DATA STRUCTURE AND ALGORITHM FINAL

PROJECT

TOPIC:

“BOOK SHOP PROJECT”

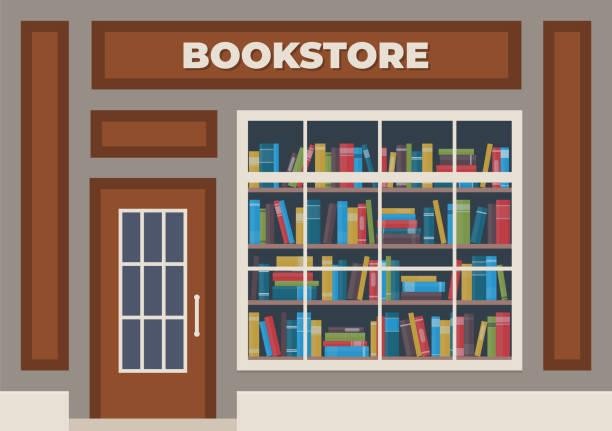
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **SUBMITTED TO:**

**DR. SIDRA EJAZ**

**SUBMITTED BY:**

**MARYAM WASEEM (2022-BSE-059)**

**Date:1/1/2024**



# Report

# Overview:

# This C++ program implements a basic bookshop management system using a single linked list. The program allows users to insert, search, update, and delete books, as well as display all books in the bookshop. The book details are stored in a linked list, and the program uses file handling to persist data.

# Key Components:

# 1. Data Structure:

# Bookshop Structure:

# id: Book ID (integer)

# price: Book Price (float)

# name: Book Name (string)

# a\_name: Author Name (string)

# p\_name: Publisher Name (string)

# next: Pointer to the next book in the linked list

# 2. Functions:

# Heading:

# Displays a welcome message.

# Insert:

# Takes user input for book details and inserts a new book at the beginning of the linked list.

# Appends the book details to a file (bookshop\_file.txt).

# Search:

# Searches for a book by ID or name in the linked list.

# Displays book details if found.

# Update:

# Updates the details of a book based on the provided ID.

# Takes user input for the new book details.

# Delete:

# Deletes a book based on the provided ID.

# Adjusts pointers in the linked list accordingly.

# Show:

# Displays details of all books in the linked list.

# main:

# Initializes an array of Bookshop objects with group members' names.

# Displays the names.

# Provides a menu for interacting with the bookshop system:

# Insert a new book.

# Search for a book.

# Update book details.

# Delete a book.

# Show all books.

# Exit the program.

# Execution:

# Insert Book:

# Prompts the user to enter book details (ID, name, author, publisher, price).

# Inserts the book at the beginning of the linked list.

# Appends the book details to the file.

# Search Book:

# Prompts the user to enter either the book ID or name for search.

# Searches for the book in the linked list.

# Displays book details if found; otherwise, displays a "Not Found" message.

# Update Book:

# Prompts the user to enter the book ID for updating.

# Modifies the book details if the book is found in the linked list.

# Delete Book:

# Prompts the user to enter the book ID for deletion.

# Deletes the book if found in the linked list.

# Show All Books:

# Displays details of all books in the bookshop.

# Exit:

# Exits the program.

# Recommendations:

# The program effectively utilizes a single linked list for managing book details.

# File handling is used for persistent data storage, and the implementation is well-structured.

# The user input validation ensures data integrity.

# Potential Improvements:

# Consider adding error handling for file operations (e.g., file opening, writing).

# Enhance user interface and error messages for a more user-friendly experience.

# Implement additional features such as sorting books by ID or price.

# Add memory deallocation (destructor) for freeing allocated memory when the program exits.

# Conclusion:

# This bookshop management system provides essential functionalities for handling books in a bookshop. The program uses a single linked list to efficiently manage book data, and file handling ensures that data is persistent across program executions. Further enhancements and additional features can be considered for future improvements.

