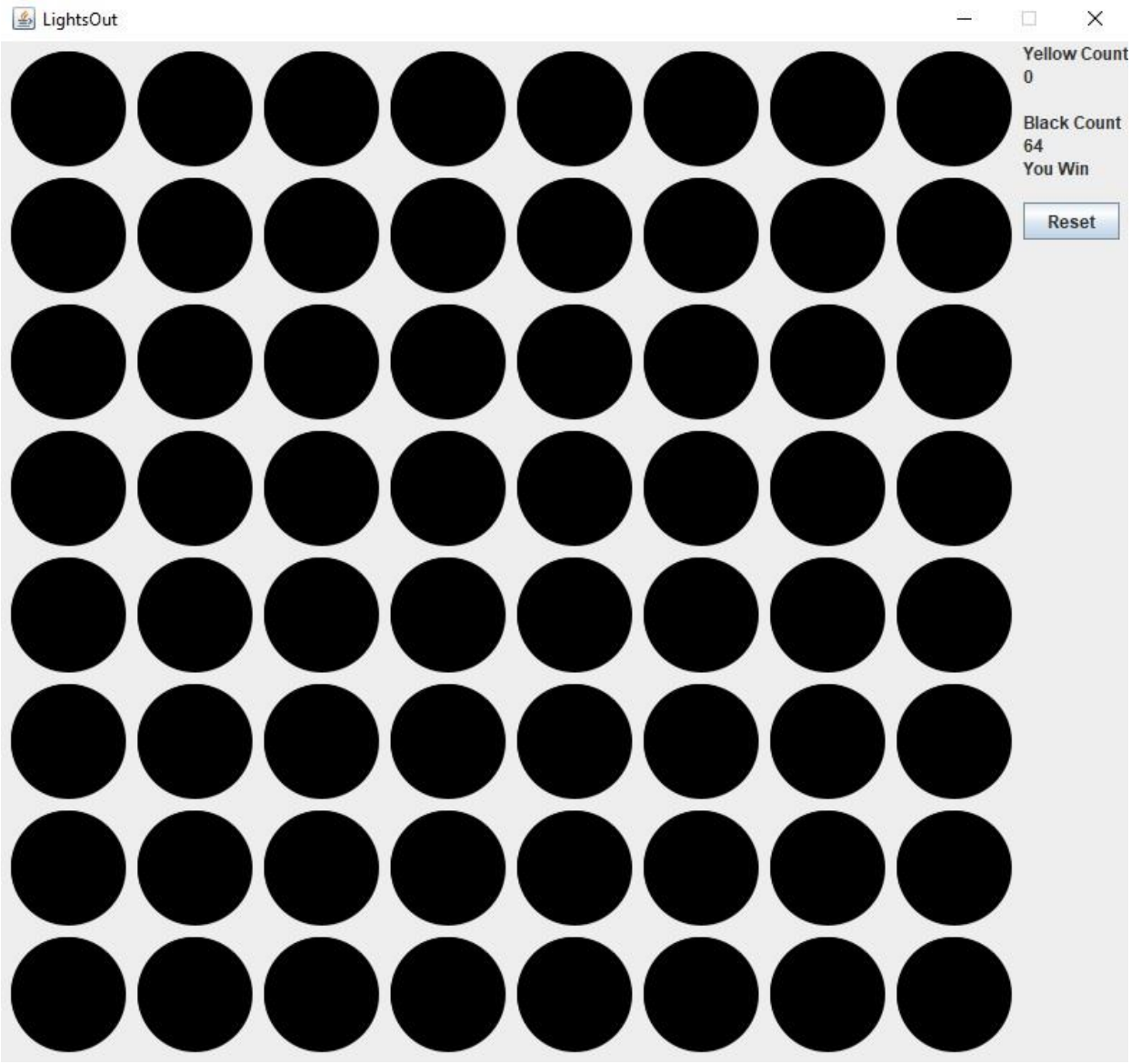
Notice you have a 8x8 grid where all the buttons are yellow. There also is a yellow count in the upper right corner displaying the total number of yellow circles, and a black count showing the total number of black circles. Lastly, there is a reset button, which when clicked will return the game to this state.



Above shows what happens when a button is clicked. Notice the circles directly above, below, to the left, and to the right have flipped to black but the circle that was clicked stays the same. Also notice the counts have changed in the upper righthand corner to correctly show the counts.



Building off the previous example, assume we click the next button as indicated above. Notice a similar thing happens, all circles that are directly up, down, left and right to the button that was clicked gets flipped. Here the up, right, and down circles changed to black, but the left circle changed back to yellow! And accordingly, the counts are updated to show the correct counts.



The goal of the game is to turn all circles black. When the game is won, you are no longer allowed to change any circles, meaning if you click a circle, nothing changes. Also notice the text “You Win” is displayed in the upper righthand corner. The only way to replay the game is to click reset, where the game sets all circles back to yellow and resets the count.

What you need to do

This program will have four classes, three you will have to write, and one I have provided for you.

Class LightsOutCircle (Provided – Do NOT modify)

- A simple class that represents a circle. The circles can be yellow or black, nothing else.
- Pay close attention to the methods in this class, they will be useful.
- You do not need to do anything to this class.

