

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

Lab 3

Point Value: 20 points

Due: Friday August 30, 2024, end of lab

Objectives

1. Create an appropriate program for outputting simple messages to the console.
2. Experiment with variables, concatenation (with output), user input, and mathematical operations.
3. 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, Lab3JonesSmith. 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. Variables and User Input – Create a new Java program in a file named Lab03Prob01.java. Write Java code that prompts the user to enter their current accumulated credit hours as a university-level student. This value should be stored in a variable with an appropriate data type. Use a descriptive and meaningful name as an identifier for that variable.

Prompt the user again and ask for the number of total credit hours needed for their undergraduate degree. Refer to your department's website and the undergraduate catalog if you are unsure of this number. Follow the same guidelines for storing this value as you did with the credit hours. Do not create multiple Scanner objects.

Both user-entered values should be stored as whole numbers. Do not change their data type for Problems 2 and 3. Output both numbers to the console with meaningful, descriptive text as shown below.

Expected output (Assume input of 17 and 125):

You have 17 credit hour(s).

Your degree requires 125 credit hour(s).

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 or SA before moving on.

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

2. Mathematical Expressions – Copy and paste your code into a new Java program in a file named Lab03Prob02.java. Modify the program so that once the two values have been retrieved from the user and stored, the following calculations are completed and displayed to the user as described:

NOTES:

- Create a comment skeleton before coding this problem. Ensure that your comment skeleton represents a developed algorithm that has been agreed upon by both members of your group. Be smart/efficient about any values that are used/re-used when coding this problem. Do NOT remove your output from Problem 1.
- Determine and display how many credit hours are left before the user graduates.
- Assuming 15 credit hours per semester, determine and display how many semesters are left before the user graduates. The result/output should be expressed as a decimal number.

- Assuming 2 semesters per year, determine and display how many years are left before the user graduates. This should be expressed as a decimal number.
- Assuming 3 semesters per year (summer @ 15 credit hours as well¹), determine and display how many years are left before the user graduates if they take summer classes. This should be expressed as a decimal number.

Expected output (Assume input of 17 and 125):

You have 17 credit hour(s).

Your degree requires 125 credit hour(s).

You have 108 credit hour(s) until graduation.

You have 7.2 semester(s) (@ 15 credit hours/semester) left until graduation.

You have 3.6 year(s) (@ 2 semesters/year) left until graduation.

You have 2.4 year(s) (@ 3 semesters/year) left until graduation.

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 or SA before moving on.

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

3. Integer Division/Modulus Operator – Copy and paste your code into a new Java program in a file named Lab03Prob03.java. Using any previous calculations and as needed, additional calculations, modify your code so that it determines how many total years, total semesters, and remaining credit hours the student has left to graduate. Use the assumption that there are 15 hours per semester and 2 semesters per year.

For instance, if the student has 52 credit hours left, that is 1 year (2 semesters/30 credit hours), and 1 semester (15 credit hours), and 7 remaining credit hours left to graduate. Years and semesters should be whole number values only, not decimal values. Do not delete any output from Problem 2.

It may be helpful to review the DisplayTime.java file from Folio.

Expected output (Assume input of 17 and 125):

You have 17 credit hour(s).

Your degree requires 125 credit hour(s).

You have 108 credit hour(s) until graduation.

You have 7.2 semester(s) (@ 15 credit hours/semester) left until graduation.

You have 3.6 year(s) (@ 2 semesters/year) left until graduation.

You have 2.4 year(s) (@ 3 semesters/year) left until graduation.

You have 3 year(s), 1 semester(s), and 3 credit hour(s) (@ 2 semesters/year) left until graduation.

¹ You should NEVER take this many credit hours in a summer semester.

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 or TA before moving on.

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 Lab3LastName1LastName2.zip (e.g. Lab3JonesSmith) 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	5 points
Problem 2	5 points
Problem 3	6 points
Zipped source files (in Folio):	4 points

Total Possible:	20 points

Last modified: August 26, 2024