
ASU Computer Science and Engineering Department

CSE 110 - Lab 2

What this Lab Is About:

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

Use the following Coding Guidelines:

- When declaring a variable, you usually want to initialize it.
- Remember you cannot initialize a number with a string.
- Remember variable names are case sensitive.
- 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.
- Use white space to make your program more readable.
- Use comments after the ending brace of classes, methods, and blocks to identify to which block it belongs.

Problem Description: Manipulating Strings

For this Lab, you will have to write a java program that asks the user to enter two strings, first name and last name. Concatenate them to form a full name. Use some methods of class "String" like *length()* and *toUpperCase()* on full name. Find the correct way for comparing strings using *equals()* method. Learn how to use *if-else* statements.

Please follow these guidelines and ask your TA for help if needed. You may also collaborate with your classmates if needed.

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**.

Assignments Documentation:

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

```
/*-----  
// AUTHOR: your name.  
// FILENAME: title of the source file.  
// SPECIFICATION: your own description of the program.  
// FOR: CSE 110- Lab #2  
// TIME SPENT: how long it took you to complete the assignment.  
//-----*/
```

Step 2: Declaring Variables and user input

When we examine this programming task, we see that we will need two inputs from the user: `firstName`, `lastName` and `fullName`. All of them are **String** type.

In order to calculate the length of `fullName` we need an integer variable `nameLength` of type **int**.

Copy the code provided below, this code is similar to Lab1, make sure you understand it and it has no errors. Remember to take care of indentation.

```
// All imports has 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){

// declare variables of different types:
String firstName = "";
String lastName = "";
String fullName = "";
int nameLength = 0;
Scanner scan = new Scanner(System.in);

// 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();
```

Step 3: Full name, String manipulation

Part1: 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 **space**.

```
//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 variable. Remember we use `toUpperCase()` method in String class to do so.

```
// Convert "fullName" variable to upper case
/-->
```

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 it to find the length of `fullName` and store result in `nameLength` variable.

```
// Find the length of "fullName" and store it
// in 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 fast way to do that using "==" sign. However, this method is not guaranteed to give correct results for all String variables. The better and more accurate way for String comparison is by using *equals()* method. Follow code below to see the difference between using both ways of comparison.

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

Sample Output

Below is an example of what your output should roughly look like when this lab is completed.

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

Last Step: Submit your lab by following the instructions below:

- Submit your Lab2.java file to the Submission Server. Go to the Submission Server site, <https://courses.eas.asu.edu/cse110b/login>, then click on Lab Submissions in the left frame. Click on the browse button and find where you saved your Lab2.java file (and not the Lab2.class file) on your computer. Upload the file to the site and then click on the Submit button.
- Your file will be submitted and a screen will show up displaying if your program compiled and what your output is when run on some sample input (in this case nothing).
- You should then check to make sure that the actual file submitted properly and is readable to the grader. To do so click on Grades in the frame on the left of the page and then click on the 0 underneath **Lab2**. You will again see that your program compiled and the sample output, but you should scroll down to the bottom of the screen and make sure your file is readable as well.

This study resource was shared via CourseHero.com