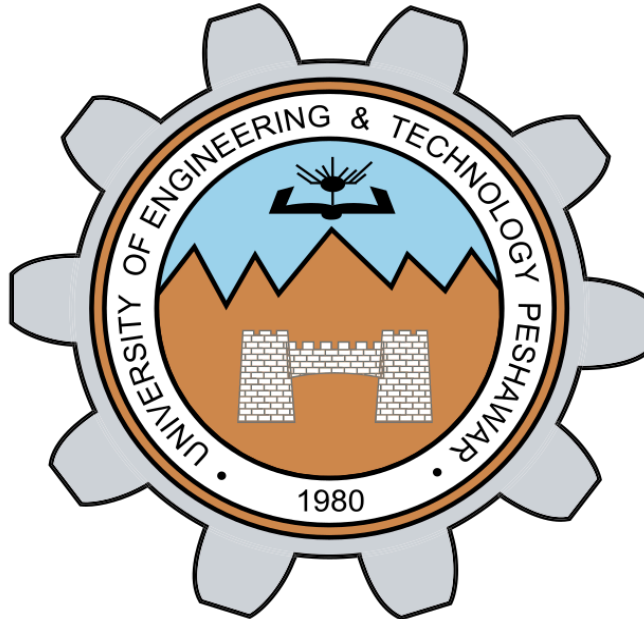


# UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR



**SUBMITTED BY:** Waqas Khan

**SUBMITTED TO:** SIR ADEEL ALI SHAH

**SECTION:** "A"

**REG NO #** 24PWBCS1141

**DATE:** 9/January/2024

**TASK NO:** 2

**DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY PESHAWAR**

Lab\_08\_OOPSVersion control

Current File

Task.javaTaskManagementSystem.java

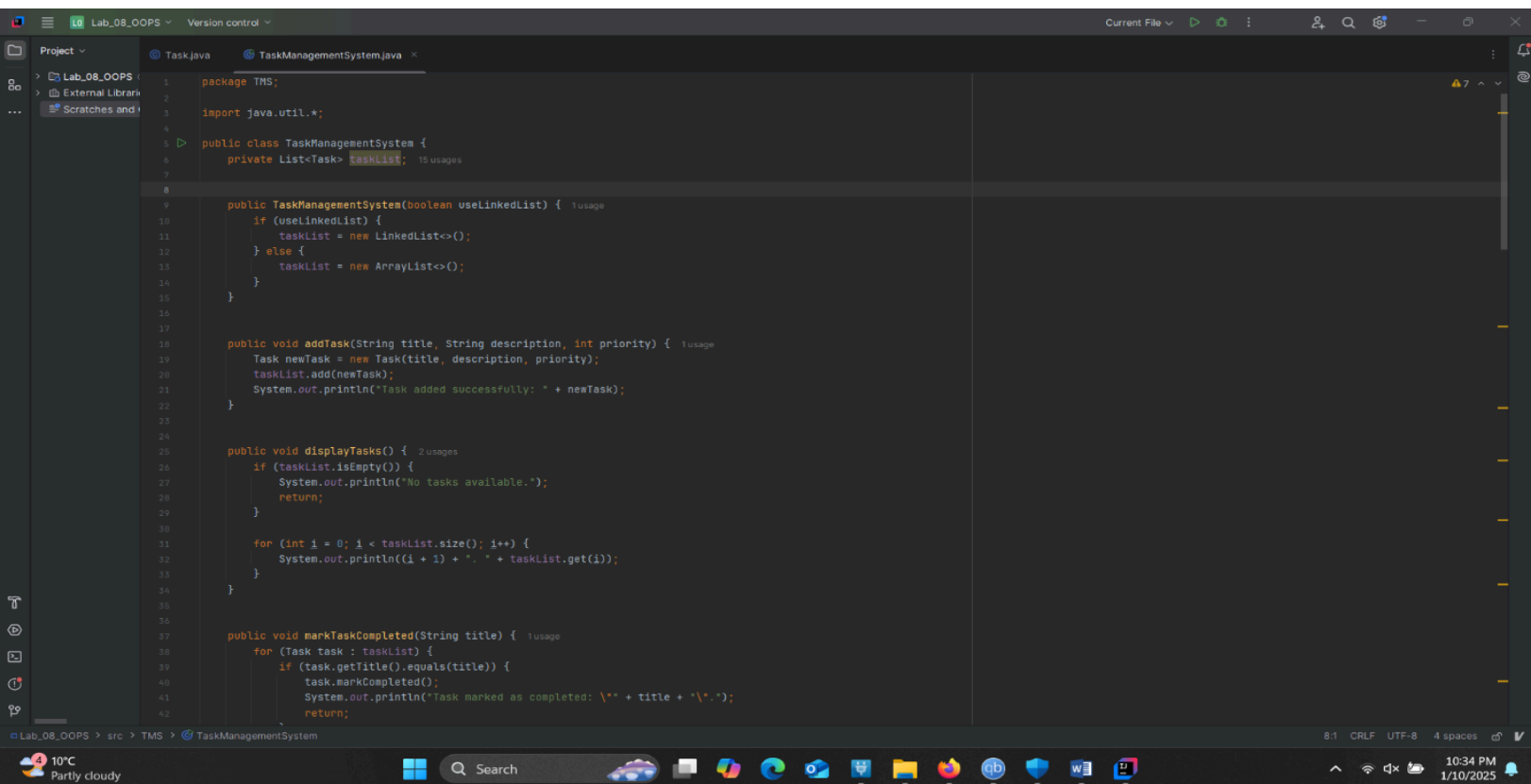
Lab\_08\_OOPSExternal LibrariesScratches and Snippets

```
1 package TMS;
2
3 public class Task {
4     private String title;
5     private String description;
6     private boolean isCompleted;
7     private int priority;
8
9
10    public Task(String title, String description, int priority) {
11        this.title = title;
12        this.description = description;
13        this.isCompleted = false;
14        this.priority = priority;
15    }
16
17
18    public String getTitle() {
19        return title;
20    }
21
22    public boolean isCompleted() {
23        return isCompleted;
24    }
25
26    public void markCompleted() {
27        this.isCompleted = true;
28    }
29
30    public int getPriority() {
31        return priority;
32    }
33
34
35    @Override
36    public String toString() {
37        return "Title: " + title + ", Description: " + description + ", Priority: " + priority + ", Completed: " + isCompleted;
38    }
39 }
40
```

