



**COMSATS UNIVERSITY ISLAMABAD**  
**ATTOCK CAMPUS**

**BS(Software Engineering)**

|                |                     |
|----------------|---------------------|
| NAME:          | Muhammad Haroon     |
| REG. NO:       | SP23-BSE-042        |
| SUBJECT:       | DS                  |
| ASSIGNMENT NO: | 01                  |
| DATE:          | 09-24-2024          |
| SUBMITTED TO:  | Mr. Muhammad Kamran |

## Introduction

The objective of this assignment is to design and implement a task management system using linked lists in C++. This system allows the user to manage tasks, which include adding tasks with different priorities, viewing all tasks, removing the highest priority task, and removing a task by its ID. The system ensures that tasks with higher priority are handled first, providing a flexible yet structured way to manage tasks efficiently.

## Code Explanation

### Task Struct:

The `Task` struct stores information for each task. It contains:

- `taskId`: The unique identifier for each task.
- `description`: A brief description of the task.
- `priority`: A number indicating the task's importance (higher numbers indicate higher priority).
- `next`: A pointer to the next task in the linked list.

### TaskList Class:

This class manages the operations related to the task list. The linked list is implemented using dynamic memory allocation.

### addTask:

This function inserts a new task into the list. The task is inserted based on its priority, ensuring tasks with higher priority appear at the front of the list. If the list is empty or the new task has the highest priority, it becomes the new head. Otherwise, it finds the correct position for insertion.

### removeHighestPriorityTask:

This function removes the task with the highest priority, which is always located at the head of the list. It adjusts the head to point to the next task and deallocates the memory used by the removed task.

**removeTaskById:**

This function removes a task by its ID. If the task to be removed is the head, it adjusts the head. Otherwise, it searches for the task in the list and adjusts the pointers to bypass the removed task, then frees the allocated memory.

**viewAllTasks:**

This function traverses the list and prints all tasks, displaying the task ID, description, and priority.

**3. Main Function:**

The main function presents a menu to the user to perform various operations:

- Option 1 allows the user to add a new task.
- Option 2 displays all tasks currently in the list.
- Option 3 removes the task with the highest priority.
- Option 4 removes a task based on the task ID.
- Option 5 exits the program.

**Output Screenshots**

STUD: X how: X Inbox: X cities: X SSL/I: X Get st: X (9) Wi: X GitHub: X New: X Online: X

programiz.com/cpp-programming/online-compiler/

Compress JPEG Ima... WebP to JPG image... Home - Merch Infor... WEBP Converter | CL... untitled report Free Online Plagiar... Add New Post + Tec... All Bookmarks

Programiz C++ Online Compiler

Markets Gold Stocks Oil Indices Crypto Trade Up Trustpilot

main.cpp

```
107:      getline(cin, details);
108:      cout << "Enter job rank: ";
109:      cin >> rank;
110:      joblist.InsertJob(id, details, rank);
111:      break;
112:  }
113:  case 2:
114:      joblist.displayJobs();
115:      break;
116:  case 3:
117:      joblist.removeTopPriorityJob();
118:      break;
119:  case 4: {
120:      int id;
121:      cout << "Enter the job ID to delete: ";
122:      cin >> id;
123:      joblist.deleteJobById(id);
124:      break;
125:  }
126:  case 5:
127:      exit(0);
128:  default:
129:      cout << "Invalid option. Try again." << endl;
130:  }
131: }
132:
133: return 0;
```

Output

```
/tmp/zJdyayoggs.o
1. Add a new job
2. Display all jobs
3. Remove the job with the highest rank
4. Delete a job by ID
5. Quit
Choose an option: 1
Enter job ID: |
```

Breaking news  
Brett Favre tells...

Search

11:33:23 PM  
9/24/2024

STUD: X how: X Inbox: X cities: X SSL/I: X Get st: X (9) Wi: X GitHub: X New: X Online: X

programiz.com/cpp-programming/online-compiler/

Compress JPEG Ima... WebP to JPG image... Home - Merch Infor... WEBP Converter | CL... untitled report Free Online Plagiar... Add New Post + Tec... All Bookmarks

Programiz C++ Online Compiler

Markets Access 60+ CFD Forex Pairs with Ultra Fast Execution Trade Now

main.cpp

```
107:      getline(cin, details);
108:      cout << "Enter job rank: ";
109:      cin >> rank;
110:      joblist.InsertJob(id, details, rank);
111:      break;
112:  }
113:  case 2:
114:      joblist.displayJobs();
115:      break;
116:  case 3:
117:      joblist.removeTopPriorityJob();
118:      break;
119:  case 4: {
120:      int id;
121:      cout << "Enter the job ID to delete: ";
122:      cin >> id;
123:      joblist.deleteJobById(id);
124:      break;
125:  }
126:  case 5:
127:      exit(0);
128:  default:
129:      cout << "Invalid option. Try again." << endl;
130:  }
131: }
132:
133: return 0;
```

Output

```
/tmp/1uapn0xet.o
1. Add a new job
2. Display all jobs
3. Remove the job with the highest rank
4. Delete a job by ID
5. Quit
Choose an option: 1
Enter job ID: 1234
Enter job details: ABD
Enter job rank: 1
1. Add a new job
2. Display all jobs
3. Remove the job with the highest rank
4. Delete a job by ID
5. Quit
Choose an option: 2
Job ID: 1234
Details: ABD
Rank: 1
1. Add a new job
2. Display all jobs
3. Remove the job with the highest rank
4. Delete a job by ID
5. Quit
Choose an option: |
```

S&P 500  
+0.14%

Search

11:40:13 PM  
9/24/2024

The screenshot shows the Programiz C++ Online Compiler interface. The code editor on the left contains a C++ program for a linked list. The output window on the right shows the program's execution results.

```
main.cpp
107
108     getline(cin, details);
109     cout << "Enter job rank: ";
110     cin >> rank;
111     joblist.InsertJob(id, details, rank);
112     break;
113 }
114 case 2:
115     joblist.displayJobs();
116     break;
117 case 3:
118     joblist.removeTopPriorityJob();
119     break;
120 case 4: {
121     int id;
122     cout << "Enter the job ID to delete: ";
123     cin >> id;
124     joblist.deleteJobById(id);
125     break;
126 }
127 case 5:
128     exit(0);
129 default:
130     cout << "Invalid option. Try again." << endl;
131 }
132
133 return 0;
```

Output

```
/tmp/1aynOpXet.o
1. Add a new job
2. Display all jobs
3. Remove the job with the highest rank
4. Delete a job by ID
5. Quit
Choose an option: 1
Enter job ID: 1234
Enter job details: ABD
Enter job rank: 1
1. Add a new job
2. Display all jobs
3. Remove the job with the highest rank
4. Delete a job by ID
5. Quit
Choose an option: |
```

## Conclusion

Through this assignment, I learned how to manage dynamic memory in C++ and implement a linked list to store and organize data efficiently. The biggest challenge was handling edge cases, such as inserting at the head or removing the last task in the list. This experience has improved my understanding of data structures and how to apply them in practical scenarios.