**CS201 Lab 7**

**40 points**  **Due**: 03/23/15

**Problem:** Your friend just bought a new house, but all of the rooms have carpet or laminate that is horribly dirty and gross. They need your help to know how much it will cost to re-do all of the floors! They want to put hardwood or carpet in each of the rooms of their house. They have all of the dimensions of the rooms, which are conveniently all rectangles!



**Purpose:** This lab gives you practice with:

* Designing and programming methods
* Checking user input
* Testing code with methods

**Details:**

You need to design, write, and test a program that calculates the cost of buying flooring for your friend’s house. You need to let the user input the information for as many rooms as they want. For each room you need to find out from the user the width (in terms of feet as a real number), length (in terms of feet as a real number), and flooring type choice. Then you need to calculate the cost for that room by using the following costs:

* Hardwood costs $1.39/sqft
* Carpet costs $2.50/sqft

Unfortunately, you don’t really trust your friend to follow the directions in your program, so you need to *protect your program from bad user input*. You should keep asking them for a double until they give you a double for the number input, and when asking for the type of flooring you should keep asking them until they enter carpet or hardwood. Be sure to have clear prompts and error messages!

**Design:**

In your algorithm file you are given a set of methods to use to solve this problem, but main and floor choice still need work on their algorithms. Read through what is there so far and make sure you understand it, and then work on each method until your algorithm makes sense as a whole. *You must show your method design to Dr. Olsen before you may start coding.*

**Use Scanner for all input and System.out for all output.**

**Steps:**

1. Complete the Algorithm that is started for you in your repository.
2. Create a new Java file.
3. Write your Java code. Write one method at a time, and make sure it compiles.
4. Create a set of test cases for each method. Be sure to list the description to make it clear why you are testing the chosen set of input values. You must have at least 2 test cases for each method other than announce (announce only needs to be tested to ensure that it runs).
5. Test your methods using your test cases.
6. Write comments in your code to make it clear what it is doing.
7. Write comments for each method in your code. See the example at the end of this file.
8. Include an updated version of the header comments. Many lines should change!

**Submit:**

1. To GitHub:
   * Your completed algorithm
   * Your .java file
   * Your test cases
2. On paper in class:
   * A short (250 words or less) individual reflection about what you did in lab, what it was like working with your partner, and what you had the most trouble with. (1 per person)

**Method Commenting Example**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: Read in all of the file, summing the numbers

\* Parameters: Scanner – Scanner for reading from file

\* Algorithm: For each value in the file, read in and add to total.

\* Return value: int – the total from the file

\*/

public static int sumFromFile( Scanner scan )

{

int total = 0;

while(scan.hasNext())

total += scan.nextInt();

return total;

}