Logo, company name

Description automatically generated

**School of Computing, Electrical and Applied Technology**

ISCG6426

Data Structures & Algorithms

# Assignment

**Semester 2, 2021**

**Due Date: Week 13 class**

**Total Marks:** 40

**Course Weighting:** 40%

**Learning outcomes covered in this assignment**

1. Apply object-oriented design and implementation techniques.
2. Interpret the trade-offs and issues involved in the design, implementation, and application of various data structures with respect to a given problem.
3. Explain the purpose and answer questions about data structures and design patterns that illustrate strengths and weaknesses with respect to resource consumption.
4. Assess the impact of data structures on algorithms.
5. Analyse the scalability of data structures and algorithms in terms of both space and time complexity.

**Cover Sheet**

|  |  |
| --- | --- |
| First Name | Priyam |
| Last Name | Patel |
| Student ID | 1512428 |
| Chosen Data Structure(s) or Algorithm(s) | Singly Linked List |

By submitting files and/or work to the approved Moodle submission link for this assessment, I declare that all work has been performed by myself unless explicitly declared. Any code not created by me has been cited adequately. I accept that failure to comply with the Unitec Guidelines of Appropriate Student Conduct will result in enforcement of the relevant consequences.

Signed: A picture containing hanger

Description automatically generated Date:21/10/2021

## Documentation

### Data Structure / Algorithm

*Introduce and describe your chosen DS/A. This may include a brief history, and its purpose.*

*A linked list is a way to store a collection of elements. Like an array these can be character or integers. Each element in a linked list is stored in the form of a node. Each node contains a data field and a reference(link) to the next node in the list.* *The last node of the list contains pointer to the null.*

*Singly linked list is the simplest type of linked list in which every node contains some data and a pointer to the next node of the same data type. The node contains a pointer to the next node means that the node stores the address of the next node in the sequence. A single linked list allows traversal of data only in one way.*

### Strengths & Weaknesses

*Discuss the aspects or characteristics of the DS/A that are good, bad, or somewhere in between. This is a good section to cover time and space complexity (use the bigocheatsheet for help with this). You can reflect on how the intent of the DS/A aligns with the performance characteristics as to whether it is an effective tool.*

***Strength of Linked List are:***

* *The linked list is a dynamic data structure.*
* *We can also decrease and increase the linked list at run-time. That is, you can allocate and deallocate memory at run-time itself.*
* *The access time is very fast, and it can be accessed at a certain time without memory overhead.*
* *In this, you can easily do insertion and deletion functions. That is, you can easily insert and delete the node.*
* *We can easily implement linear data structures using the linked list like a stack, queue.*
* *Memory is well utilized in the linked list. Because in it, we do not have to allocate memory in advance.*

***Weakness of Linked List are:***

* *The linked list requires more memory to store the elements than an array, because each node of the linked list points a pointer, due to which it requires more memory.*
* *It is very difficult to traverse the nodes in a linked list. In this, we cannot access randomly to any one node. (As we do in the array by index.) For example: – If we want to traverse a node in an n position, then we have to traverse all the nodes that come before n, which will spoil a lot of our time.*
* *Reverse traversing in a linked list is very difficult, because it requires more memory for the pointer.*
* *Random access is not possible in a linked list due to its dynamic memory allocation.*

### Real-World Example

*List and briefly describe at least one use of your DS/A in a real-world use case.*

***Real-world examples of Linked List are:***

* *It is used to implement stacks and queues which are like fundamental needs throughout computer science.*
* *To prevent the collision between the data in the hash map, we use a singly linked list.*
* *If we ever noticed the functioning of a casual notepad, it also uses a singly linked list to perform undo or redo or deleting functions.*
* *We can think of its use in a photo viewer for having look at photos continuously in a slide show.*
* *In the system of train, the idea is like a singly linked list, as if you want to add a Boggie, either you have to take a new boggie to add at last or you must spot a place in between boggies and add it.*

### Implementation

*Discuss how you implemented the DS/A in code, and how you changed it to be demonstrated for a presentation. This is your chance to show off your time and effort, and cover details that you may not have time to cover in a presentation format. This section and your demo will have significant overlap – use it to your advantage.*

*Screenshots, code snippets, and other supplementary sources are recommended for this section, but not required.*

I used c# console application to create this project.

Text

Description automatically generated

Fig1: Blue Ball Method

Text

Description automatically generated

Fig2: Method to check the ball collision for the edges

Text

Description automatically generated

Fig3: Linked List algorithm

Text

Description automatically generated

Fig4: Method to draw the circles and label the values

Text

Description automatically generated

Fig5: Method to add the ball

A picture containing shape

Description automatically generated

Fig6: SFML Output