CIS351-Java conditionals and loops Lab

Submission Instructions

Submit a zip folder containing completed NameConverter.java and Stars.java

Objective

Practice with decisions statements and loops in Java.

Bakground

If you have not finished the reading assignment for Tuesday and Wednesday, then please finish them before attempting the lab, which may help you to complete this labwork.

Part 1: Working with input and decisions in java

Many documents use a specific format for a person's name. For this part, you need to write a **java class** named `NameConverter` that takes in a person's name, as input, in a particular format and displays the person's name in a different format.

If the input name has the form:

`firstName middleName lastName`

then output is:

`lastName, firstInitial.middleInitial.`

If the input name has the form:

`firstName lastName`

the output is:

`lastName, firstInitial.`

Input test cases and expected output

The NameConverter class should be able do the following:

Case 1: Ex: If the input is:

`Pat Silly Doe`

the output is:

`Doe, P.S.`

Case2: Ex: If the input is:

```
Julia Clark
```

the output is:

```
Clark, J.
```

Instructions

1. Create a java class in JGrasp and save it as `NameConverter.java`
2. Now declare a class named NameConverter within this file
3. Next, write the main method. Remember how main method in Java is written from your first lab.
4. The first task of your program should be to take input of the user name and save it as a String variable. We will be using `Scanner` class of Java to take input from the user. For this you need to import the Scanner class in your NameConverter java file. Include the following statement at the top of your java file-

   `import java.util.Scanner;`

   Next you need declare an object of the Scanner class and use nextLine() method to take input from the user. Write thw following 2 lines of code within your main method.

   `Scanner scnr = new Scanner(System.in);` //declares object of Scanner class

   `String userInput = scnr.nextLine();` //allows the user to enter a String(i.e. name) and saves it in userInput variable which is a String variable

5. Now you need to use different methods of String class to convert the name stored in `userInput` variable in your desired format.
6. A list of all String methods can be found here

   HINT: pay careful attention to following two methods of String which might be useful for this program - indexOf(), charAt() and substring()

Part 2: Working with loop in java

1. For this part of the lab, you will be working with the provided `Stars.java` class.

2. Your task is to **edit** the `Stars.java` class, so each of the star patterns listed below can be printed using loops. As you finish each pattern, test it and then move on to the next one.

3. Be sure that you are thinking through the pattern and most specifically how the stars and spaces relate to the row and column values. It is sometimes easier to use a special character, say #, instead of the spaces in the star pattern, so that you can easily count how many spaces you are printing.

4. After printing each pattern, you should print a blank line, so the output looks clean.

ADVICE: Before starting to code, I recommend you compile and execute the Stars.java file to observe the outout pattern corresponding to the GIVEN example code. Now, study the GIVEN example code in Stars.java file carefully until you understand how the pattern is built using for loops.

Pattern A

Add code to your program to carry out pattern A as follows:

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

The leftmost stars are in the leftmost output column.

Pattern B

Add code to your program to carry out pattern B as follows:

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

The leftmost star of the last row is in the first position of the output column. **HINT:** You used starCnt in the GIVEN example of Stars.java pattern. You should think about using a spaceCnt variable as well.
ADVICE: Also, while developing the code, you may replace space with another character which is visible. This will help you to see what is happening in the code. Once your code is complete, you can replace that character with space.

Pattern C

Add code to your program to carry out pattern C as follows:

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

The top row, leftmost star is in the first position of output.

Grading Criteria

Completed NameConverter.java - 40pt and Completed Stars.java - 60pt

Farzana Rahman / frahman@syr.edu