# COMP1005/1405 Fall 2024 Tutorial 7

**Do not post the tutorial or the tutorial and/or solutions on any website.**

## Objectives

- Practice writing/running Python code in the terminal.
- Practice applying **Functions, Multidimensional Lists, and using modules** in the code.

## Expectations

To receive full grades for this tutorial, you must complete Problems 1-3.

## Grading Scheme for Tutorials

For each tutorial, you will be graded based on the following scale:

- 2/2 for demonstrating problems 1-3 and your **YourStudentNumber_T7.zip** file
  - Section A Tutorial Sessions: submit your work to the tutorial BrightSpace. The zip file should contain your solutions to all the required problems.
  - Section C, D, and E Tutorial Sessions: By the end of the tutorial session, demonstrate your tutorial work in person to a teaching assistant.
- 1/2 if you are missing problems or your solutions need significant improvement.
- 0/2 if you do not submit to BrightSpace or demonstrate to a teaching assistant
  - Section A Tutorial Sessions: no submission to the tutorial BrightSpace
  - Section C, D, and E Tutorial Sessions: not demonstrating your tutorial work to a teaching assistant by the end of the tutorial session

## Problem 1 (2D List)

Write a Python file named **multiList.py** in VSCode. No functions are required.

a. **Create** a two-dimensional list with three empty lists in it. At this point, your program must have a list like this [[ ],[ ],[ ]].
b. Using a **nested loop** **(can not be two individual for loops)**, store three random integers in the range [1,25], including each empty list. A sample list after this step should look like this:
   [[4, 17, 25], [7, 7, 16], [21, 17, 5]]
c. Using **another nested loop,** multiply all the numbers in this list and print the result. For the sample list above, the result printed should be **2379048000**.

## Problem 2 (Multidimensional Lists)

Write a Python file named **displayCustomer.py** in VSCode to be our self-created module that includes the following functions: (No global variables or constants should be used for the problem)

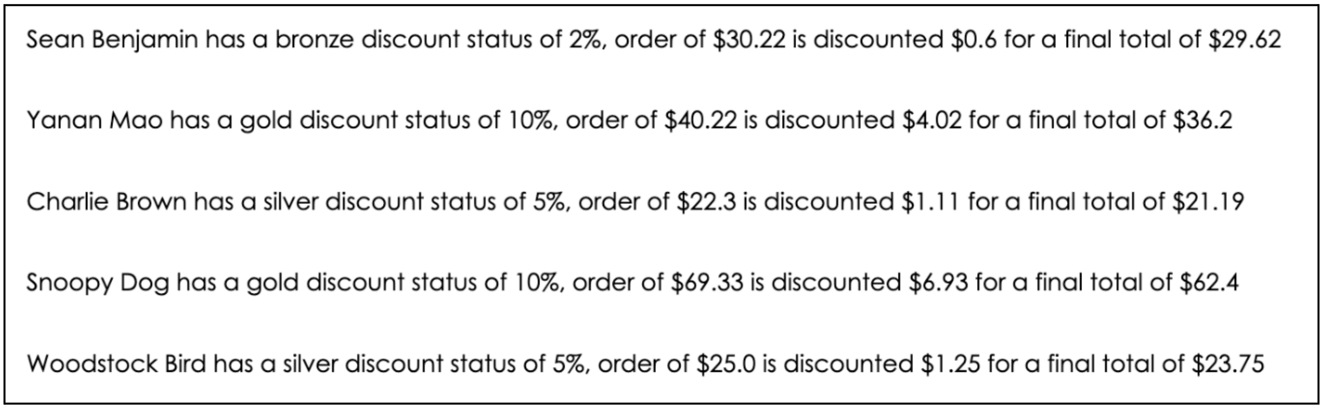
1. **printLists(customersList)**: (no return value needed)
   a. This function should take a list of lists as an argument.
   b. It should use a loop to iterate over each list and call the function printIndividualList(customer) to print the customer information for each list.

