

Data Structures and Algorithms

Exercise 5:

Task Management System

This project implements a Task Management System using Java and a singly linked list. It supports efficient task addition, deletion, searching, and traversal while demonstrating dynamic memory handling. Here’s a detailed explanation:

Step 1: Understand Linked Lists

- **Singly Linked List:** Each node points to the next.
 - **Doubly Linked List:** Nodes point both forward and backward (not used here).
-

Step 2: Setup

- *Task* class with *taskId*, *taskName*, and *status*.
-

Step 3: Implementation

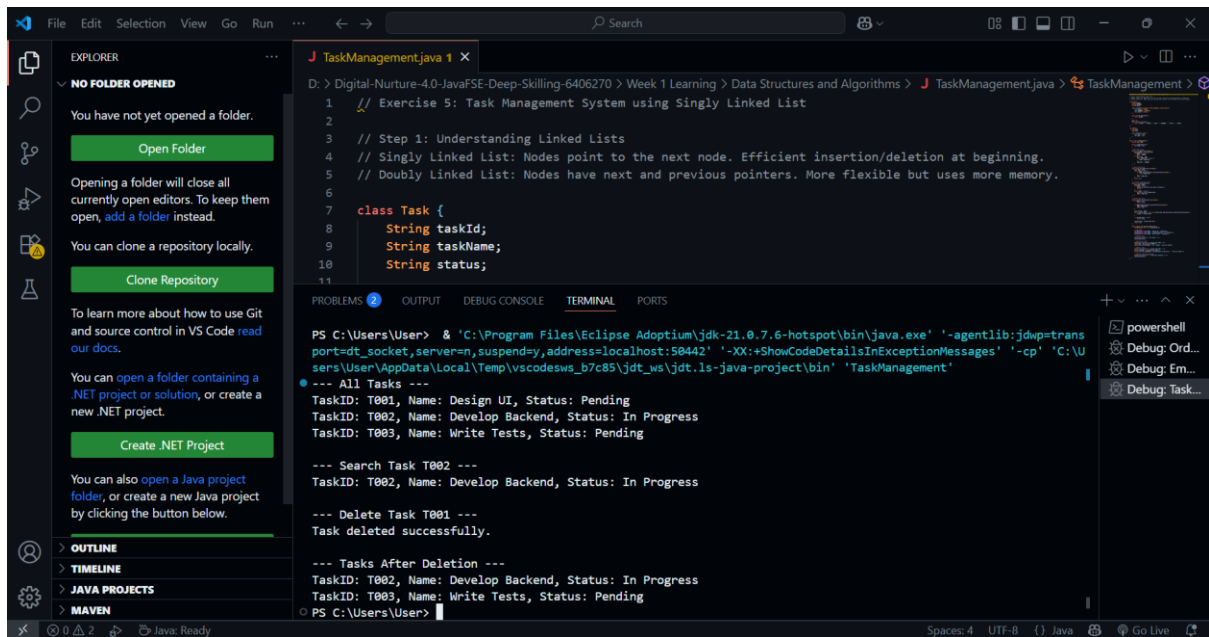
- Node structure created for Linked List.
 - Methods implemented:
 - *addTask()* – at end
 - *searchTask()* – by ID
 - *displayTasks()* – traversal
 - *deleteTask()* – by ID
-

Step 4: Time Complexity Analysis

Operation	Time Complexity
Add	O(n)
Search	O(n)
Traverse	O(n)
Delete	O(n)

- **Linked lists** are ideal for dynamic datasets where frequent insertions/deletions occur without memory reallocation overhead.

Output



The screenshot displays the Visual Studio Code interface. The Explorer panel on the left shows a project structure with folders for 'OUTLINE', 'TIMELINE', 'JAVA PROJECTS', and 'MAVEN'. The main editor window shows a Java file named 'TaskManagement.java' with the following content:

```
1 // Exercise 5: Task Management System using Singly Linked List
2
3 // Step 1: Understanding Linked Lists
4 // Singly Linked List: Nodes point to the next node. Efficient insertion/deletion at beginning.
5 // Doubly Linked List: Nodes have next and previous pointers. More flexible but uses more memory.
6
7 class Task {
8     String taskId;
9     String taskName;
10    String status;
11}
```

The bottom panel shows the 'TERMINAL' output, which displays the execution of the program. The command executed is:

```
PS C:\Users\User> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.7.6-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:50442' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Local\Temp\vscodesws_b7c85\jdt_ws\jdt.ls-java-project\bin' 'TaskManagement'
```

The output shows the following sequence of operations:

- All Tasks ---
- TaskID: T001, Name: Design UI, Status: Pending
- TaskID: T002, Name: Develop Backend, Status: In Progress
- TaskID: T003, Name: Write Tests, Status: Pending
- Search Task T002 ---
- TaskID: T002, Name: Develop Backend, Status: In Progress
- Delete Task T001 ---
- Task deleted successfully.
- Tasks After Deletion ---
- TaskID: T002, Name: Develop Backend, Status: In Progress
- TaskID: T003, Name: Write Tests, Status: Pending

The terminal output is displayed in a PowerShell window. The status bar at the bottom indicates 'Spaces: 4', 'UTF-8', 'Java', and 'Go Live'.