

Homework #7

Introduction to String manipulation, keyboard input, and more practice with Java methods that take parameters and return values.

1. Aim

The purpose of this lab is to:

- a. Introduce you to String processing.
- b. Introduce you to writing code to read input from the keyboard.

It is recommended that you read and understand all the instructions below before starting this exercise.

2. Files Needed

You will not be provided with any starter files for this homework. Create a project in Eclipse for this assignment. For each exercise below, create a Java class with the appropriate name. When you create each program, be sure to have Eclipse create the public static void main method for you. Be sure to include the standard header comments at the top of each file.

3. To be Handed In

The file **MadLib.java** should be sent to me when you have them completed. Be sure to name the files/classes exactly as specified.

4. Exercises

Part I: MadLib.java

Problem Statement

Write a program that asks the user to enter the following information:

- (1) their first name (string data),
- (2) their last name (string data),
- (3) their hometown, including city and state (also string data),
- (4) their age (integer data), and finally
- (5) a profession (string data).

The program should then print out the following **three** lines, with the user's input replacing the items in bold underlined italics:

There once was a person named **fname lname** who lived in **hometown**.
At the age of **age**, **fname** moved to Nashville and joined Remnant Fellowship.
fname grew up and went to work as a successful **profession**.

For example, if the user entered ***John*** for the first name, ***Blue*** for the last name, ***Sioux Falls, South Dakota*** for the hometown, ***21*** for the age, and ***computer programmer*** for the profession, then the output would be

There once was a person named John Blue who lived in Sioux Falls, South Dakota.
At the age of 21, John moved to Nashville and joined Remnant Fellowship.
John grew up and went to work as a successful computer programmer.

Your output should be exactly three lines long, just as in this example. All lines should contain the user input in the appropriate places. Remember to put in all the necessary punctuation and spaces (though the user is expected to include a comma between the city and state names). To avoid excessively long Java statements in your program, you should use a combination of println() and print() statements to produce the output. [Note that it is considered bad style to have excessively long lines in your Java program. Most programmers do not write code beyond column 80 of the editor window. In this class we relax this a bit and allow you to go to column 100.]

You are also required to capitalize the first and last names; i.e., changing the first character of the name to upper case and all remaining characters to lower case. This will ensure that if the user entered *john* or *JOHN* for the person's first name, the program will still print *John*. This can be accomplished with a combination of calls to the String methods `substring()`, `toUpperCase()`, `toLowerCase()`, and the concatenation operator. You should write a static method call `capitalize` that takes a String parameter and returns a String result that performs the capitalization for you. [Do not attempt to capitalize the hometown, as it will consist of multiple words (a city and a state).]

Note that the first and last names are expected to be single words, and thus you should use the `next()` method to read those data items in rather than using the `nextLine()` method. Since the hometown contains multiple words and potentially the profession contains multiple words, you will need to use the `nextLine()` method to read those in. The age is integer data and thus would be read with the `nextInt()` method.

Example Executions (user input in green)

```
Enter your first and last name: cLAirE doe
Enter your hometown, including city & state: Kansas City, Missouri
Enter your age: 18
Enter a profession: human resource manager
```

```
There once was a person named Claire Doe who lived in Kansas City, Missouri.
At the age of 18, Claire moved to Nashville and joined Remnant Fellowship.
Claire grew up and went to work as a successful human resource manager.
```

5. Additional requirements

- A. You must start your program with header comments that provide your name, email address, the date the program was last modified, and a short description of the program.
- B. You should use a consistent programming style. This should include the following.
 - a. Meaningful variable & method names
 - b. Consistent indenting
 - c. Use of "white-space" and blank lines to make the code more readable
 - d. Use of comments to explain pieces of tricky code

See the code examples in the class text for a good formatting style.

6. Submission for grading

The files **MadLib.java** should be sent to me when you have them completed. Be sure to name the files/classes exactly as specified.

7. Grading

This lab is worth 15 points. Your grade will be based on whether your solution is correct or not, and on how closely you followed the directions above. Remember that programming style will now be a part of your grade (on this and all subsequent assignments).