

**CSCI 1301 – Programming Principles I**  
**Georgia Southern University**  
**Department of Computer Science**  
**Fall 2024**

**Lab 5**

**Point Value: 20 points**

**Due: Friday September 20, 2024, end of lab**

**Objectives**

1. Use built-in mathematical functions in Java to write working programs.
2. Explore the properties of characters and Strings within Java.
3. Use characters and Strings in Java to write working programs.
4. Use formatted Strings to control output for a Java program.
5. Use the debugger to effectively diagnose and fix errors in code.
6. Archive Java source code files and other documents into a zip file.

**Description**

Complete the following steps as described and upload requested materials to Folio in the appropriate dropbox in a single zip file. Submit Java source code files to Gradescope.

- Students should work in groups of two for this lab. When submitting your zip file, fill out the comment header and indicate both group member names.
- Your submitted zip file should be named Lab#LastName1LastName2.zip. For instance, Lab5JonesSmith. Any requested screenshots and/or answers to questions should be included in a separate Word document.
- All programs should have a comment header. That comment header should include information similar to:  

```
/**  
 * File: PAssign0.java  
 * Class: CSCI 1301  
 * Author: Christopher Williams  
 * Created on: Jun 6, 2016  
 * Last Modified: Aug 16, 2018  
 * Description: Display three messages to the console  
 */
```
- Before writing any code, for each problem, discuss with your partner how you plan to approach solving the given problem. Once you both agree on a way to attempt the problem, only then should code be written.
- Pay attention to names and any other requested material. Failure to include material will result in the loss of points as described below.

## Lab Problems

For all problems, use inline comments to describe major actions. Ask for assistance if needed.

1. Mathematical Functions – Write a Java program named Lab05Prob01.java that determines the area of a triangle given three floating-point side lengths entered by the user. To calculate this area, use Heron's (aka Hero's) formula.

Once the area is calculated, print all three side lengths as well as the area calculation with descriptive text. Do not use `System.out.printf()` for this problem.

**Expected Output (may be multiple values after decimal, input of 3.5, 4.5, and 5.5):**

For side lengths of 3.5, 4.5, 5.5, the area is 7.854885024620029.

Note that Gradescope will provide its own input for this program and your program should correctly calculate the values for the given input, not just your tested values.

After passing the Gradescope tests, show a working version of this program to the instructor before moving on.

**After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

2. Characters and Strings – Write a Java program named Lab05Prob02.java that asks the user to enter a String value and read it in using the Scanner's `nextLine()` method. Using Java character/String methods, determine and display the first character, the last character, and the "middle" character for any Strings that are 4 characters or longer. If the String is not long enough, display an error message, otherwise, display the requested information with descriptive text. Do not use `System.out.printf()` for this problem.

The "middle" character should be determined using the ceiling of the String's length divided by two. For a string  $s$ , the "middle" character would be at index  $\left\lceil \frac{s_{length}}{2} \right\rceil$ .

For instance, the String "chuckle" would have a middle index of  $\text{ceiling}(7/2.0) \rightarrow 7/2.0 = 3.5 \rightarrow \text{ceiling}(3.5) = 4 \rightarrow$  index 4 would be the character 'k'. Keep in mind that `Math.ceil()` returns a double and may need to be converted to the appropriate type to use correctly.

**Expected Output (4+ characters, input of chuckle):**

For the string chuckle  
first letter is c  
last letter is e  
"middle" letter is k

**Expected Output (Less than 4 characters):**

The string you entered (XXX) contains less than 4 characters

Note that Gradescope will provide its own input for this program and your program should correctly calculate the values for the given input, not just your tested values.

After passing the Gradescope tests, show a working version of this program to the instructor before moving on.

**After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

3. Characters and Strings, Swapping – Write a Java program named Lab05Prob03.java that allows the user to enter a five-letter word. If the word is shorter than or longer than five characters, display an error message and quit the program. If you receive the correct length word, swap the first letter and the last letter and then swap the second letter and the next-to-last letter with each other.

Accomplish this task by storing each character into a char variable and swapping the appropriate pairs. Reassemble the characters into a new String in a new variable.

**Do NOT just reorder the characters and form a new String, you must swap the appropriate pairs before creating the new String.** Using appropriate variable names will help accomplish this correctly. Pay attention to data types.

Print the original string and the newly reversed String to the console with descriptive text. Do not use System.out.printf() for this problem.

**Expected Output (5-letter word, input of zebra):**

For the string llama the reverse is amall

**Expected Output (non 5-letter word):**

The string you entered (cat) does not contain 5 characters

Note that Gradescope will provide its own input for this program and your program should correctly calculate the values for the given input, not just your tested values.

After passing the Gradescope tests, show a working version of this program to the instructor before moving on.

**After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

4. Formatting Strings – Update Problems 1, 2, and 3 to use `System.out.printf()` to format the output so that all relevant information is displayed to the user. Copy Problem 1 into a new file named `Lab05Prob04.java` and modify it so that all values displayed have exactly two decimal points. This updated program from Problem 1 is the only one that will be submitted to Gradescope for this problem and should be used as a model to complete the `printf()` conversion for the other problems.

**Expected Output (exactly two values after decimal, input of 3.5, 4.5, and 5.5):**

For side lengths of 3.50, 4.50, 5.50, the area is 7.85.

Note that Gradescope will provide its own input for this program and your program should correctly calculate the values for the given input, not just your tested values.

After passing the Gradescope tests, show a working version of this program to the instructor before moving on.

**After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

4. Demonstrate all working programs to the instructor to receive credit for each problem. Turn in each problem inside the zip file detailed below.
5. Name the zip file `Lab5LastName1LastName2.zip` (e.g. `Lab5JonesSmith`) and submit that zip file to Folio. Both students must submit the exact same zip file for either to receive a grade.

**Grade Breakdown**

Demonstrate running application to instructor:

Problem 1	4.5 points
Problem 2	4.5 points
Problem 3	4.5 points
Problem 4	4.5 points
Zipped source files (in Folio):	2 points
-----	
Total Possible:	20 points

Last modified: September 18, 2024