

CS5004-5005-SV – Lab 4 – May 28, 2020
Practice with Classes

The goal of today's lab is to practice creating classes, instance variables, and methods, to firm up key ideas (info hiding, encapsulation, and overloading).

0. Make a copy of the PersonText.java code from Lecture 4, in a new folder. Verify that it works as expected.

a. Modify it as follows:

Instead of the age being an instance variable, it should use the person's year of birth (yob). The getAge method should compute the age based on the current year and the person's year of birth. To find the current year, see: <https://www.javatpoint.com/java-date-getyear-method>. Fix setAge to update year of birth, taking into account the current year. Check whether the new age is "reasonable" and handle appropriately. For example, you may assume that the person's age has to be non-negative and less than 120.

b. Add String instance variables for firstName and lastName. Add getters and setters for each.

c. Correct the default constructor to set their firstName and lastName to the empty string.

d. Change the **toString** method to include their full name, age, and height, with appropriate formatting (e.g., "Mark Miller, age 70, height 65").

e. Create an **equals** method to consider two objects to be equal if both are instances of class **person** and all of their instance variables are equal.

f. Create examples in your main method to test the above changes, including both true and false cases of equality, unreasonable age values, and so on. A setter should return false, and not change anything, if the proposed value results in an unreasonable situation.

1. Make a copy of the sample solution I shared for Tic.java from Lab 03 in a new folder. Verify that it works as expected.

a. Modify the method setBoardCell to first check whether the value to be stored is one of "X", "O", or "_". It should also check that the number of "X"s equals, or is 1 greater than, the number of "O"s. The

method should return false and make no change if these conditions are not met.

- b. Create a Boolean method, winner(player). Assume the input parameter should be a String, either "X" or "O". (It is not necessary to worry about possible other values at this time.) Use the Boolean operators && and || to determine whether the player has filled any row or column or either diagonal.
 - c. Create test examples within the main method for the new code you have written.
2. In a new folder, create a class, Fraction.
- a. Make two versions of a constructor for it. The first version takes a numerator and a denominator. The numerator must be an **int** and the denominator must be a *positive int*. If the denominator is ≤ 0 , throw `IllegalArgumentException`, like this:

```
if (denominator <= 0) {  
    throw new IllegalArgumentException("Denominator not positive");  
}
```

The second version should be a default constructor. The default for the numerator is 0. Since we require the denominator to be non-negative, its default value should be 1.
 - b. Create getters and setters for the numerator and the denominator. The setter for the denominator should be a Boolean method that returns **false** and does nothing if presented with a non-positive input.
 - c. *[Optional if time permits]* Create a method, toDouble, that returns the value of the fraction as a double. For example, 2/4 should return 0.5.
 - d. *[Optional if time permits]* Create a method, isEqualTo, which takes another instance of Fraction as input. It should determine whether the two fractions represent the same value. (Hint: cross-multiply.)
 - e. *Create tests in your main method for each of the above.*

Submit answers to above items to Canvas for CS5005 (due Thu June 4 midnight). Only your .java files need to be uploaded.

3. Work on homework assignment PS3. If finished, **submit to Canvas for CS5004** (due Mon June 1 at 2 AM).