# Turnitin Originality Report

Processed on: 30-Dec-2023 00:08 +08
ID: 2265515841
Word Count: 1212
Submitted: 1

Assignment 2 By MUHAMMAD DANIAL

| Similarity Index | Similarity by Source | |
|---|---|---|
| 0% | Internet Sources: | 0% |
| | Publications: | 0% |
| | Student Papers: | 0% |

Department of Computer Science Faculty of Computing Assignment 2 (Linked List Implementation) Programme Subject Code Subject Name Session-Sem : Bachelor of Computer Science (Data Engineering) : SECJ2013 : Data Structure & Algorithm : 2023/2024-1 Prepared by : 1) MUHAMMAD DANIAL BIN AHMAD SYAHIR (A22EC0206) 2) DANIAL HARRIZ BIN MOHD ASINEH @ MOHD ASNEH (A22EC0152) 3) THEVAN RAJU A/L JEGANATH (A22EC0286) Section : 02 Group : DTD Lecturer : Dr. Lizawati Mi Yusuf 1.0 Objective..... 4 2.0 Synopsis..... 4 3.0 Design..... 4 4.0 Implementation Of Linked List..... 6 5.0 Implementation Of Sorting..... 6

1.0 Objective The primary objective of this program is to create an efficient library management system utilizing a linked list data structure. The system aims to provide essential functionalities such as adding, finding, deleting, sorting, and displaying books within the library. 2.0 Synopsis The program utilizes a linked list to store information about books. Each book is represented by an instance of the Book class. The linked list is implemented using the Node and Library classes. The Library class provides methods to perform various operations on the linked list, such as adding, deleting, finding, sorting, and displaying books. The main function interacts with the user through a menu, allowing them to manage the library efficiently. 3.0 Design Class Diagram Pseudocode : 1. Start the program. 2. Display the main menu. 2.1. Display the menu options such as “Add Book”, “Find Book”, “Delete Book”, “Sort Books”, “Display Books”, and “Exit”. 3. 2.2. Users are prompted to enter a choice. If the user enters choice : 1, users will be able to add books into the existing book list. 3.1. Sub-menu for adding books at Front, Middle, or End will be displayed. 3.2. Users are then prompted to enter a choice based on the 3 choices. 3.3. After entering a choice, users will enter book details. 3.4. The book will be added into the list and the updated list will be displayed. 3.5. A question will be asked whether the user still wants to continue using the program. 3.5.1. If the user enters “Y” or “y”, the program will go back to the main menu. 3.5.2. If the user enters “N” or “n”, jump to step 8. 4. If the user enters choice : 2, users will be able to find books from the existing book list. 4.1. Sub-menu for users to enter a search key. 4.2. Users will need to enter either the book title or ISBN number of the book. 4.3. The book will be searched from the list of books. 4.4. Display the found book details or a "not found" message 4.5. A question will be asked whether the user still wants to continue using the program. 4.5.1. If the user enters “Y” or “y”, the program will go back to the main menu. 4.5.2. If the user enters “N” or “n”, jump to step 8. 5. If the user enters choice : 3, users

will be able to delete books from the existing book list. 5.1. Sub-menu for users to enter a delete key. 5.2. Users will need to enter either the book title or ISBN number of the book. 5.3. The book will be deleted from the list of books. 5.4. Display the updated list of books. 5.5. A question will be asked whether the user still wants to continue using the program. 5.5.1. If the user enters “Y” or “y”, the program will go back to the main menu. 5.5.2. If the user enters “N” or “n”, jump to step 8. 6. If the user enters choice : 4, users will be able to sort books in the existing book list. 6.1. Sub-menu for sorting options will be displayed. The books can be sorted by title, author, year or ISBN number. 6.2. Users are then prompted to enter a choice based on the 4 choices. 6.3. After entering a choice, the books will be sorted. 6.4. The updated list will be displayed. 6.5. A question will be asked whether the user still wants to continue using the program. 6.5.1. If the user enters “Y” or “y”, the program will go back to the main menu. 6.5.2. If the user enters “N” or “n”, jump to step 8. 7. If the user enters choice : 5, the current list of books will be displayed. 7.1. A question will be asked whether the user still wants to continue using the program. 7.1.1. If the user enters “Y” or “y”, the program will go back to the main menu. 7.1.2. If the user enters “N” or “n”, jump to step 8. 8. If the user enters choice : 6, the program will stop. 8.1. A thank you message will be displayed. 9. End. 4.0 Implementation Of Linked List The linked list implementation is a crucial aspect of the program, as it enables dynamic management of book data. The Library class encapsulates operations such as adding books to the beginning, middle, and end of the list, finding and deleting books based on title or ISBN, and sorting books by title, author, year, or ISBN. The addNode method facilitates the addition of a new book node at the front of the linked list.By creating a new node with the book's information and updating the head, this operation maintains chronological order in the library. This operation is crucial for efficiently inserting new books and maintaining a chronological order within the library. The addNodeMiddle operation enhances the library's functionality by allowing the insertion of a new book node at the middle position of the linked list. This operation calculates the middle position based on the current size of the list and intelligently inserts the new node, providing a more organized structure. The addNodeEnd operation provides users with the flexibility to append new books to the end of the linked list. it traverses the list to find the last node and seamlessly integrates the new book, accommodating users who prefer appending new entries.This is beneficial when users prefer to add books without disrupting the existing order or when dealing with a continuously expanding collection. The findNode method searches for a book node in the linked list based on the title or ISBN. This operation is essential for retrieving specific books from the library. Utilizing a pointer to traverse the list and compare titles or ISBNs, this operation returns a pointer to the matching node or null if no match is found. The deleteNode method enables the removal of a book node from the linked list based on the title or ISBN. Employing two pointers to traverse and locate the target node, this operation ensures the list remains updated and accurate in its representation of the library. 5.0 Implementation Of Sorting In the program, sorting is implemented using straightforward algorithms, such as the bubble sort algorithm, for simplicity and ease of understanding. Each sorting function corresponds to a specific criterion: title, author, year, or ISBN. The sorting algorithms iterate through the array and compare adjacent elements, swapping them if they are out of order.