

# CSE 1321L: Programming and Problem Solving I Lab

Fall 2025

Lab 13

Module 7

## What students will learn

- o Input and Output handling in Java.
- o Selection Structures in Java (IF/ELSE IF/ELSE, and MATCH CASE)
- o Repetition Structures in Java (WHILE and FOR loops)

## Content

- o Overview
- o Lab 13A: Meal Selector
- o Lab 13B: Lab Grade Calculator

## Overview

Through this lab, you will practice applying the programming concepts we learned throughout the semester in a different language.

You will notice that some things in Java are very similar to what you have done in Python. Even though the syntax may be different, some of the underlying concepts such as Input/Output and Flow control structures are similar.

Remember that you must submit your programs written in Java, if your Lab 13 is written in python or in a file format other than **.java**, you will not receive points.

### **Final note: *Do not cheat***

If your temptation is to look online, don't. Come see us instead and ask questions – we are here to help. Remember, you are going to have to write code in your future job interviews, so learn it now to secure a high-paying job later.

Lastly, make sure you review the sample output and make sure the output of your program follows the exact same format including the input statements, print statement, etc. As always, user input is shown in **red** and **bold**.

## Lab 13A: Meal Selector

You are going to build a simple program that will prompt the user to ask some questions and try to suggest what meal to prepare.

The meal suggestion will be based on how much time the user has to prepare the meal, what is their “mood” (Healthy, Comfort, or Light snack), and if they prefer a vegetarian option.

Use the outline below as reference for the logic of your program:

- o **Healthy options:**
  - Less than 15 minutes and vegetarian: Hummus and veggie wrap
  - More than 15 minutes and vegetarian: Veggie stir fry with tofu
  - Less than 15 minutes: Chicken salad wrap
  - More than 15 minutes: Grilled chicken with vegetables
- o **Comfort options:**
  - Less than 15 minutes and vegetarian: Cheese quesadilla
  - More than 15 minutes and vegetarian: Mac and cheese
  - Less than 15 minutes: Ham and Cheese sandwich
  - More than 15 minutes: Pasta with meat sauce
- o **Light Snack options:**
  - Less than 15 minutes and vegetarian: Fruit and yogurt bowl
  - More than 15 minutes and vegetarian: Tomato soup with bread
  - Less than 15 minutes: Yogurt with granola and honey
  - More than 15 minutes: Egg drop soup with rice

### Requirements:

- o The program **must** implement both **switch case** statements and **if/else if/else** statements.
- o Use the **switch case** statements to match the “mood” (Healthy, Comfort, Light snack).
- o Use **if/else if/else** statements to determine the meal suggestion.
- o The **if/else if/else** statements should be nested in their respective cases of the **switch case** structure.
- o Prompt the use of time (in minutes) they have available. This value should be stored and handled as an **int**.
- o Prompt the user for the mood. This value should be stored and handled as an **int**.
- o Prompt the user to know if they prefer a vegetarian option. This value should be stored and handled as a **Boolean** type.
  - The user must enter either “true” or “false”, all in lowercase.
- o Use the input values to determine the meal following the outline describe above.

### Sample Output #1:

[Meal Selector]

How many minutes do you have to prep your meal?: 40

What is your mood?:

1. Healthy

- 2. Comfort
  - 3. Light Snack
- > 2

Vegetarian option? (true/false): false

You should try Pasta with meat sauce

### **Sample Output #2:**

[Meal Selector]

How many minutes do you have to prep your meal?: 10

What is your mood?:

- 1. Healthy
- 2. Comfort
- 3. Light Snack

> 3

Vegetarian option? (true/false): true

You should try Fruit and yogurt bowl

## Lab 13B: Lab Grade Calculator

Create a program that will be capable of calculating the grade of your labs. Also, the program should also be able to drop the lowest lab when calculating the lab average.

### Requirements:

- o The user will be able to repeat the lab grade calculation how many times they want to, so the program must include a **while** loop. We will call this our **program loop**.
- o On each iteration of the **program loop**, we are going to ask the user to enter their grade for all 13 labs.
  - You must a **for** loop for this part.
- o We will handle lab grade values as **float**.
- o We have not covered arrays yet, so we will handle the lowest lab grade like what you did for Lab10A.
  - Instead of looking for the highest value, we look for the lowest value.
  - **Hint:** Remember that the highest value a lab can have is 100.
- o The program must output the grade of the lowest lab, the average lab grade without the policy, and the average lab grade with the policy applied.

### Sample Output #1:

Enter your lab grades:

Lab 1: 100  
Lab 2: 100  
Lab 3: 100  
Lab 4: 0  
Lab 5: 100  
Lab 6: 100  
Lab 7: 100  
Lab 8: 100  
Lab 9: 100  
Lab 10: 100  
Lab 11: 100  
Lab 12: 100  
Lab 13: 100

The lowest lab grade was: 0.0

Your lab grade average is 92.30769

Your lab grade average after policy is 100.0

Do you want to try again? (Y/N): Y

Enter your lab grades:

Lab 1: 0  
Lab 2: 95  
Lab 3: 80  
Lab 4: 79  
Lab 5: 84  
Lab 6: 100  
Lab 7: 98.5  
Lab 8: 60

Lab 9: **78**  
Lab 10: **81**  
Lab 11: **83.5**  
Lab 12: **100**  
Lab 13: **100**

The lowest lab grade was: 0.0  
Your lab grade average is 79.92308  
Your lab grade average after policy is 86.583336  
Do you want to try again? (Y/N): **N**

## Submission Instructions:

- o Programs must follow the output format provided. This includes each blank line, colons (:), and other symbols.
- o Programs must be working correctly.
- o Programs must be written in Java.
- o Programs must be submitted with the correct **.java** format.
- o **Program submitted as python files will have an automatic 0.**
- o Programs must be saved in files with the correct file name:
  - Lab13A.py
  - Lab13B.py
- o Programs (source code files) must be uploaded to Gradescope by the due date.