# Top of Form

# Top of Form

# 

# SOME MORE INSIGHT

# Here's a brief overview of the key components and functionalities:

# Data Structure:

# INTRODUCTION

In bookshop [management system](https://www.codewithc.com/sales-management-system-project-c/) project , users can add book records, show book records, modify book records,enter price of book, delete book records.It will reduce the paper work and work load of user.to reduce the paper work and provide fast service to customers.The main objective is to provide the customer fast and error free transaction.The header files we have used

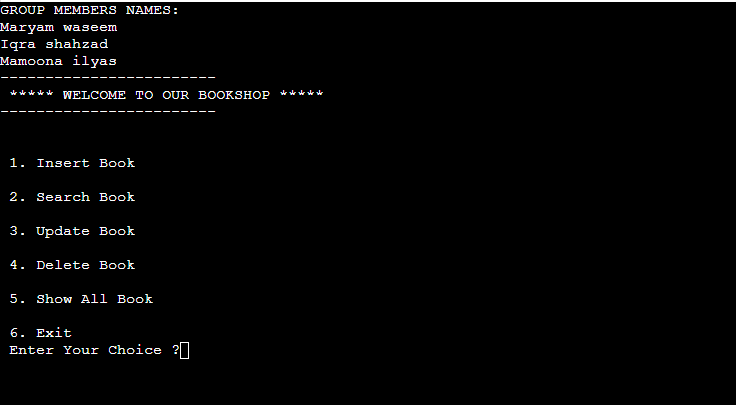
#include<iostream #include<conio>

#include <limits>

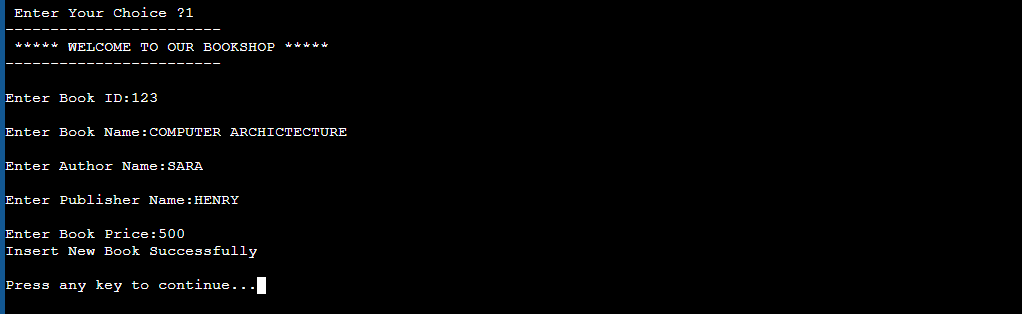
#include <algorithm>

#include<fstream>

**This is the output shown when the programs begin to run.**

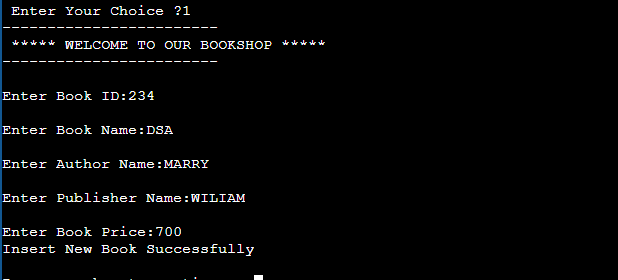


**When user enter 1 the following is displayed on the output screen.**

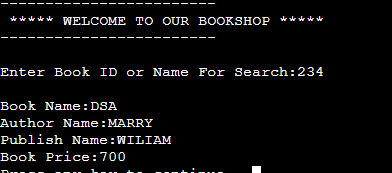


**Again enter your choice**

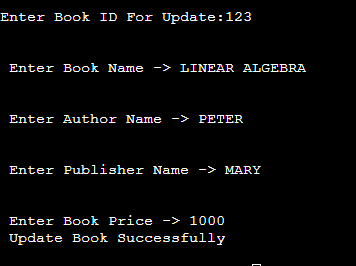
**When user enter 1 the following is displayed on the output screen.**