Class GameBoard

- This class will be an 8 x 8 grid of LightsOutCircle objects. I suggest creating a variable that will hold the size of your grid. This will make it easy to modify how many LightsOutCircle's will be on the game board.
- You must:
    - Add all LightsOutCircle's in a grid format
        - Take note of the constructor for a LightsOutCircle object.
        - I suggest changing the layout of the class and creating an instance variable to "mimic" your grid.
    - Write a ActionListener to perform the logic of the game as described in the examples above.
        - Watch out for out of bounds exceptions!
    - Provide a method that resets the board of the game.
    - Keep track of total number of yellow and black circles.
    - Determine if the game is over when all the circles are black. If so, the user can not change any more circles.

Class ControlPanel

- This class will display the yellow count, black count, reset button, and the you win tag.
- You will need to use JLabels and JButtons.
- I suggest using a BoxLayout to have everything appear in a stacked fashion.
- You will need an ActionListener to have the reset button function correctly.

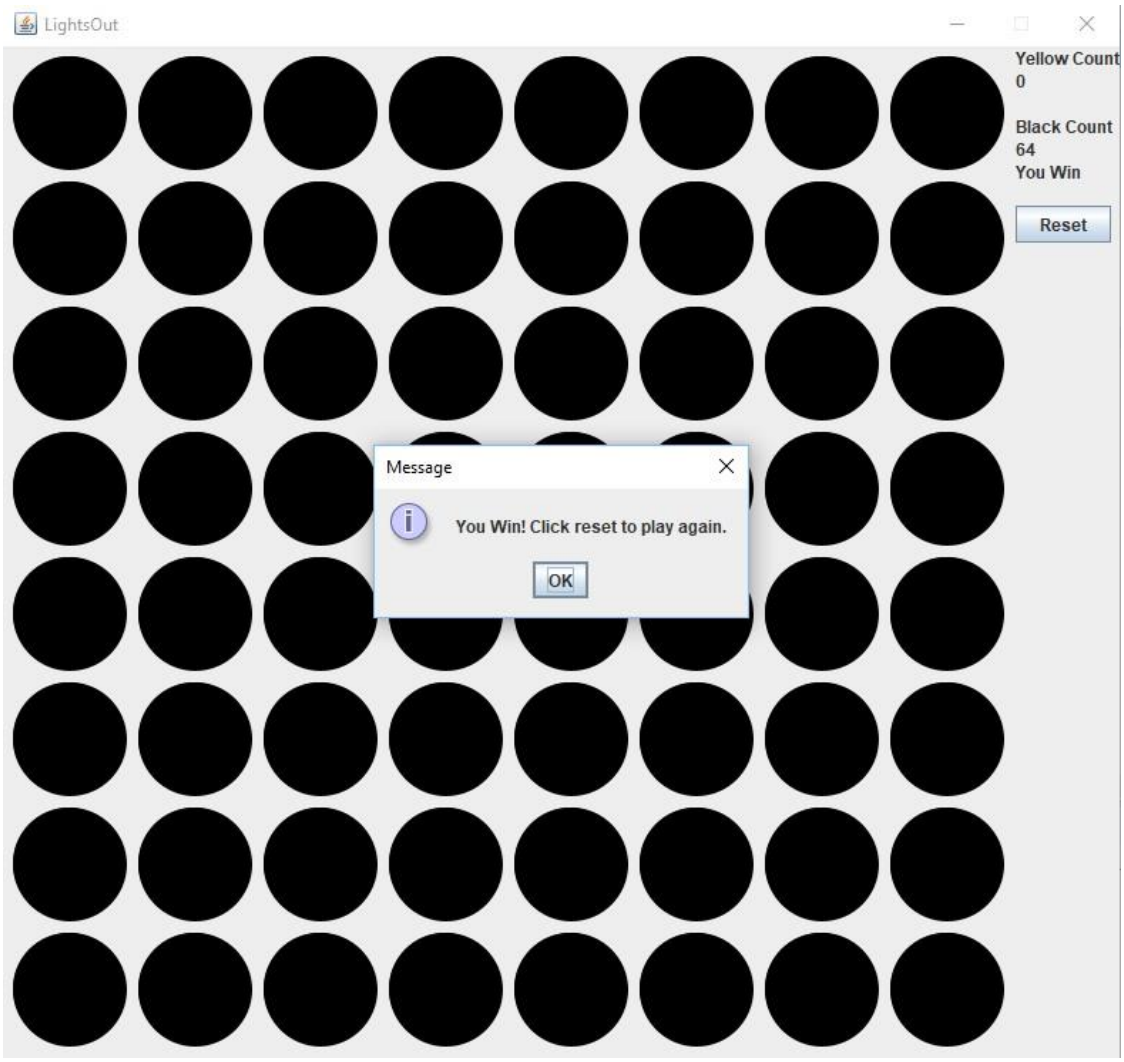
Class LightsOut

- This class will be the container for a GameBoard and a ControlPanel.
- It will contain the main method of the game.
- You must:
    - Set the title.
    - Set the size (800 x 750 seems to works well).
    - Add a GameBoard (Hint: it should be the center).
    - Add a ControlPanel.
    - Make it so the window is not resizable.
    - Make it so the program ends when closed.

There is one thing I did not mention during the class descriptions, you must figure out a way to have your GameBoard and your ControlPanel "talk" to each other. The GameBoard keeps track of yellow and black circle counts, but the ControlPanel displays these numbers! You've also noticed the ControlPanel holds the reset button, but reset is a method of the GameBoard! All variables in your GameBoard and ControlPanel must be private. Can you think of a way to make this work? How can one object use another in a "has a" relationship? Aggregation would be the correct choice here by the way.

Graduate Students

Display a message to the screen as shown below. I'll let you figure out how to do this. It's actually a fairly simply piece of code when you figure it out.



Hope you all enjoyed the class and you enjoy your winter break!