**(SECJ1013) PROGRAMMING TECHNIQUE 1**

**SEM 1, SESSION 2023/2024**

**ASSIGNMENT 3**

**INSTRUCTIONS TO THE STUDENTS**

* This exercise must be done **in a group of two**.
* Any form of plagiarism is **NOT ALLOWED**. Students who copied other students' assignments will get **ZERO** marks (both parties, students who copied, and students who shared their work).
* Please put your **name and matric number** and your member’s **name and matric number** **AS COMMENTS IN THE BEGINNING OF THE CODE**.

**SUBMISSION PROCEDURE**

* Please submit this exercise no later than **December 29, 2023, Friday (00:00 MYT)**.
* Only SOURCE CODE files (**.CPP**) is required for the submission (**NOT ALLOWED .ZIP** **FILE**).
* Only **ONE** submission per pair (group).
* Submit the assignment via the UTM's e-learning system (<https://elearning.utm.my/23241/>).

# SET 1

Write a complete C++ program for a simple demonstration of Library Management System, which allows users to add books, display the book list, and search for books by title.

The program should perform the following tasks:

1. Define the maximum number of books that the library can hold as a constant variable `**MAX\_BOOKS**`.
2. Create arrays to store book information, including titles, authors, and publication years, with each array having a capacity of `**MAX\_BOOKS**`.
3. Define function prototypes for the main functionalities of the library management system, including displaying the main menu, adding a book, displaying the library contents, and searching for a book by title.
4. Implement the `**main**` function to control the flow of the program, allowing users to interact with the library management system through a menu-driven interface.
5. Within the `**main**` function, display the main menu and prompt the user to input their choice, then use a switch statement to execute the corresponding functionality based on the user's choice.
6. Create a function `**displayMainMenu**` to display the main menu options for the library management system, providing users with the ability to add a book, display the library contents, search for a book by title, or quit the program.
7. Develop the `**addBook**` function to allow users to add a new book to the library, capturing the book title, author name, and publication year from the user input, and updating the book count.
8. Implement the `**displayLibrary**` function to present the contents of the library, including the titles, authors, and publication years of the stored books, ensuring a clear and organized display for the user.
9. Create the `**searchByTitle**` function to enable users to search for a book by its title, providing detailed information about the book if found, and notifying the user if the book is not found in the library.
10. Incorporate input validation and error handling mechanisms to ensure that user inputs are correctly handled and processed, preventing potential errors or unexpected behavior.
11. Test the functionality of the library management system to ensure that it operates as intended, allowing users to add books, display the library contents, search for books, and exit the program seamlessly.

**SAMPLE PROGRAM EXECUTION**

|  |
| --- |
| <<<<<Library Management System>>>>>  ========================================  1. Add a Book  2. Display Library  3. Search by Title  4. Quit  Enter your choice: 1  Enter book title: Programming  Enter author name: Ali  Enter publication year: 2020  Book added successfully!  <<<<<Library Management System>>>>>  ========================================  1. Add a Book  2. Display Library  3. Search by Title  4. Quit  Enter your choice: 2  Library Contents:  ====================  Title: Programming  Author: Ali  Year: 2020  <<<<<Library Management System>>>>>  ========================================  1. Add a Book  2. Display Library  3. Search by Title  4. Quit  Enter your choice: 3  Enter the title to search: Programming  Book found:  ====================  Title: Programming  Author: Ali  Year: 2020  <<<<<Library Management System>>>>>  ========================================  1. Add a Book  2. Display Library  3. Search by Title  4. Quit  Enter your choice: 4  Goodbye!  -------------------------------- |

**SET 2**

Write a complete C++ program for a simple demonstration of Library Management System, which allows users to add books, display the book list, and search for books by title.

The program should perform the following tasks:

1. Define a constant variable `**MAX\_OPERATIONS**` with a value of 100, representing the maximum number of mathematical operations that the program can handle.
2. Create arrays `**operands1**` and `**results**` to store the operands and results of the mathematical operations, with a capacity of ` **MAX\_OPERATIONS** `.
3. Implement a function `**multiplyUsingAddition**` that takes two integers `**a**` and `**b**` as input and returns the result of multiplication using repeated addition.
4. Develop a function `**displayMainMenu**` to display the main menu, providing options for performing multiplication, displaying results, and quitting the program.
5. Create a function `**performMultiplication**` that takes a reference to `**operationCount**` as input and allows the user to specify the number of operands for multiplication. The function then prompts the user to input the operands and performs multiplication using repeated addition, storing the result and the number of operands in the respective arrays.
6. Design a function `**displayResults**` that takes `**operationCount**` as input and displays the results of the mathematical operations, including the operation number, the result, and the number of operands used for each operation.
7. In the `**main**` function, initialize `**operationCount**` to 0 and prompt the user to input their choice through a menu-driven interface. Based on the user's choice, the program will either perform multiplication, display the results, or exit the program.
8. Ensure that the program handles input validation, providing feedback for invalid choices and preventing unexpected behavior.
9. Utilize appropriate comments and descriptive variable names to enhance code readability and maintainability.

**SAMPLE PROGRAM EXECUTION**

|  |
| --- |
| <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 1  Enter the number of operands for multiplication: 2  Enter operand 1: 1  Enter operand 2: 2  Multiplication performed successfully!  <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 2  Results of Mathematical Operations:  ========================================  Operation 1: 2 (Operands: 2)  <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 1  Enter the number of operands for multiplication: 3  Enter operand 1: 4  Enter operand 2: 5  Enter operand 3: 6  Multiplication performed successfully!  <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 2  Results of Mathematical Operations:  ========================================  Operation 1: 2 (Operands: 2)  Operation 2: 120 (Operands: 3)  <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 1  Enter the number of operands for multiplication: 5  Enter operand 1: 6  Enter operand 2: 7  Enter operand 3: 8  Enter operand 4: 9  Enter operand 5: 0  Multiplication performed successfully!  <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 2  Results of Mathematical Operations:  ========================================  Operation 1: 2 (Operands: 2)  Operation 2: 120 (Operands: 3)  Operation 3: 0 (Operands: 5)  <<<<<Main Menu>>>>>  =============================  1. Perform Multiplication  2. Display Results  3. Quit  Enter your choice: 3  Goodbye!  -------------------------------- |