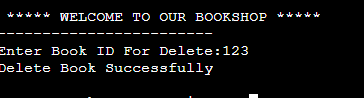
**When user enter 2 the following is displayed on the output screen.**



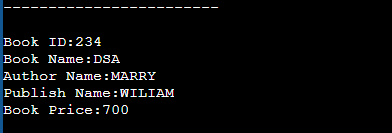
**On pressing 3 the following will be displayed**



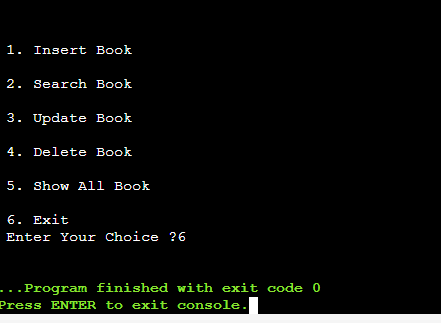
**On pressing 4 the following will be displayed**



**On pressing 4 the following will be displayed**



**Pressing 6 all the data will exit**



MAINCODE

//Book shop Project Using LINKLIST

#include <iostream>

#include <fstream>

#include <limits>

#include <algorithm> // Include the algorithm header for all\_of

using namespace std;

struct Bookshop

{

int id;

float price;

string name, a\_name, p\_name;

Bookshop\* next;

};

**// Heading Function**

void Heading()

{

cout << "------------------------\n";

cout << " \*\*\*\*\* WELCOME TO OUR BOOKSHOP \*\*\*\*\*";

cout << "\n------------------------\n";

}

**// Insert New Book**

void Insert(Bookshop\*& head)

{

ofstream f("E:\\bookshop\_file.txt", ios::out | ios::app);

if (f.is\_open())

{

Heading();

Bookshop\* newBook = new Bookshop;

cout << "\nEnter Book ID:";

while (!(cin >> newBook->id))

{

cout << "Invalid input. Please enter a valid integer for Book ID: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

cin.ignore();

cout << "\nEnter Book Name:";

getline(cin, newBook->name);

cout << "\nEnter Author Name:";

getline(cin, newBook->a\_name);

cout << "\nEnter Publisher Name:";

getline(cin, newBook->p\_name);

cout << "\nEnter Book Price:";

while (!(cin >> newBook->price))

{

cout << "Invalid input. Please enter a valid float for Book Price: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

newBook->next = head;

head = newBook;

f << newBook->id << "\t" << newBook->name << "\t" << newBook->a\_name << "\t" << newBook->p\_name << "\t" << newBook->price << "\n";

cout << "Insert New Book Successfully " << endl;

f.close();

}

else

{

cout << "Error opening the file for writing" << endl;

}

}

**// Search Book**

void Search(Bookshop\* head)

{

system("cls");

Heading();

if (!head)

cout << "Structure Is Empty " << endl;

else

{

string input;

cout << "\nEnter Book ID or Name For Search:";

cin.ignore();

getline(cin, input);

int found = 0;

Bookshop\* current = head;

if (all\_of(input.begin(), input.end(), ::isdigit))

{

int t\_id = stoi(input);

while (current != nullptr)

{

if (t\_id == current->id)

{

cout << "\nBook Name:" << current->name;

cout << "\nAuthor Name:" << current->a\_name;

cout << "\nPublish Name:" << current->p\_name;

cout << "\nBook Price:" << current->price;

found++;

break;

}

current = current->next;

}

}

else

{

while (current != nullptr)

{

if (input == current->name)

{

cout << "\nBook ID:" << current->id;

cout << "\nAuthor Name:" << current->a\_name;

cout << "\nPublish Name:" << current->p\_name;

cout << "\nBook Price:" << current->price;

found++;

break;