3

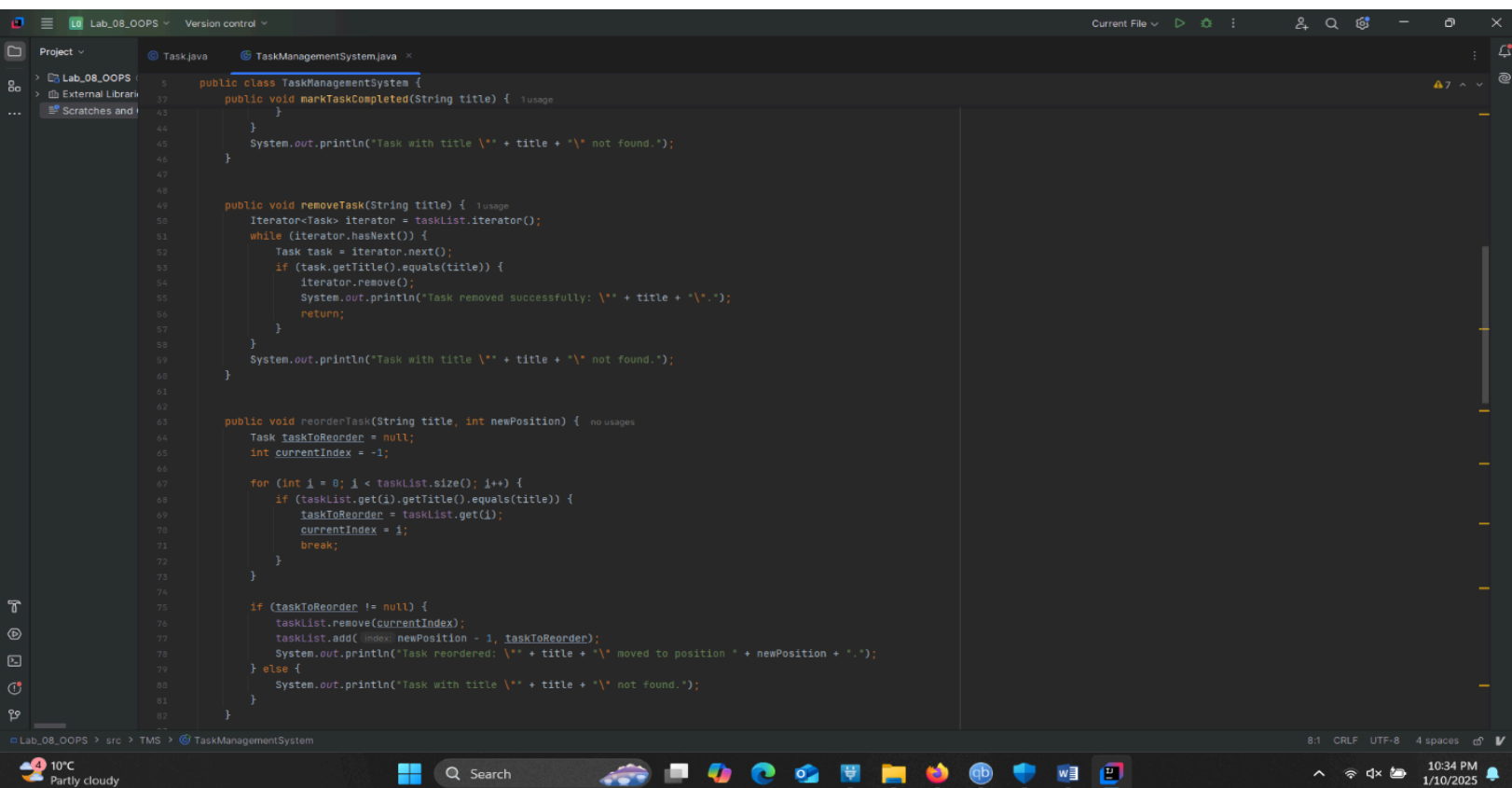
Lab\_08\_OOPS > src > TMS > Task > isCompleted

