Ira A. Fulton Schools of Engineering
Arizona State University

CSE 110 - Lab 2

Lab Topic: Strings

- 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)

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 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 text are user inputs.

Note: When the program runs in submission system, 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
```

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 lab. 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: <Please put your name here>
// FILENAME: Lab2.java
// SPECIFICATION: <Describe your program>
// FOR: CSE 110 - Lab #2
// TIME SPENT: <Estimate time to complete this work>
//-------------------------------------------------------------*/
```

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

```java
// 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 add your code inside the main method.

Step 2: Declaring Variables and User Input

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

To store the length of full name, we also need an integer variable *`nameLength`* of type int. For the user input, we will use *`Scanner`* from Lab1. 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.

```java
// 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

// 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` pacakage. 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...
```

Practice strings:
Write the results of each expression with **Strings in "quotes"** and **characters in 'single quotes'.**
*Hint: Strings index starting at **0**. A String with 10 characters has the indices 0-9!*
You do not need to turn this part for lab2: is just practicing methods from class String

```
String str1 = "Java Programming";
String str2 = "Learning programming is cool";
    str1.length() ->
    str2.charAt(0) ->
    str1.indexOf("o") ->
    str2.toUpperCase() ->
    str1.substring(5) ->
    str2.substring(3, 14) ->
    str2.replace("o", "aa") ->
```

Step 3: Full Name, String Manipulation

Part1: Concatenation

Now that we have both first and last name from user input, 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 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
// -->
```

Part2: Convert to upper case

Now convert `fullName` to upper case. Remember we use `toUpperCase()` method in String class to do so.

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

Part3: Find length of a String

Remember the method `length()` in String class. It 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.
// -->
```

Part4: 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 tendency to use "==" sign. However, this method does not work correctly for String variables since Strings are objects and not primitive data types. You should compare two strings using `equals ()` method. Follow the code below to see the difference between using == versus the method.

If you are using the template, the only thing you need to do is put in the print functions and observe the difference between "==" and "equals()".

```
// Define two String variables, title1 and title2 using
// String constructor to initialize them
String title1 = new String("cse110");
String title2 = "cse110";

// Compare the two strings and print which one of the two ways works
// follow code below:
if ( title1 == title2 ) {
    // Print "String comparison using "==" sign works"
    // -->
} else {
    // Print "String comparison using "==" sign does NOT work"
    // -->
}
if ( title1.equals(title2) ) {
```

```
    // print "String comparison using "equals" method works"
    // -->
} else {
    // print "String comparison using "equals" method does NOT work"
    // -->
}
```

Step 5: Make sure to upload your Lab2.java

Please submit your Lab2.java ONLY to “Lab 2 Upload Link” in Weekly Labs section. Do not include any Eclipse project files.