CSE101

DUE OCT. 19

23:59 KST

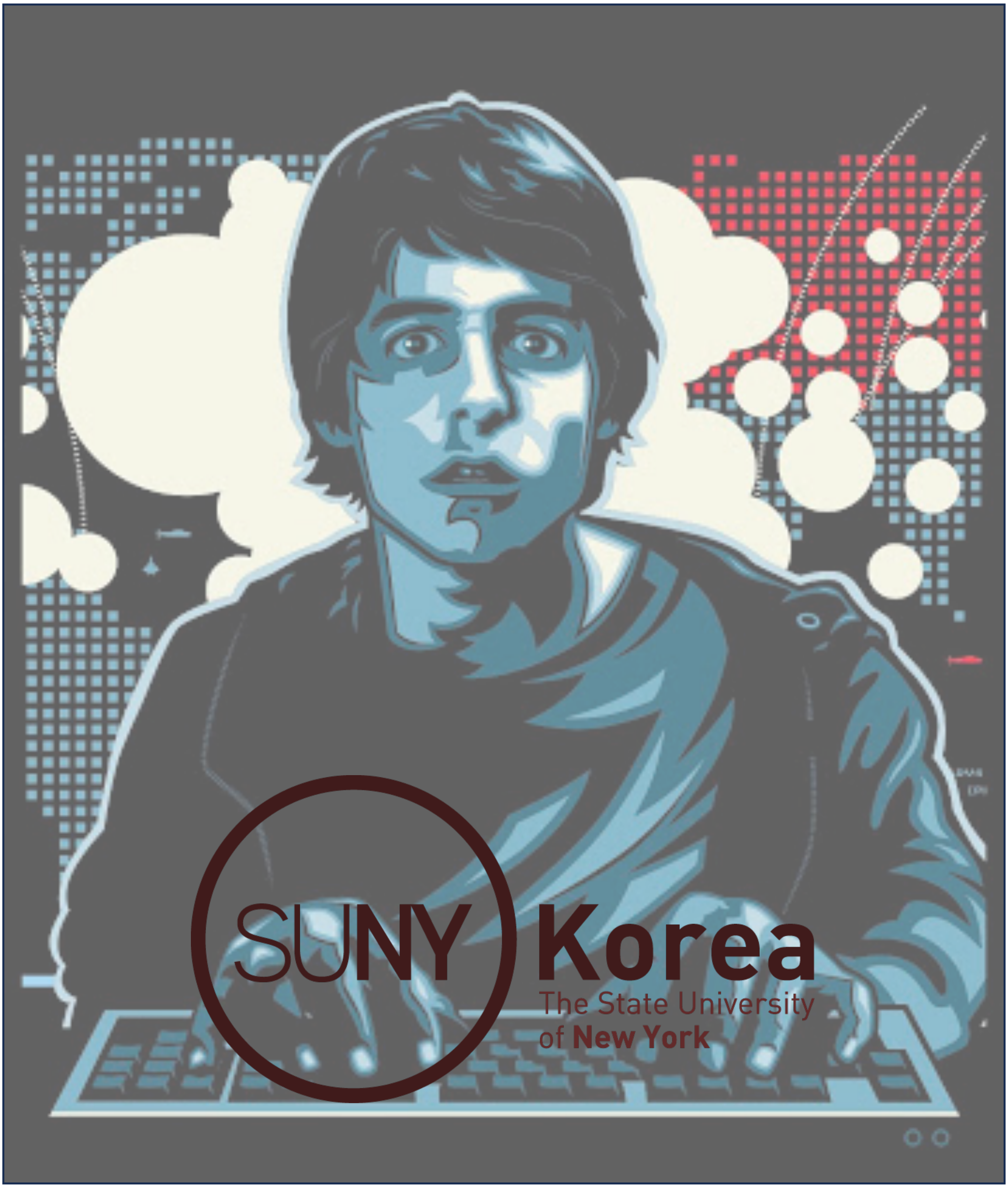
ASSIGNMENT II

PROFESSOR

FRANCOIS

RAMEAU

LIST & LOOP



General instructions

- Please add your name and email in the dedicated location at the top of your code
- Do not use any external library, except when explicitly requested
- Try to follow the naming convention proposed by PEP-8 seen in class
- Use meaningful names for your variables and functions
- If you face a problem submitting your code via GitHub, please contact the professor and the TA by email
- Note that the received code will be tested on a classifier to detect potential usage of Large Language Model. We will also pay particular attention to plagiarism
- Leave comments in your code to explain your code and describe the difficulties you faced

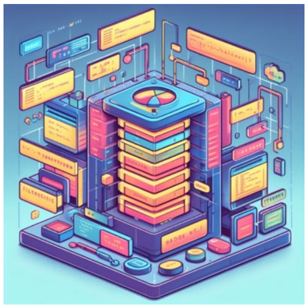
INVITATION LINK

https://classroom.github.com/a/6B2Xzw9G

Abstract

Today, you are a new recruit in a startup specialized in software for personal development. They are exploring a new product line for cellphone users, specifically a new ToDo list software. You have been selected as the team leader for the back-end development of this application.