24:6 CRLF UTF-8 4 spaces



This screenshot shows the TaskManagementSystem.java file in an IDE. The code defines a class with a private List of Task objects. It includes three methods: addTask, displayTasks, and markTaskCompleted. The addTask method creates a new Task object and adds it to the list. The displayTasks method iterates through the list and prints each task's details. The markTaskCompleted method iterates through the list and marks a task as completed if its title matches the provided title.

```
1 package TMS;
2
3 import java.util.*;
4
5 public class TaskManagementSystem {
6     private List<Task> taskList; // 15 usages
7
8
9     public TaskManagementSystem(boolean useLinkedList) { // usage
10         if (useLinkedList) {
11             taskList = new LinkedList<>();
12         } else {
13             taskList = new ArrayList<>();
14         }
15     }
16
17
18     public void addTask(String title, String description, int priority) { // usage
19         Task newTask = new Task(title, description, priority);
20         taskList.add(newTask);
21         System.out.println("Task added successfully: " + newTask);
22     }
23
24
25     public void displayTasks() { // 2 usages
26         if (taskList.isEmpty()) {
27             System.out.println("No tasks available.");
28             return;
29         }
30
31         for (int i = 0; i < taskList.size(); i++) {
32             System.out.println((i + 1) + ". " + taskList.get(i));
33         }
34     }
35
36
37     public void markTaskCompleted(String title) { // usage
38         for (Task task : taskList) {
39             if (task.getTitle().equals(title)) {
40                 task.markCompleted();
41                 System.out.println("Task marked as completed: \"" + title + "\".");
42                 return;
43             }
44         }
45     }
46 }
```



