

---

**Cut-Off Date:** 25-29 April, 2024

**Total Marks:** 60 marks turned to 15 marks

## **Contents**

Warnings and Declaration.....	1
Marking Criteria .....	2
Question 1 .....	3

## **Plagiarism Warning:**

As per AOU rules and regulations, all students are required to submit their own TMA work and avoid plagiarism. The AOU has implemented sophisticated techniques for plagiarism detection. You must provide all references in case you use and quote another person's work in your TMA. You will be penalized for any act of plagiarism as per the AOU's rules and regulations.

## **Declaration of No Plagiarism by Student (to be signed and submitted by student with TMA work):**

I hereby declare that this submitted TMA work is a result of my own efforts and I have not plagiarized any other person's work. I have provided all references of information that I have used and quoted in my TMA work.

Name of Student:.....

Signature:.....

Date:.....

**Apply the following procedure for solving the TMA problem.**

- A. Read and analyze the problem statement.
- B. Formulate the algorithm using pseudocode and assume any required missed data reasonably.
- C. Write a C# application.
- D. Write a comment for each line of code using your own words.**
- E. Test, debug and execute the C# application.
- F. Copy and paste your code from Visual Studio to the Answer Booklet (**Don't put your code as a screenshot**).
- G. Provide Screenshots for both the input and output of **ALL** cases of each problem. **DO NOT submit sample of screenshots.**
- H. Your TMA Answer Booklet should be structured as follows:

**Problem: To-Do List Application:**

1.1 Algorithm pseudocode

1.2 Complete problem's application code (**Copied**)

1.3 Screenshots of both input and output for **ALL** cases of the problem

Distribution of Marks	
Requirement	Mark
Algorithm pseudocode	20%
Testing	30%
Valid report format	5%
Comments	15%
Running application	30%

## To-Do List Application (60 marks)

Create a console application that simulates a To-Do List. The application should allow users to manage their tasks by performing various operations such as adding tasks, listing tasks, editing tasks, marking tasks as completed, deleting tasks, and setting task priorities.

### Requirements:

- 1- When the application starts, display a welcome message to the user.
- 2- Implement a menu system that allows the user to select different options.
- 3- The following options should be available:
  - 3.1 "**add**": Allow the user to add a new task to the list, including a title, description, and due date.
  - 3.2 "**list**": Display all the tasks in the list, including their titles, due dates, priorities, and completion status. The tasks should be sorted by priority, with high priority tasks displayed first, followed by medium and low priority tasks.
  - 3.3 "**edit**": Allow the user to edit an existing task by specifying the task ID. They should be able to modify the title, description, due date, and priority of the task.
  - 3.4 "**complete**": Allow the user to mark a task as completed by specifying the task ID.
  - 3.5 "**delete**": Allow the user to delete a task from the list by specifying the task ID.
  - 3.6 "**set priority**": Allow the user to set the priority of a task by specifying the task ID and the new priority (high, medium, or low).
  - 3.7 "**exit**": Quit the application.
- 4- Implement validation for user inputs to ensure correct data entry. For example, validate that due dates are in the correct format (YYYY-MM-DD), task IDs exist in the list before performing edit, completion, deletion, or priority setting operations, and prioritize tasks with valid priority levels (high, medium, or low).
- 5- Provide appropriate feedback messages to the user after each operation. Display an error message if the user enters an invalid option.
- 6- Store the tasks in memory during the runtime of the application. Do not implement any permanent data storage.
- 7- Ensure that the code is clean, well-structured, and easy to understand. Use proper naming conventions and meaningful comments where necessary.

## Sample of Screens

### Main Screen

```
Welcome to the To-Do List Application!
```

```
Please select an option:
```

1. Add a task
2. List all tasks
3. Edit a task
4. Mark a task as completed
5. Delete a task
6. Set task priority
7. Exit

```
Enter your choice: 1
```

### Add Task

```
Enter task details:
```

```
Title: Finish report
```

```
Description: Complete the financial report for Q3
```

```
Due date (YYYY-MM-DD): 2022-03-15
```

```
Priority (high, medium, low): high
```

```
Task added successfully!
```

### Edit Task

```
Enter the task ID to edit: 1
```

```
Enter new task details:
```

```
Title: Complete TMA report
```

```
Description: Finish the TMA report for M109 Course
```

```
Due date (YYYY-MM-DD): 2024-03-20
```

```
Priority (high, medium, low): High
```

```
Task updated successfully!
```

### List All Tasks

ID	Title	Due Date	Priority	Completed
-----				
1	Finish report	2022-03-15	High	No
2	Study	2022-02-20	Medium	Yes
3	Prepare my list	2022-02-14	Low	No

### Delete Task

```
Enter the task ID to delete: 2  
Task with ID 2 deleted successfully!
```

### Change Task's Priority

```
Enter the task ID to set priority: 1  
Enter the new priority (high, medium, low): medium  
Priority of task with ID 1 set to medium successfully!
```

### Exit Application

```
Thank you for using the To-Do List Application. Goodbye!
```

*Good Luck*