TODO List Menu Driven

Given three Python files:

1. $TodoMain.py$
2. $TodoLogic.py$
3. $ToolBox.py$

You have to write a program to build a menu-driven program for personal task reminder. The brief system architecture is shown in Figure 1.

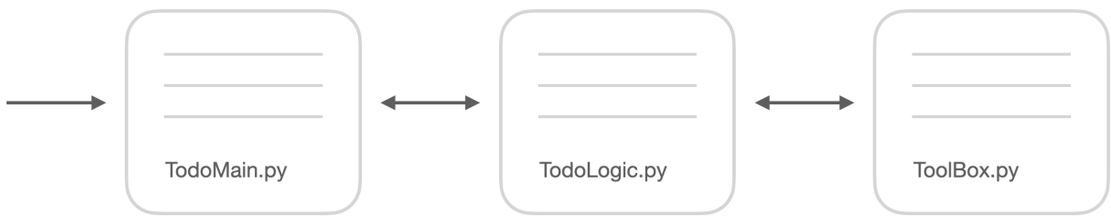

Figure 1: TODO List system architecture

The $TodoMain.py$ is a main program where the user can interact with. The $TodoLogic.py$ provides the console interface as a menu-driven program. You can see the example of starting console interface below. **The main program must not contain any functionality logics, but you must import and use the functionality logics where you defined in $TodoLogic.py$.**

1. **(5 points) Main Program**

```
TODO List Menu Driven
PERSONAL TASK REMINDER
---- ---- ---- ---- ---- ---- ----
1. Create a new task
2. Mark completes for a specific task
3. Show incomplete tasks sort by priority
4. Show incomplete tasks for a specific priority
5. Show summary report
6. Remove a specific task
7. Mark completes for every tasks you have

Enter option (1-7, 0 to exit): _
```


2. (2 points) Create a new task

This method takes task information from the user: name and priority. There are 3 priority levels: 1 Low, 2 Medium, and 3 High. The new task will then be added to the list altogether with one more information which is status. The status will be assigned as "I" (Incomplete). The program also display a current time which you can get value from function "getTime()", in $ToolBox.py$. The example is shown below.

```
... program output message continues ...

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Programming Homework
Enter priority level: 3
New task added at 8/10/21 13:30:55

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Essay Writing
Enter priority level: 2
New task added at 8/10/21 13:31:01

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Watch Netflix Series
Enter priority level: 1
New task added at 8/10/21 13:31:06

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Typing Practice
Enter priority level: 3
New task added at 8/10/21 13:31:11

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Play Piano
Enter priority level: 2
New task added at 8/10/21 13:31:12

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Listen to English news
Enter priority level: 2
New task added at 8/10/21 13:31:18

Enter option (1-7, 0 to exit): 1
---- ---- ---- ---- ---- ---- ----
Enter task name: Shopping with friends
Enter priority level: 1
New task added at 8/10/21 13:31:25

Enter option (1-7, 0 to exit): _

... program output message continues ...
```


3. **(2 points) Mark completes for a specific task**

   This method takes task index and update the task status. The status will be changed to "C" (Complete). The example is shown below and assume that there are 7 tasks in the list.

```
... program output message continues ...

Enter option (1-7, 0 to exit): 2
---- ---- ---- ---- ---- ---- ----
Enter task index: 0
Programming Homework completed at 8/10/21 13:32:12

Enter option (1-7, 0 to exit): 2
---- ---- ---- ---- ---- ---- ----
Enter task index: 9
Task not found!

Enter option (1-7, 0 to exit): _

... program output message continues ...
```

4. **(3 points) Show incomplete tasks sort by priority**

   This method takes nothing and print out list of incomplete tasks which sort by priority level from High to Low. The incomplete tasks can be indicated with the status equals to "I" (Incomplete). The priority name can be derived from function "priorityName(level)" in $ToolBox.py$. The example is shown below.

```
... program output message continues ...

Enter option (1-7, 0 to exit): 3
---- ---- ---- ---- ---- ---- ----
Priority   Task(s)
High       Typing Practice
Medium     Essay Writing
Medium     Play Piano
Medium     Listen to English news
Low        Watch Netflix Series
Low        Shopping with friends

Enter option (1-7, 0 to exit): _

... program output message continues ...
```


5. **(2 points) Show incomplete tasks for a specific priority**

   This method takes priority level as a number and print out list of incomplete tasks with given priority level. The incomplete tasks can be indicated with the status equals to "I" (Incomplete). The priority name can be derived from function "priorityName(level)" in $ToolBox.py$. The example is shown below.

```
... program output message continues ...

Enter option (1-7, 0 to exit): 4
---- ---- ---- ---- ---- ---- ----
Enter priority level: 1
Priority   Task(s)
Low        Watch Netflix Series
Low        Shopping with friends

Enter option (1-7, 0 to exit): _

... program output message continues ...
```

6. **(2 points) Show summary report**

   This method takes nothing. This method displays the summary information. The example is shown below.

```
... program output message continues ...

Enter option (1-7, 0 to exit): 5
---- ---- ---- ---- ---- ---- ----
Summary Report
No. of incomplete tasks is 6 (High=1, Medium=3, Low=2)
No. of complete tasks is 1 (High=1, Medium=0, Low=0)

Enter option (1-7, 0 to exit): _

... program output message continues ...
```

7. **(2 points) Remove a specific task**

   This method takes task index and remove that task from the list. If the specified task index is not in the correct range, the program displays "Task not found!". The example is shown below.

```
... program output message continues ...

Enter option (1-, 0 to exit): 6
---- ---- ---- ---- ---- ---- ----
Enter task index: 3
Typing Practice removed!

Enter option (1-7, 0 to exit): _

... program output message continues ...
```


8. (2 points) Mark completes for every tasks you have

This method takes nothing. This method updates task status to "C" (Complete) on all the tasks you have (only incomplete tasks will be updated). Like the second menu option. The example is shown below.

```
... program output message continues ...

Enter option (1-7, 0 to exit): 7
---- ---- ---- ---- ---- ---- ----
Essay Writing completed at 8/10/21 13:41:21
Watch Netflix Series completed at 8/10/21 13:41:21
Play Piano completed at 8/10/21 13:41:21
Listen to English news completed at 8/10/21 13:41:21
Shopping with friends completed at 8/10/21 13:41:22

Enter option (1-7, 0 to exit): 7
---- ---- ---- ---- ---- ---- ----
No incomplete tasks

Enter option (1-7, 0 to exit): _

... program output message continues ...
```

Please note that there might be some cases for handling an incorrect input which are not defined in the example. You have to think of those cases and handle them properly.

End of the examination.

