C212 Lab 4

Intro to Software Systems

Objectives:

- Using Conditional Statements
- Using loops
- Random Number generator
- Junit Testing

Instructions:

- Please review the requirements given below carefully and complete your work. You should compress all source files (including main to test your work) into a zip file and submit it through Canvas. The grading scheme is provided on Canvas.
- You are free to use any looping construct of your choice
- Make sure that you display appropriate messages that are intuitive and clear
- Use Java's random number generator method (limit it to integers only)
- Do **NOT** write the entire logic of the program inside main(). Create separate methods that perform all the work and then invoke it from main()
- Write Junit Tests for **Exercises 2 and 4** only.

Lab Requirements

1. **Secret Number:**
   Write a program that chooses a random number between 1 and 10 (including 10) and asks the user to guess the number. After each incorrect guess, the program will tell the user whether the secret number is higher or lower than their guess. Once the user guesses the number correctly, the program will tell the user how many guesses it took them to get the correct number.

   A Note on Random:
   For problem 1, you'll need to use the java Random class, and more specifically, the Random.nextInt() method. This method takes one argument, which is an integer specifying an upper bound, and returns a random integer between 0 and the upper bound (but not including the upper bound). For example:

   ```
   Random rand = new Random();
   int myInteger = rand.nextInt(5);
   ```

   After this code, myInteger will be populated with a random integer, either 0,1,2,3, or 4.

   You can read the following documentation for more information:
   Random (Java Platform SE 8) (oracle.com)

2. **Exponent Calculator:**
   Write a method *exponentCalculator(int base, int exponent)* that takes in two integers and returns the base integer raised to the power of the exponent. **Do not use Java's built in power function.** Instead, use looping to find the returned value. Write Junit tests for this method.

   Edit:
   Since we are dealing with integers, your program should not produce an output if the user enters a negative number for the exponent value. Instead, your program should print an error statement and the loop should not run.

3. **Designing Patterns**:
   Create a menu-based program to generate the given patterns. A menu-based program is a program that provides the user a "menu of options" and user has a choice to select any item from the menu. The program will not stop till user chooses to exit the program with option 'E'. Program should only accept 'A', 'B', 'C', 'D' and 'E' as valid options.

   The program should then choose a random number between 3 and 10, which will determine the size of the pattern. The program should tell the user what the randomly selected number is before printing the pattern.

   Patterns with size 5 are shown below.

   A.

```
    *
   **
  ***
 ****
*****
 ****
  ***
   **
    *
```

   B.

```
5
5 5
5 5 5
5 5 5 5
5 5 5 5 5
```

   C.

```
* * * * *
*       *
*       *
*       *
* * * * *
```

D. Make your own pattern (different than the ones given above)

Example Menu:

A. Pattern 1
B. Pattern 2
C. Pattern 3
D. Pattern 4
E. Exit

Note: You MUST NOT hardcode the patterns and you must use loops (or nested loops) to print the patterns of given size. Be sure to print the empty spaces appropriately to make the pattern look like the ones shown above.

4. Write a java method (with appropriate method name, return type and argument lists) that takes two integers as arguments and prints a box of #'s with width and height determined by the arguments, respectively.

   For example, calling myMethod(3,4) would yield

```
###
###
###
###
```

Do's And Dont's for this Lab

Here are some tips and tricks for lab4
Note: No arrays or any other data structure (arraylist, hashmap, sets, linkedlist, stacks, queue) is allowed

Problem 1.
Since you don't know how many guesses it will take the user to guess the correct number, you may want to consider using a loop.

Problem 2.

- Make sure to account for edge cases (like if either of the numbers is 0)

Problem 3.
- Note that this program will always terminate when the user hits 'E'
- After printing the pattern again display the menu to the user.
- Take care of printing the spaces in the patterns
- The pattern should look exactly how it does in the pdf.
- You have to use nested loops to do this problem.
- An easier way to approach this problem is by breaking down a bigger pattern into smaller parts.

Problem 4.
-This problem is like the one before it, and you should approach it in a similar way