CS113 Basic Data Structures + Algorithms
Homework #1

Clue Assignment (50 HW points)

Dr. Foo has been murdered! Detective Jill must determine the murderer, crime scene, and weapon. There are six possible murders (numbered 1 through 6, Professor Plum to Mrs. Peacock), 10 locations (1 through 10, ballroom to cellar), and six weapons (1 through 6, lead pipe to spanner). Detective Jill tries to guess the correct combination. Each guess is a theory. She asks her assistant, Jack, to confirm or refute each theory.

Use the **hw1files.zip** found on Blackboard for code to test your answer. Inside you will find:

- *Theory.java* = encapsulates murdered, crime scene, and weapon data
- *TheoryItem.java* = convenient methods/constants for translating integers
- *AssistantJack.java* = created with answer set, contains method to check your guesses (theory)
    - Answer set 1 = 1, 1, 1
    - Answer set 2 = 6, 10, 6
    - Answer set 3 (or any other integer) = randomly chosen theory
- *RandomClue.java* = Driver that tests theory with a simple strategy: use random guesses for each

Note that Assistant Jack will randomly tell you which of your guesses are wrong, if multiple are wrong. So if you ask Assistant Jack to check your answer with the same set of guesses, you may get a different item that is wrong which means that both the first and second items returned are wrong. When checking guesses with Assistant Jack, your number of guesses (times you ask to check the potential answer) is kept track of. Your goal is to write a program as if you were Detective Jill, which finds the solution with fewer than 20 checks for any of the three answer sets above.

For example, you are provided RandomClue.java, a driver that randomly guesses each weapon, location, and person, until the correct solution is found. Run the program a couple of times for each theory (1, 2, 3) and answer the following questions:

- How often do you get a solution in fewer than 20 guesses? Does this change depending on the theory you test (1, 2, 3)?
- What is good or bad about this strategy?

Answer the above questions for your implementation as well. Place all answers in .txt file in the project.

Include Lab1 and Lab2 from Class. (10 LA points)

1. Lab1 – Write a program that prints the numbers from 1 to 100, but for multiples of 3 print “Fizz” instead of the number and for multiples of 5 print “Buzz”. For numbers that are multiples of both three and five print “Fizzbuzz” (note the lowercase ‘b’).
2. Lab2 – Write a method that finds out if a number is a power of two (without using bitwise operators). Create a tester program to test a couple of values to prove your method works.

Pro-tip: In Eclipse, go to Project > Generate JavaDoc… *and click on the checkbox next to your project twice so its an actual checkmark, not a dash. Click on* Finish *(yes on any dialogs that popup). Under your project, open the* doc *folder, then right click on* index.html *and select* Open With… > Web Browser*, you should see a tab in Eclipse open with a web browser window of the documentation (you can also open this in your own web browser) to easily navigate the documentation there, rather than through the code.*

Create separate Eclipse projects for each problem (HW1 Clue, Lab 1, Lab 2) and commit+push to your repository before the submission deadline.