**CSE1341 - Lab 4 Assignment**

*Arrays, Strings, more practice with multiple methods*

**PRE-LAB (10 POINTS)**

Write a Java program called SmallestIntegerwhich does the following:

1. Prompts the user to enter 10 integers
2. Uses a **for** loop to store the integers in an array
3. Calls the method returnSmallest with the following header

**public static int returnSmallest(int[] array)**

to return the smallest integer in the array

1. Displays the smallest integer value

Bring the working SmallestInteger.java and SmallestInteger.class files to your lab session for pre-lab credit.

**SAMPLE OUTPUT:**

**$ java SmallestInteger**

Enter 10 integers: 5 3 8 0 -10 16 100 -20 25 1

The smallest integer is -20

**LAB (90 POINTS)**

Complete the problems below. *Note:* For this and all lab assignments you may **not** collaborate with anyone *except* the instructor, TA, or CSE help desk tutors when completing these assignments. Also, you **may not** include all or part of coded solutions developed by others in your submission.

Submit the *java* and *class* files via Canvas (as a single zip-file). Include a comment block at the top of your *Java* file that includes your name, student id number, and “Lab 4-Fall 2017”.

NOTES: Comment your program to explain your steps. Each program should compile without errors and should run to produce outputs described for each exercise. The following points will be discounted if the related element is missing or incorrect:

* Reasonable output formatting [20 points]
* Proper names for classes and variables [15 points]
* Comments [15 points]
* Program doesn't compile [ 20 points]
* Source code (java file) missing [ 15 points]
* Executable (class file) missing [15 points]
* Missing usage of Strings and Arrays [40 points]

**Problem 1 (45 points)**

The bean machine, also known as 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 nails (or 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 a 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.

Write a Java program called BeanMachine that simulates the bean machine. Your program should

1. Prompt the user to enter the number of the balls and the number of the slots in the machine.
2. Simulate the falling of each ball by printing its path. For example, the path for the ball in part (b) of the figure is LLRRLLR.

*Hint:* Create an array named **slots**. Each element in the array 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. **(Bonus/Optional + 10 pts)** Display the final buildup of the balls in the slots in a histogram as shown below.

Here is a sample run of the program:

**> java BeanMachine**

Enter the number of balls to drop: 5

Enter the number of slots in the bean machine: 8

RLLRLRR

RLRLLLR

LLLLRRR

LLRRLRL

RLRLRLL

   O

   O

   O

   OO

**Problem 2 (45 points)**

Write a hangman game that randomly generates a word and prompts the user to guess one letter at a time. Each letter in the word is displayed as an asterisk. When the user makes a correct guess, the actual letter is then displayed. When the user finishes a word, display the number of misses and ask the user whether to continue to play with another word. Declare an array to store words as follows:

**//Add any words you wish in this array**

**String[] words = {“write”, “that”, …**}

*Hints:*

1. To retrieve a hidden word from the array **words**, randomly generate an index based on the length of the array.
2. Create a String array called **guessedWord** representing the guessed word and store asterisks in it. The length of the guessWord array should be equal to the length of the hidden word.
3. To display the array **guessedWord**, use the following:

**System.out.print("(Guess) Enter a letter in word " + String.join("", guessedWord) + " > ");**

Here is a sample run:

|  |
| --- |
| (Guess) Enter a letter in word \*\*\*\*\*\*\* > p  (Guess) Enter a letter in word p\*\*\*\*\*\* > r  (Guess) Enter a letter in word pr\*\*r\*\* > p  p is already in the word  (Guess) Enter a letter in word pr\*\*r\*\* > o  (Guess) Enter a letter in word pro\*r\*\* > g  (Guess) Enter a letter in word progr\*\* > n  n is not in the word  (Guess) Enter a letter in word progr\*\* > m  (Guess) Enter a letter in word progr\*m > a  The word is program. You missed 1 time.  Do you want to guess another word? Enter y or n > n |

**This assignment is due by 6:00AM Saturday, October 14, 2017.**