ASSIGNMENT #5

Purpose

The purpose of this assignment is to practice building a User Interface and getting correct input from the user.

Instructions

In this lab you will finish our Todo-List application. When you run the program the application will launch with an empty Todo-List. There will be a menu that will offer the user with the following options:

- Create New Item
- Edit an Item
- Delete an Item
- View All Items
- View Specific Item
- Delete All Items
- Quit Program

For Extra Credit you can add the following:

- Save the list to a file
- Retrieve a previous saved list from a file

You will implement a driver (which loads the menus described above) and a UI for the Todo-List application. Your driver just need to create an instance of **TodoUI** and then call the **Menu()** function.

Class 1 - TodoUI

Contains the Menus and all other UI Operations. **All** input and output from this program must happen in the UI

Private Data Members: An instance of **`CinReader`** (or you can create your own input handler if you wish)
A dynamic instance of **`TodoList`**

Constructor: Allocates memory for the **`TodoList`** object

Destructor: Frees the memory associated with the dynamic **`TodoList`** object

Member Function 1: Named **`Menu`**. Contains all of the options specified above. All options may be contained in this function, but it would be better to make private member helper functions to help out the **`Menu`**

function.

Objectives

- Properly manage memory
- Use pointers and their related syntax
- Implement a User Interface

Requirements

Your code must follow the styling and documenting guidelines presented in class. Please note that I do not give points for style and documentation. You can only lose points. Please make sure your source code is documented correctly and is neatly and consistently formatted using guidelines provided in class.

Your program must provide the features described above:

[40 pts] - Fully complete **`TodoUI`**

[10 pts] - Fully complete driver

IF YOUR PROGRAM DOES NOT COMPILE YOU WILL RECEIVE A ZERO!!!

Deliverables (via Blackboard)

Your files need to be uploaded/attached to the Assignment Submission on Blackboard. You should upload the following seven files:

- `driver.cpp`
- `todo_item.h`
- `todo_item.cpp`
- `todo_list.h`
- `todo_list.cpp`
- `todo_ui.h`
- `todo_ui.cpp`

Extra Credit Options

In addition to the previously mentioned save to/retrieve from file option above you can add the following functionality to your application:

- Search for and view items using a keyword search on TodoItem text
- Search for and delete items using a keyword search on TodoItem text
- Search for and delete items that have been completed

ASSIGNMENT #5

- Add the following functions
  - Sort by description (case-insensitive)
  - Group by completion status