

National University of Sciences and Technology  
School of Electrical Engineering and Computer Science  
Department of Computing

CS-250 Data Structures and Algorithms

Spring 2024

Assignment 1

**Linked Lists**

Announcement Date: 20<sup>th</sup> Feb 2024

Due Date: 27<sup>th</sup> Feb 2024 at 11:59 pm (on LMS)

Instructor: Bostan Khan

**Objective:** The objective of this assignment is to explore various applications of linked lists and gain hands-on experience with implementing and manipulating linked list data structures.

Please read the instructions for each task with great care and be sure to implement the requirements stated against each task.

**Tasks:**

1. **Implementation of Linked List [Marks: 5]:** Implement a linked list using classes in C++ for data type int. Your linked list should support the following operations:
  - Insertion of a new node at the beginning, middle, and end of the list.
  - Deletion of a node from the list.
  - Searching for a node based on a given value.
  - Traversing and printing the contents of the list.
2. **Application Scenarios [Marks: 40]:** Please write code for the following application scenarios where linked lists are commonly used. For each scenario, implement the necessary functionality using linked lists. You can use singly, doubly and circular linked lists. Please provide a separate .cpp file for each of the 4 tasks below.
  - **Employee Management System [10]:** Implement an employee management system where each employee is represented as a node in the linked list. Each employee must have the following fields: employee ID, name, contact\_number, age, salary, marital\_status. The system should support operations such as adding new employees, removing employees based on **searching for employee id**, updating employee information based on **searching for employee id**, and displaying the list of employees.
  - **Library Catalog System [15]:** Implement a library catalogue system that manages books in a library. Each book can be represented as a node in the linked list. The system should support functionalities such as adding new books, removing books, searching for books by title or author. There can be multiple books by the same author and same name. The system should also have the following functions:
    1. `books_by_author()`: Get an authors name as input and display number of books by that Author in the catalog.
    2. `display_catalogue()`: On execution, display all the books and number of copies of each book present in the list. This means that one book should be displayed only once with the number of copies in the list.

Implement the system and display its working in the documentation report.

- **Event Scheduler [10]:** Implement an event scheduler that manages upcoming events or appointments. Each event can be represented as a node in the linked list. Each event node should have a name, date, and event details. The system should support operations such as adding new events, removing events, updating event details, and displaying the list of upcoming events. Choose an appropriate format of your choice for the date. The events in the linked list should always be sorted based on date.

Whenever you add a new event, it should be added to the position with respect to its date.

- **Shopping Cart [10]:** Implement a shopping cart system for an online store. Each item in the shopping cart can be represented as a node in the linked list. Each node should have an item ID, item name and price. The system should support functionalities such as adding items to the cart, removing items, updating quantities for each item, calculating total price for the shopping cart (list), and displaying the contents of the cart.
3. **Documentation Report PDF [Marks: 30]:** Provide a report PDF for all your linked list implementations, including explanations of the data structures and algorithms used, as well as any design decisions made for each task. Document the usage of your linked list in each of the application scenarios, including how the linked list data structure is utilized to solve the problem. **The marks of your coding tasks will be directly influenced by the documentation report.** Include clear instructions on how to compile/run your code and how to interact with the implemented applications.
- **Testing Results in Documentation [Marks: 20]:** Test your implementations thoroughly to ensure correctness and robustness. Provide sample input-output scenarios for each application scenario to demonstrate the functionality of your linked list implementations.

**Remember to document your work well in the documentation report. Explain the code, its design and test results with details for obtaining max marks for the whole assignment.**

4. **Submission:** You will have to submit a separate .cpp source file for each task. Present The documentation report including the test results in a well-prepared PDF.
- The report PDF and the sources files should be compressed in a single zip file and submitted on LMS with the name in following format: <Student\_name>\_<Reg\_no>.zip**

**Additional Notes:**

- You are encouraged to be creative and innovative in your implementations. Consider additional features or optimizations that could enhance the functionality or performance of your applications.
- Collaboration with classmates is allowed for discussing concepts and problem-solving strategies, but each student must submit their own individual solution.
- **Plagiarism or copying of code from other students will result in zero marks.**