# WORKSHEET #9

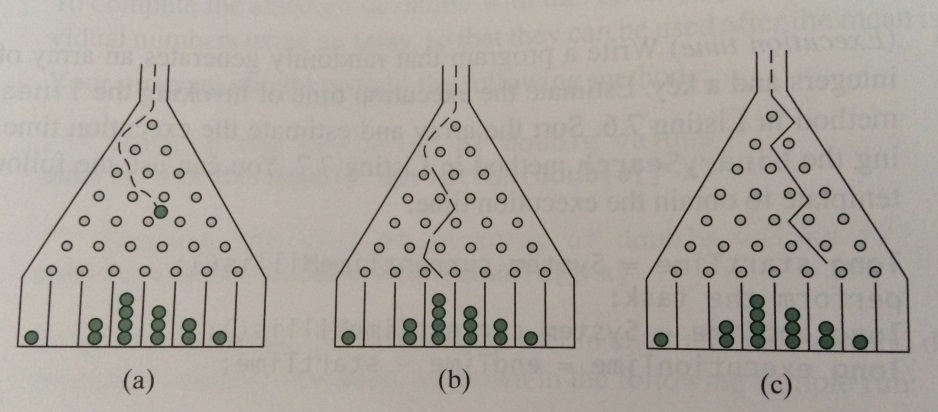
## Assumes: Ch1, Ch2, Ch3, Ch5, Ch6, Ch7, Ch8

1. Following Object Oriented methodology, write a service class that has an array of ints as its instance variable called **theNumbers**. The default constructor fills the array with random numbers between 10 and 99. The service class should have appropriate accessor, mutator, toString and equals methods implemented. It also has the following “business” methods:
   1. public int[] eliminateDuplicates() – the method creates and returns a new array which is a copy of the instance variable theNumbers but with all duplicated values eliminated
   2. public boolean[] hasSecretNumbers(int… numbers) – the method returns a boolean array of the same size as the instance variable theNumbers, with values set to true for all the elements that are the same as the passed numbers, and false for the remaining numbers
   3. public int indexOfSmallestElement() – the method returns the index of the smallest element in the instance variable array theNumbers
   4. public int[] countOccurencesOfEachElement() – the method returns an array of integers that contains counts of each number between 10 and 99 in theNumbers array (some numbers will have count of 0)
   5. public int countMultiplesOfTen() – the method returns the number of elements in theNumbers array that are multiples of 10
   6. public double computeDeviation() – the method computes the standard deviation of the values in the instance variable theNumbers. The following formulas should be utilized:

To compute standard deviation your method should call the following method that you also need to implement: public double computeMean().

Write a client program that thoroughly tests your methods

1. The bean machine, also known as quincunx or the Galton box, is a device for statistics experiments named after English scientist Sir Francis Galton. It consists of an upright board with evenly spaced pegs in a triangular form as shown below:

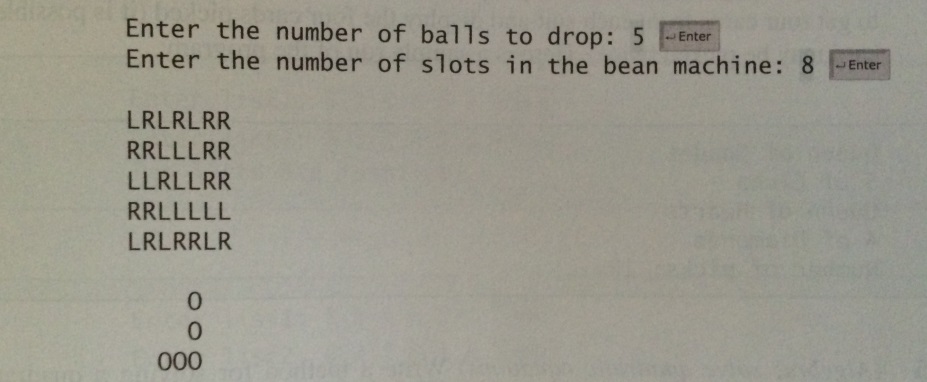


Balls are dropped from the opening of the board. Every time a ball hits a nail, it has 50% chance of falling to the left or to the right. The piles of balls are accumulated in the slots at the bottom of the board.

Using Object Oriented methodology, design classes that will simulate the bean machine. Start with the UML diagram. How many classes do you need, what objects should be utilized, what methods do you need? Focus on **WHAT** is needed, then, once the UML diagram is ready, focus on **HOW** to implement each method.

