

- Both partners should grab a piece of paper or create an empty document.  
The scribe should draw the following table.  
Each of these variables are *instance variables* of a Student object.

Driver: Alan Xiao

Scribe: Froilan Zarate

name	totalScore	quizCount
Phil	$\theta \Rightarrow 8$	$\theta \Rightarrow 1$
Joe	$\theta \Rightarrow 12$	$\theta \Rightarrow 2$
Mama	$\theta \Rightarrow 21$	$\theta \Rightarrow 3$

The driver should draw the following empty table. This will hold the values of variables that are local to the addQuiz method.

score	newTotalScore	newQuizCount	(return averageScore)
8	$8 + \theta$	$\theta + 1$	$8 / 1 = 8$

score	newTotalScore	newQuizCount	(return averageScore)
4	$4 + 8$	$1 + 1$	$12 / 2 = 6$

score	newTotalScore	newQuizCount	(return averageScore)
9	$9 + 12$	$2 + 1$	$21 / 3 = 7$

- The scribe should continue to read the instructions for both partners.  
The driver should open a copy of the code for the Student [class](#) on screen so you can refer to it.
- The scribe will 'create' the Student object with name 'Phil' by adding the values of the instance variables to their table after this line is executed:  
`Student phil = new Student("Phil");`
- Emulate the execution of the addQuiz method. For each step, write down what the code would look like for that step:
  - The scribe will choose a number y between 1-10 (randomly).  
`phil.addQuiz(y);`
  - The driver will write a variable 'score=y' where y is the number selected. Into their xtable

- ```
addQuiz(int score)
```
- c. The driver will create two local variables `newTotalScore` and `newQuizCount`
  - d. And ask the scribe what their values should be
 

```
double newTotalScore = score + totalScore;
int newQuizCount = quizCount + 1;
```
  - e. The 'addQuiz' method will add the input to the new total score, and increment the number of tests by one. (in the paper, erase the old value, and write a new value where the old value was).
 

```
newTotalScore+=score;
quizCount+=1;
```
  - f. The 'addQuiz' method will tell the Student Object to update the `totalScore` and `quizCount`

```
this.totalScore = newTotalScore;
this.quizCount = newQuizCount;
```
  - g. The 'addQuiz' method will exit. What this means is that you need to delete all created and passed variables. So either cross out or delete the entire table. If you call this method again, create a new table.
  - h. The 'Student' object STILL EXISTS. This means that after the `addQuiz` method exits, your variables will persist.
5. Do step 4 at least two times. Swap roles between executions of 'addQuiz'  
You only need to record one iteration of this activity.
  6. The person playing as the method will look at another method 'getAverageScore'. It has no parameters.
  7. Execute the 'getAverageScore' method
    - a. Method asks the class for the two values 'totalScore' and 'quizCount'.
 

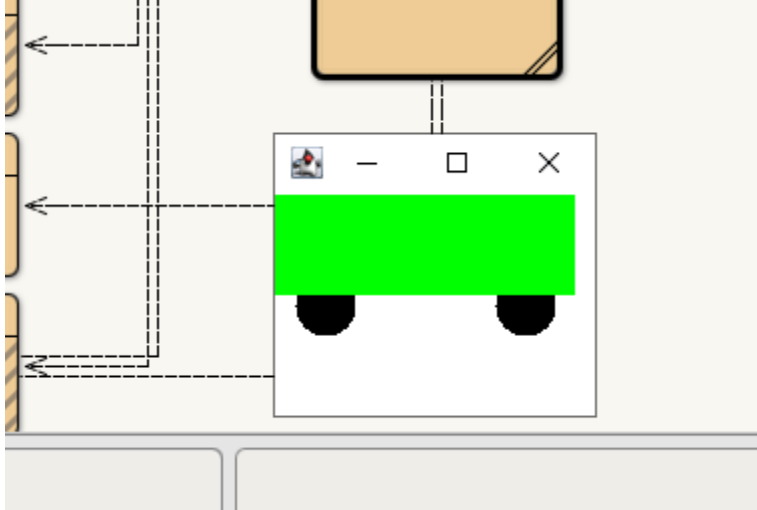
```
double totalScore = this.totalScore
int quizCount = this.quizCount
```
    - b. The person playing the method will execute the 'getAverageScore' method and will tell the 'Student' object what number was returned
 

```
return totalScore / quizCount;
```

Student object: Was it the average of the quiz scores?

## Some more advanced uses of graphics in Java

In this lab, you'll create a more advanced shape than just a rectangle or circle. You'll make a car!



Now we will use your implementation of `OneNewCar` and use it to create a new 'Car' class that makes a car for us when we provide a given input.

**Driver: Froilan Zarate**

**Scribe: Alan Xiao**

Create a new class `Car` with three instance variables. `Int`, `int`, `Color.color`.

There are two ways to show a color. You can create a color by creating it like an object. Or you can reference it with the pre-set colors.

For example: `Color redColor = Color.RED` or `Color blueColor = Color.BLUE`.

**Scribe: In your lab report, list how many different color options are present in the graphics package. For lab credit, just use the preset colors. If you have more time, feel free to try to make a car whose colors you like better.**

**There are 13 colors available in the preset colors package**

Driver: Create another class called `carTester` and copy the code from [here](#)

If your output is the following, you are finished with the lab!

**Driver: Take a screenshot of these three cars over your code.**

OneNewCar XCar XColor XRectangle XCarTester X

CompileUndoCutCopyPasteFind...CloseSource Code

```
/**
 * Creates a car with two wheels.
 *
 * @author Alan Xiao & Froilan Zarate
 * @version 09-23-2022
 */
public class Car
{
    // instance variables - replace the example below with your own
    private int x;
    private int y;
    private Color carColor;

    Rectangle carBody;
    Ellipse backTire;
    Ellipse frontTire;

    /**
     * Constructor for objects of class Car
     *
     * @param x the initial starting position of the horizontal position.
     * @param y the initial starting position of the vertical position.
     * @param bodyColor the color of the car.
     */
    public Car(int x, int y, Color bodyColor)
    {
        carBody = new Rectangle(x, y, 150, 50);
        backTire = new Ellipse(x + 10, y + 40, 30, 30);
        frontTire = new Ellipse(x + 110, y + 40, 30, 30);
        carBody.setColor(bodyColor);

        backTire.fill();
        frontTire.fill();
        carBody.fill();
    }
}
```

Class compiled - no syntax errors

saved