

Lab Report 7

1.

The objectives and concepts explored through this assignment were comparing different implementations of ordered lists. Concepts such as efficiency, performance of ordered lists, templates, classes, derived classes were covered. These fundamental concepts in data structures and algorithm are crucial for computer science and engineering students. These concepts increase readability and reusability of code which is very important when working as software engineer in the industry.

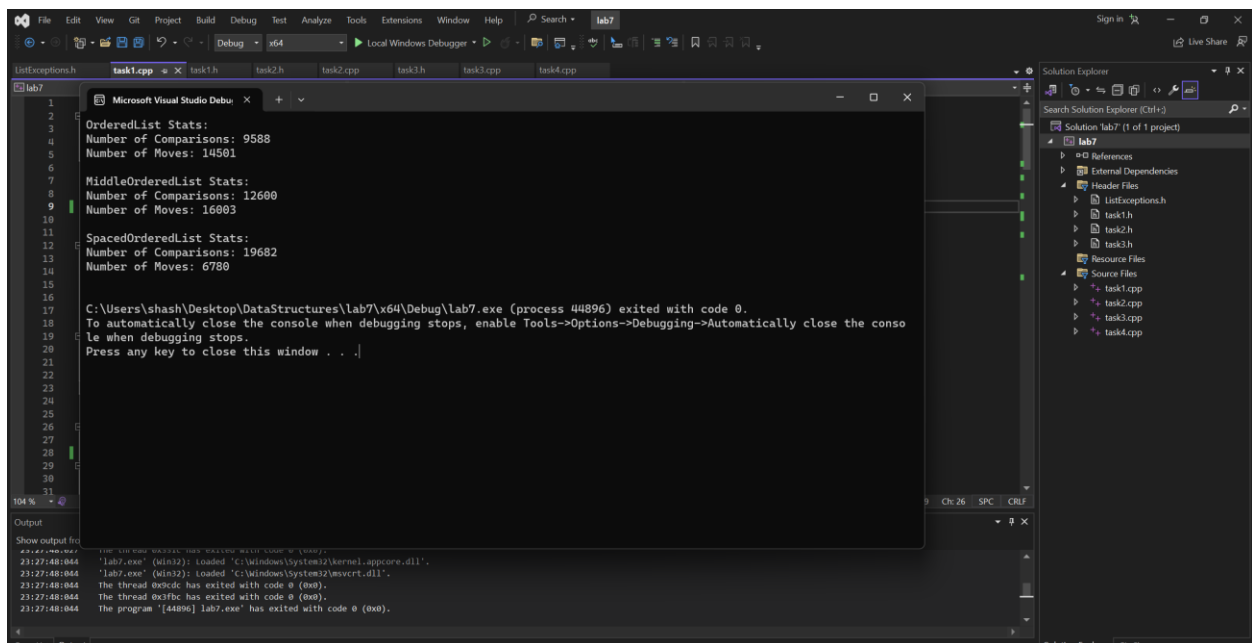
2.

In task 1, I created a ordered list class using an array. This class is a template and includes methods for adding and removing items while maintaining order. It also includes error handling to address common issues, such as adding an item to a full list.

In task 2 involved creating a derived class from the basic ordered list. This version starts insertions from the middle of the array and investigates whether this strategy reduces the number of moves during insertions.

In task 3, another derived class was created, exploring the idea of leaving blank spots in the array during insertions to reduce the number of moves.

Output when array size = 25



```
1
2
3 OrderedList Stats:
4   Number of Comparisons: 9588
5   Number of Moves: 14501
6
7 MiddleOrderedList Stats:
8   Number of Comparisons: 12600
9   Number of Moves: 16003
10
11 SpacedOrderedList Stats:
12   Number of Comparisons: 19682
13   Number of Moves: 6780
14
15
16 C:\Users\shash\Desktop\DataStructures\lab7\x64\Debug\lab7.exe (process 44896) exited with code 0.
17 To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
18 Press any key to close this window . . .
19
20
21
22
23
24
25
26
27
28
29
30
31
```

Output

```
23:27:48:044 The thread 0x1fbc has exited with code 0 (0x0).
23:27:48:044 The program '[44896] lab7.exe' has exited with code 0 (0x0).
```

Output when array size = 50

```
#pragma once
#include "task2.h"
#include "task1.h"
#include "task3.h"
using namespace std;

// Global Scope
// Global Variables
// Global Constants
// Global Functions

// Local Scope
// Local Variables
// Local Constants
// Local Functions

// Main Function
int main()
{
    // OrderedList Stats
    cout << "OrderedList Stats:" << endl;
    cout << "Number of Comparisons: 9101" << endl;
    cout << "Number of Moves: 14324" << endl;

    // MiddleOrderedList Stats
    cout << "MiddleOrderedList Stats:" << endl;
    cout << "Number of Comparisons: 12602" << endl;
    cout << "Number of Moves: 16024" << endl;

    // SpacedOrderedList Stats
    cout << "SpacedOrderedList Stats:" << endl;
    cout << "Number of Comparisons: 18715" << endl;
    cout << "Number of Moves: 6877" << endl;

    return 0;
}
```

C:\Users\shash\Desktop\DataStructures\lab7\Debug\lab7.exe (process 17724) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

Output when array size = 10

```
#pragma once
#include "task2.h"
#include "task1.h"
#include "task3.h"
using namespace std;

// Global Scope
// Global Variables
// Global Constants
// Global Functions

// Local Scope
// Local Variables
// Local Constants
// Local Functions

// Main Function
int main()
{
    // OrderedList Stats
    cout << "OrderedList Stats:" << endl;
    cout << "Number of Comparisons: 9455" << endl;
    cout << "Number of Moves: 14101" << endl;

    // MiddleOrderedList Stats
    cout << "MiddleOrderedList Stats:" << endl;
    cout << "Number of Comparisons: 12496" << endl;
    cout << "Number of Moves: 16274" << endl;

    // SpacedOrderedList Stats
    cout << "SpacedOrderedList Stats:" << endl;
    cout << "Number of Comparisons: 19384" << endl;
    cout << "Number of Moves: 6835" << endl;

    return 0;
}
```

C:\Users\shash\Desktop\DataStructures\lab7\Debug\lab7.exe (process 41740) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

As seen in all the outputs, the ordered list appears to have the fewest number of comparisons but a relatively higher number of moves. It starts at the front of the array (index 0) and may involve fewer comparisons because it always inserts at the beginning, but moving items to make space can result in more moves.

MiddleOrderedList version has a higher number of both comparisons and moves compared to the basic ordered list. Starting from the middle (index 12) requires more comparisons, and there are still a significant number of moves.

SpacedOrderedList version has the highest number of comparisons but significantly fewer moves. It inserts items halfway between existing items in the array, which requires more comparisons but reduces the number of moves.

Therefore, the "SpacedOrderedList" appears to be the most efficient in moves as it has the lowest number of moves although it has the most number of comparisons. On the other hand "OrderedList" has the fewest comparisons and a higher number of moves.