2. **printIndividualList(customer):** (no return value needed)
   a. This function should take a single list as an argument.
   b. The output depends on the user's discount status.
      i. Gold (G) users will receive a 10% discount on their order.
      ii. Silver (S) users will receive a 5% discount on their order.
      iii. Bronze (B) users will receive a 2% discount on their order.
   c. Print the elements in the list and show the discount info at the end.

3. **main() using the main guard**:
   a. Use the following list of lists that represent customer records:

**customerRecords = [[“Sean Benjamin”, “B”, 30.22],
[“Yanan Mao”, “G”, 40.22], [“Charlie Brown”, “S”, 22.30],
[“Snoopy Dog”, “G”, 69.33], [“Woodstock Bird”, “S”, 25.00]]**

b. Call function **printLists(customerRecords),** and the output should be :



```
Sean Benjamin has a bronze discount status of 2%, order of $30.22 is discounted $0.6 for a final total of $29.62

Yanan Mao has a gold discount status of 10%, order of $40.22 is discounted $4.02 for a final total of $36.2

Charlie Brown has a silver discount status of 5%, order of $22.3 is discounted $1.11 for a final total of $21.19

Snoopy Dog has a gold discount status of 10%, order of $69.33 is discounted $6.93 for a final total of $62.4

Woodstock Bird has a silver discount status of 5%, order of $25.0 is discounted $1.25 for a final total of $23.75
```

**Hint:**

- The program should apply a for loop to iterate through the list to retrieve a customer order. Example of a customer order: **[“Sean Benjamin”, “B”, 30.22]**
- Use a nest **if-else** or **if-elif-else** to determine the discount dependent on the discount status, which is the second value in the list (i.e., at position 1).
- You must use the f-string formatting specification (refer to lecture 9 on page 16.) in the print function to round and format dollar values to two decimal places.

## Problem 3 (Multidimensional Lists + Module)

Write a Python file named **`createCustomerList.py`** in VSCode to implement a discounted final order dependent on the user's discount status. **Note: no error checking on user input is required.**

a. **Import your displayCustomer module.**

b. **Create an createRecord() function:**

   i. Prompt the user for the customer’s name and append it to the list `customerInfo`.
   ii. Ask for the discount type (G, S, or B) and append it to `customerInfo`.
   iii. Get the cost of the order before the discount, convert it to a float, and append it to `customerInfo`.
   iv. **Return** the completed `customerInfo` list.

c. **Create a main() function:**

   i. Prompt the user for the number of customer records they want to insert.
   ii. Use a loop to create the required number of customer records.
   iii. Store these records in a list `customersList`.
   iv. Call a function `createRecord()` for each record to create customer information and add the returned customer record list to the list `customersList`. Hence creating a list of customer records.
   v. Use the `printLists(customersList)` function from the `displayCustomer` module (created in Problem 2) to display the customer records.
   vi. Ensure the script runs properly using a main guard (`if __name__ == "__main__":`).

## Compress Files (zip files)

1. Create a directory/folder and copy/move all your tutorial files.
2. Enter this directory, select all files, and create a zip file as described above. Name it **`YourStudentNumber_T7`** (replace **YourStudentNumber** with your 9-digit student number).

## Final Step

**For Section A (Submit the work before the tutorial ends):**

1. **Submit** your **zip** file to our Merged Tutorial Brightspace. The due date of your submission is aligned with your tutorial session.
2. **After** you submit the file, download your submission from Brightspace and confirm that it is a zip file containing **`multiList.py, displayCustomer.py,`** and **`createCustomerList.py`**. **Extract** the .py files and execute them again to ensure they work properly. Occasionally, a problem can occur during uploading, and files can become corrupted.

**For Sections C, D, and E (Show the TAs your work before the tutorial ends):**

1. **Problem 1-3:** Run your Python programs in VS Code to demonstrate they are working.
2. Answer the questions the TA may ask. Show the TA your zip file and extract the files.