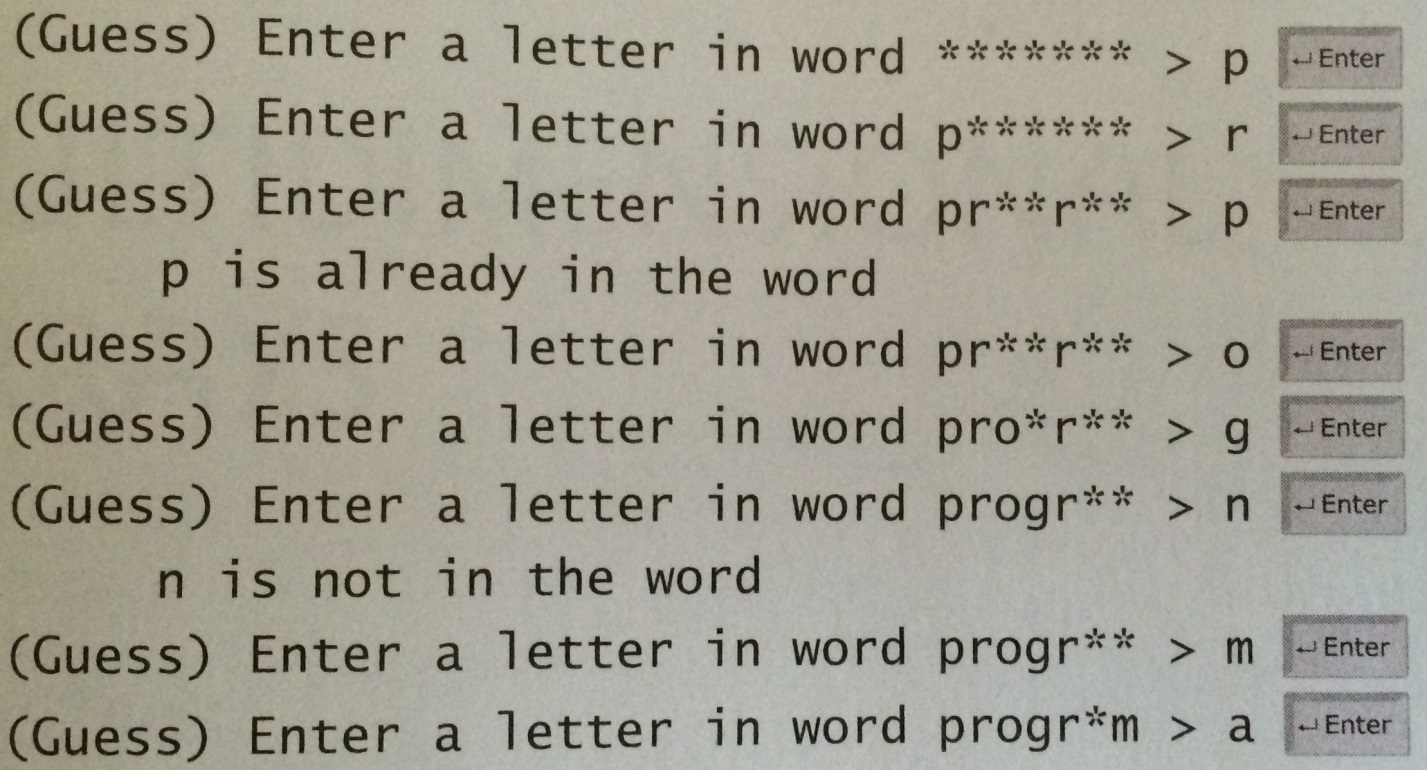
The client should prompt the user to enter the number of balls and the number of slots in the machine. These values should be passed to your class constructor!

The output of the program should give the paths for each ball. For example, the path for the ball (b) is LLRRLLR and the path for the ball (c) is RLRRLRR. See the sample run below:



HINT: one of the instance variables that you need is an array named slots. Each element in slots stores the number of balls in a slot. Each ball falls into a slot via a path. The number of Rs in a path is the position of the slot where the ball falls. For example, for the path LRLRLRR, the ball falls into slots[4], and for the path RRLLLLL, the ball falls into slots[2].

1. Using Object Oriented methodology, write a hangman game that randomly selects a word from the predefined list and prompts the user to guess one letter at a time. To start with, each letter is displayed as an asterisk. When the user makes a correct guess, the actual letter is displayed.



When the user finishes the word, the program should display the number of misses and ask the user if (s)he wants to play another round. Once the game is over the program should display the final statistics: the number of rounds played, the number of guesses in each round, the number of misses in each round, and the best round. An array to store words should be declared as an instance variable in one of your classes. For example:

private String[] words = {“write”, “program”, “java”, “csuci”};

1. ☺