This screenshot shows the TaskManagementSystem.java file in an IDE, continuing from the previous one. It adds two more methods: removeTask and reorderTask. The removeTask method iterates through the list and removes a task if its title matches the provided title. The reorderTask method iterates through the list and reorders a task to a new position if its title matches the provided title.

```
37     public void markTaskCompleted(String title) { // usage
38     }
39
40     System.out.println("Task with title \"" + title + "\" not found.");
41 }
42
43
44     public void removeTask(String title) { // usage
45         Iterator<Task> iterator = taskList.iterator();
46         while (iterator.hasNext()) {
47             Task task = iterator.next();
48             if (task.getTitle().equals(title)) {
49                 iterator.remove();
50                 System.out.println("Task removed successfully: \"" + title + "\".");
51                 return;
52             }
53         }
54         System.out.println("Task with title \"" + title + "\" not found.");
55     }
56
57
58     public void reorderTask(String title, int newPosition) { // no usages
59         Task taskToReorder = null;
60         int currentIndex = -1;
61
62         for (int i = 0; i < taskList.size(); i++) {
63             if (taskList.get(i).getTitle().equals(title)) {
64                 taskToReorder = taskList.get(i);
65                 currentIndex = i;
66                 break;
67             }
68         }
69
70         if (taskToReorder != null) {
71             taskList.remove(currentIndex);
72             taskList.add((index: newPosition - 1, taskToReorder);
73             System.out.println("Task reordered: \"" + title + "\" moved to position " + newPosition + ".");
74         } else {
75             System.out.println("Task with title \"" + title + "\" not found.");
76         }
77     }
78 }
```

```
Lab_08_OOPS - Version control - Current File - 7 - 10:35 PM 1/10/2025

Project - Task.java - TaskManagementSystem.java x
> Lab_08_OOPS
> External Libraries
Scratches and Snippets

5 public class TaskManagementSystem {
6
7     public void filterTasksByCompletion(boolean showCompleted) { no usages
8         boolean found = false;
9         for (Task task : taskList) {
10             if (task.isCompleted() == showCompleted) {
11                 System.out.println(task);
12                 found = true;
13             }
14         }
15         if (!found) {
16             System.out.println("No tasks found with the specified completion status.");
17         }
18     }
19
20     public void sortTasksByPriority() { no usages
21         taskList.sort((Task t1, Task t2) -> Integer.compare(t2.getPriority(), t1.getPriority()));
22         System.out.println("Tasks sorted by priority.");
23         displayTasks();
24     }
25
26     public static void main(String[] args) {
27         Scanner scanner = new Scanner(System.in);
28         System.out.println("Choose data structure for storing tasks:");
29         System.out.println("1. LinkedList");
30         System.out.println("2. ArrayList");
31         int choice = scanner.nextInt();
32         scanner.nextLine(); // Consume newline left by nextInt
33
34         TaskManagementSystem system = new TaskManagementSystem((useLinkedList: choice == 1);
35
36         while (true) {
37             System.out.println("\nTask Management Menu:");
38             System.out.println("1. Add Task");
39             System.out.println("2. Display All Tasks");
40             System.out.println("3. Mark Task as Completed");
41             System.out.println("4. Remove Task");
42             System.out.println("5. Reorder Task");
43             System.out.println("6. Filter tasks by Completion");
44             System.out.println("7. Sort tasks by Priority");
45             System.out.println("8. Exit");
46
47             int option = scanner.nextInt();
48             scanner.nextLine();
49
50             switch (option) {
51                 case 1:
52                     System.out.print("Enter task title: ");
53                     String title = scanner.nextLine();
54                     System.out.print("Enter task description: ");
55                     String description = scanner.nextLine();
56                     System.out.print("Enter task priority (1 for highest): ");
57                     int priority = scanner.nextInt();
58                     system.addTask(title, description, priority);
59                     break;
60
61                 case 2:
62                     system.displayTasks();
63                     break;
64
65                 case 3:
66                     System.out.print("Enter task title to mark as completed: ");
67                     title = scanner.nextLine();
68                     system.markTaskCompleted(title);
69                     break;
70
71                 case 4:
72                     System.out.print("Enter task title to remove: ");
73                     title = scanner.nextLine();
74                     system.removeTask(title);
75                     break;
76
77                 case 5:
78                     System.out.print("Enter task title to reorder: ");
79                     title = scanner.nextLine();
80                     break;
81
82                 default:
83                     System.out.println("Invalid option. Please try again.");
84                     break;
85             }
86         }
87     }
88 }
```

```
Lab_08_OOPS - Version control - Current File - 7 - 10:35 PM 1/10/2025

Project - Task.java - TaskManagementSystem.java x
> Lab_08_OOPS
> External Libraries
Scratches and Snippets

5 public class TaskManagementSystem {
6
7     public static void main(String[] args) {
8
9         Scanner scanner = new Scanner(System.in);
10
11         int option = scanner.nextInt();
12         scanner.nextLine();
13
14         switch (option) {
15             case 1:
16                 System.out.print("Enter task title: ");
17                 String title = scanner.nextLine();
18                 System.out.print("Enter task description: ");
19                 String description = scanner.nextLine();
20                 System.out.print("Enter task priority (1 for highest): ");
21                 int priority = scanner.nextInt();
22                 system.addTask(title, description, priority);
23                 break;
24
25             case 2:
26                 system.displayTasks();
27                 break;
28
29             case 3:
30                 System.out.print("Enter task title to mark as completed: ");
31                 title = scanner.nextLine();
32                 system.markTaskCompleted(title);
33                 break;
34
35             case 4:
36                 System.out.print("Enter task title to remove: ");
37                 title = scanner.nextLine();
38                 system.removeTask(title);
39                 break;
40
41             case 5:
42                 System.out.print("Enter task title to reorder: ");
43                 title = scanner.nextLine();
44                 break;
45
46             default:
47                 System.out.println("Invalid option. Please try again.");
48                 break;
49         }
50     }
51 }
```

```
Run TaskManagementSystem x
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=58451:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\bin" -Dfile.encoding=UTF-8
Choose data structure for storing tasks:
1. LinkedList
2. ArrayList
3
Task Management Menu:
1. Add Task
2. Display All Tasks
3. Mark Task as Completed
4. Remove Task
5. Reorder Task
6. Filter Tasks by Completion
7. Sort Tasks by Priority
8. Exit
9
Enter task title: Task1
Enter task description: No
Enter task priority (1 for highest): 1
Task added successfully: Title: "Task1", Description: "No", Priority: 1, Completed: false

Task Management Menu:
1. Add Task
2. Display All Tasks
3. Mark Task as Completed
4. Remove Task
5. Reorder Task
6. Filter Tasks by Completion
7. Sort Tasks by Priority
8. Exit
9
1. Title: "Task1", Description: "No", Priority: 1, Completed: false

Task Management Menu:
1. Add Task
2. Display All Tasks
3. Mark Task as Completed
4. Remove Task
5. Reorder Task
6. Filter Tasks by Completion
7. Sort Tasks by Priority
8. Exit
9
Enter task title to mark as completed: 1
```