Data Structure



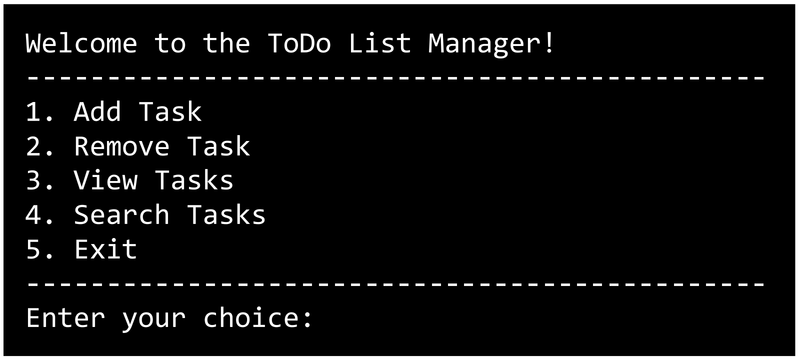
Use a list of lists (todo_list) to store tasks. Each inner list should contain the task description as its first element and its priority (Low, Medium, or High) as the second element.

Example: `[['Buy groceries', 'High'], ['Read a book', 'Medium']]`

This data structure will be defined in the main and modified in the functions add and remove tasks. It can be initialized as an empty list.

I- Create the Main Menu

Your very first task will be to develop the display of the main menu of your application. This functionality will be implemented in the function `display_menu().`



```
Welcome to the ToDo List Manager!
----------------------------------------------
1. Add Task
2. Remove Task
3. View Tasks
4. Search Tasks
5. Exit
----------------------------------------------
Enter your choice:
```

Here is what your function is expected to do:

1. Display the menu to the user
2. Prompt the user with `“Enter your choice: ”`
3. Read the user's input. Ensure that it is a valid choice (i.e., a number between 1 to 5 inclusive). If the input is invalid (either a number outside this range or a different type of input like a letter), display the message "Invalid choice. Please select a valid option." and prompt the user again until a valid choice is provided. You can check if the input is a number using the string method `isdigit()`.
4. Once a valid choice is made, the function should return this choice as an integer.

II- Add a Task

Complete the function `add_task(todo_list)`, here is what your function is expected to do:

1. Prompt the user with `“Enter the task description: ”`
2. Ask for its priority: `"Set task priority (Low, Medium, High): ".`
3. Validate the priority. If it's neither 'Low', 'Medium', nor 'High', show an error and ask again until a valid priority is provided.
4. Add the validated task and priority to `todo_list`.

III- View Tasks

Complete the function `view_tasks(todo_list)`, here is what your function is expected to do:

1. Show tasks (Bonus 0.5 points: show in order of priority: High, Medium, then Low.)
2. Display in the format: `"[index]. [task description] (Priority: [priority])"`

IV- Remove a Task

Complete the function `remove_task(todo_list)`, here is what your function is expected to do:

1. Display tasks (you can use the function you have created before)
2. Ask the user: `"Enter the index number of the task to remove: "`.
3. Validate the index. If it is not valid, show an error message and ask the user again
4. Remove the task from `todo_list`.

V- Search Tasks

The final feature we would like to implement for this project is the possibility to display all the tasks containing a certain keyword. In the function `search_tasks(todo_list)` you will:

1. Prompt: `"Enter keyword to search: "`.
2. Display tasks containing the keyword or show a message if none match.

Hint: You can use string methods such as `count` or `find` to achieve this task.

VI- Quit the Application

Keep in mind that whatever action the user takes, he should always be redirected to the main menu. So you should expect a while loop in the `main()` function of your code. When the user selects the option “5.” you have to quit this loop to exit the program.

Grading Scale (10 points + 0.5 bonus)

1. Displaying the Main Menu (2 points):

- *Displaying the menu to the user*: 1 point

- *Prompting the user correctly*: 0.5 points

- *Reading and validating the user input*: 0.5 points

2. Adding a Task (2 points):

- *Correctly prompting and adding task description*: 1 point

- *Correctly prompting for and validating the task priority*: 1 point

3. Viewing Tasks (1.5 points):

- *Correctly displaying all tasks with proper format*: 1 point

- *(Bonus) Displaying tasks in order of priority*: 0.5 points

4. Removing a Task (1.5 points):

- *Displaying tasks correctly*: 0.5 points

- *Correctly prompting and validating task index*: 0.5 points

- *Removing the task from the list*: 0.5 points

5. Searching Tasks (1.5 points):

- *Correctly prompting for a keyword*: 0.5 points

- Displaying tasks that contain the keyword: 0.5 points

- *Handling cases where no tasks match the keyword*: 0.5 points

6. Quitting the Application and Main Loop Handling (2 points):

- *Properly quits the application when "5" is selected*: 1 point

- *The user is always redirected back to the main menu after each action*: 1 point