CalPolyPomona

CS 1400 - Lab #2

Maximum Points: 10 pts.

Topics

- Working on "String" class and some of its methods.
- Familiarization with basic data types.
- Using the Scanner Class
- Printing output
- Find the correct way of string comparisons.
- Getting familiar with Control Statements (If, else)

Coding Guideline (You will be graded on this)

1) Download the template file Lab2.java from Canvas and fill-in-the-blanks to create your Java program.
2) Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
3) Keep identifiers to a reasonably short length.
4) Use uppercase for constants. Use upper camel case for classes. Use lower camel case for all other identifiers (variables, methods, objects).
5) Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches, and loops. Be consistent with the number of spaces or tabs that you use to indent.
6) Use white space to make your program more readable.
7) Use comments to explain how the parts of your program work.

Problem Description

You will have to write a Java program that asks the user to enter two strings, firstName and lastName, and concatenates the strings to make a full name. The program will use methods of the class `String` like `length()` and `toUpperCase()` on the full name and compare strings by `equals()` method and if-else statements.

Sample Output

Below is an example of what your output should roughly look like when this lab is completed. The red words are user inputs.

Note: when your program runs on Gradescope, you will NOT see any input like the red text above. If you use `print()` to show prompts, your output might be on the same line. To avoid this issue, please use `println()` instead of `print()` to show the prompts.

Sample Run 1:

```
Please enter first name: magnus
Please enter last name: carlsen
Full name (in capitals): MAGNUS CARLSEN
Length of full name: 14
String comparison using "==" sign does NOT work
String comparison using "equals" method works
```

Sample Run 2:

```
Please enter first name: wesley
Please enter last name: so
Full name (in capitals): WESLEY SO
Length of full name: 9
String comparison using "==" sign does NOT work
String comparison using "equals" method works
```

Instruction

Step 1: Getting Started

Create a class called **Lab2**. Use the same setup for setting up your class and main method as you did in previous labs and assignments. Be sure to name your file **Lab2.java.**

At the beginning of each programming assignment you must have a comment block with the following information:

```
/*--------------------------------------------------------------------------------------------
// AUTHOR: YOUR NAME
// FILENAME: TITLE OF THIS SOURCE FILE
// SPECIFICATION: DESCRIPTION OF THIS PROGRAM
// FOR: CS 1400- Lab #2
// TIME SPENT: HOW LONG IT TOOK YOU TO FINISH THIS LAB
//------------------------------------------------------------------------------------------*/
```

Your code should also have the class definition and one main function as follows:

```
// class name should match the file name
public class Lab2 {
     // we must have a main method to run the program
     public static void main(String[] args) {
          // Your main logic goes here
     }
}
```

Please, make sure you put code inside the main method all the time (except that the code location is pointed out).

Step 2: Declaring Variables and User Input

When we examine this programming task, we see that we will need three `String` variables: `firstName`, `lastName` and, `fullName`.

To store the length of the full name, we also need an integer variable `nameLength` of type int. For the user input, we will use the `Scanner` class. In total, you should have at least 5 variables, which store 3 strings, 1 integer, and 1 Scanner respectively.

An example is showed as follows.

```
// declare variables of different types
String firstName = "";
String lastName = "";
String fullName = "";
int nameLength = 0;
Scanner scan = new Scanner(System.in); // Don’t forget to import the Scanner class

// Use Scanner to ask the user for first name
System.out.println("Please enter first name: ");
firstName = scan.nextLine();
// Use Scanner to ask the user for last name
System.out.println("Please enter last name: ");
lastName = scan.nextLine();
```

To use Scanner, don’t forget to import Scanner from `java.util` package. This code snippet should be on the top of your program and outside class definition.

```
// All imports have to be outside class
import java.util.Scanner;

// class name should match the file name
public class Lab2 {
     // we must have a main method to run the program
     public static void main(String[] args) {
         // something here...
```

Step 3: Full Name, String Manipulation

Part #1: Concatenation

Now that we have both first and last name, we need to form the full name from them. Remember that string concatenation can be done using '+' sign between variables. Form `fullName` by adding `firstName` to `lastName` separated by one space.

```
 // Example: ("abc" + " " + "def"); gives you "abc def"
 // Add "firstName" to "lastName" variables using "+" sign, don't forget the space.
// store the result in the "fullName" variable
// -->
```

Part #2: Convert to upper case

Now convert `fullName` content to uppercase. Remember we use `toUpperCase()` method from the `String` class to do so.

```
// Example: "abc".toUpperCase(); gives you "ABC"
// Convert "fullName" variable to upper case and store it back to itself
// -->
```

Part #3: Find length of a String

Remember the `length()` method is used to find number of characters in a String variable. Use `length()` to find the length of `fullName` and store result in `nameLength` variable.

```
// Example: "hello".length(); gives you an integer 5.
// Find the length of "fullName" and store it as "nameLength" variable.
// -->
```

Part #4: Display results

Print out the `fullName` and `nameLength` on screen. Use `System.out.println()` to do that. Always look at the Sample Output section (below) to make sure your output look like the expected output.

```
// Print "fullName", it should be in upper case
// -->
// Print "nameLength", this should be number of characters
// in "fullName" variable, including space
// -->
```

Step 4: String Comparison

For String data types, you can compare two variables to check if both hold the same value or not. There is a fast way to do that using "==" sign. However, this method does not guarantee to give correct results for all String variables. The better and more accurate way for String comparison is by using the `equals()` method. Follow the code below to see the difference between using both ways of comparison.

If you are using the template, the only thing you need to do is to print what is requested below on lines 9, 12, 16, and 19 on lines 10, 13, 17, and 20 respectively. Observe the difference between “==” and “equals()”.

```
1. // Define two String variables, title1 and title2 using
2. // String constructor to initialize them
3. String title1 = new String("cse110");
4. String title2 = "cse110";
5.
6. // Compare the two strings and print which one of the two ways works
7. // follow code below:
8. if ( title1 == title2 ) {
9.      // Print "String comparison using "==" sign works"
10.     // -->
11. } else {
12.     // Print "String comparison using "==" sign does NOT work"
13.     // -->
14. }
15. if ( title1.equals(title2) ) {
16.     // print "String comparison using "equals" method works"
17.     // -->
18. } else {
19.     // print "String comparison using "equals" method does NOT work"
20.     // -->
21. }
```

Step 5: Submit Your Lab2.java to Gradescope

Please submit **ONLY** the file Lab2.java to the “Lab 2” link on Gradescope. Make sure it is compiling and producing the expecting outputs. You are done.