}

current = current->next;

}

}

if (found == 0)

cout << "Book Is Not Found " << endl;

}

}

**// Update Book**

void Update(Bookshop\* head)

{

system("cls");

Heading();

if (!head)

cout << "Structure Is Empty " << endl;

else

{

int t\_id, found = 0;

cout << "Enter Book ID For Update:";

while (!(cin >> t\_id))

{

cout << "Invalid input. Please enter a valid integer for Book ID: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

Bookshop\* current = head;

while (current != nullptr)

{

if (t\_id == current->id)

{

cout << "\n\n Enter Book Name -> ";

cin.ignore();

getline(cin, current->name);

cout << "\n\n Enter Author Name -> ";

getline(cin, current->a\_name);

cout << "\n\n Enter Publisher Name -> ";

getline(cin, current->p\_name);

cout << "\n\n Enter Book Price -> ";

while (!(cin >> current->price))

{

cout << "Invalid input. Please enter a valid float for Book Price: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

cout << " Update Book Successfully " << endl;

found++;

break;

}

current = current->next;

}

if (found == 0)

cout << "Book Is Not Found " << endl;

}

}

**// Delete Book**

void Delete(Bookshop\*& head)

{

system("cls");

Heading();

if (!head)

cout << "Structure Is Empty " << endl;

else

{

int t\_id, found = 0;

cout << "Enter Book ID For Delete:";

while (!(cin >> t\_id))

{

cout << "Invalid input. Please enter a valid integer for Book ID: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

if (head->id == t\_id)

{

Bookshop\* temp = head;

head = head->next;

delete temp;

cout << "Delete Book Successfully " << endl;

found++;

}

else

{

Bookshop\* current = head;

while (current->next != nullptr)

{

if (t\_id == current->next->id)

{

Bookshop\* temp = current->next;

current->next = current->next->next;

delete temp;

cout << "Delete Book Successfully " << endl;

found++;

break;

}

current = current->next;

}

}

if (found == 0)

cout << "Book Is Not Found " << endl;

}

}

**// Show All Books**

void Show(Bookshop\* head)

{

system("cls");

Heading();

if (!head)

cout << "Structure Is Empty " << endl;

else

{

Bookshop\* current = head;

while (current != nullptr)

{

cout << "\nBook ID:" << current->id;

cout << "\nBook Name:" << current->name;

cout << "\nAuthor Name:" << current->a\_name;

cout << "\nPublish Name:" << current->p\_name;

cout << "\nBook Price:" << current->price << endl;

current = current->next;

}

}

}

**// Main Function**

int main()

{

cout << "GROUP MEMBERS NAMES:\n";

Bookshop names[] = { {1, 0, "Maryam waseem", "", ""}, {2, 0, "Iqra shahzad", "", ""}, {3, 0, "Mamoona ilyas", "", ""} };

Bookshop\* ptrNames[3];

for (int j = 0; j < 3; j++)

{

ptrNames[j] = &names[j];

}

for (int j = 0; j < 3; j++)

{

cout << ptrNames[j]->name << endl;

}

Bookshop\* head = nullptr;

int choice;

while (true)

{

Heading();

cout << "\n\n 1. Insert Book";

cout << "\n\n 2. Search Book";

cout << "\n\n 3. Update Book";

cout << "\n\n 4. Delete Book";

cout << "\n\n 5. Show All Book";

cout << "\n\n 6. Exit\n";

cout << " Enter Your Choice ?";

cin >> choice;

switch (choice)

{

case 1:

Insert(head);

break;

case 2:

Search(head);

break;

case 3:

Update(head);

break;

case 4:

Delete(head);

break;

case 5:

Show(head);

break;

case 6:

exit(0);

default:

cout << " Please Select Correct Option " << endl;

}

cout << "\nPress any key to continue...";

cin.ignore();

cin.get();

system("cls");

}

return 0;

}