CCP6124 Object Oriented Programming and Data Structures

Trimester 2520

Assignment

Criteria	Basic 0-20%	Developing 21-40%	Proficient 41-60%	Good 61-80%	Excellent 81-100%
Content Accuracy & Completeness	No solution provided or evidence of plagiarism. Minimal effort, incorrect or incomplete solution. No outputs shown.	Partial solution with significant errors or missing parts. Some outputs shown but incomplete or incorrect	Mostly correct solution with minor issues. Outputs mostly shown but may have small inaccuracies.	Correct and complete solution with all outputs shown. Follows instructions accurately.	Outstanding solution, fully accurate, and well-documented. All outputs correct, with clear and exemplary execution.

Question 1 (25 Marks)

Bank Account Management System Using Singly Linked List in C++

Object-Oriented Bank Account Management System Using Singly Linked List in C++

Design and implement a **Bank Account Management System** in C++ using **Object-Oriented Programming (OOP)** concepts and **singly linked lists**.

Your system should define appropriate **classes**, encapsulating data and behaviors using principles such as **encapsulation**, **modularity**, and **dynamic memory management**.

Each node in the list must represent a bank account with:

- Account Number
- Customer Name
- Account Balance

Implement class methods to:

1. Add a new account (prevent duplicates)
2. Display all accounts
3. Search by account number
4. Deposit and withdraw (with balance validation)
5. Delete an account
6. Exit the system

Demonstrate use of:

- Class encapsulation
- Proper memory handling with constructors/destructors
- A menu-driven interface for user interaction

Marking Rubric (Total: 25 Marks)

Criteria	Marks
Class design & OOP principles (encapsulation, constructors, destructors)	5
Linked list implementation (add, search, display)	5
Functionality: deposit, withdraw, error handling	5
Dynamic memory management	4
Menu-driven interface & usability	3
Code readability & comments	3

Question 2 (15 Marks)

Warehouse Inventory and Shipping System Using Stack and Queue in C++

You are required to build a simple warehouse system in C++ that uses a **stack** to manage incoming items (LIFO) and a **queue** to manage outgoing shipments (FIFO). The

program must allow users to add new inventory, process it into the shipping queue, and handle shipments. Additionally, the user must be able to view the most recent item received and the next item to ship. Ensure proper code structure, clarity, and comments.

Sample Output Preview

1. Add Incoming Item

2. Process Incoming Item

3. Ship Item

4. View Last Incoming Item

5. View Next Shipment

6. Exit

Enter your choice: 1

Enter item name: Laptop

Item "Laptop" added to inventory.

Enter your choice: 1

Enter item name: Monitor

Item "Monitor" added to inventory.

Enter your choice: 4

Last incoming item: Monitor

Enter your choice: 2

Processed "Monitor" and added to shipping queue.

Enter your choice: 2

Processed "Laptop" and added to shipping queue.

Enter your choice: 5

Next item to ship: Monitor

Enter your choice: 3

Shipping item: Monitor

Enter your choice: 3

Shipping item: Laptop

Enter your choice: 3

No items to ship.

Enter your choice: 6

Exiting...

}

Marking Rubric (Total: 15 Marks)

Criteria	Marks
Correct implementation of Stack operations (push, pop, display)	3
Correct implementation of Queue operations (enqueue, dequeue, display)	3
Proper integration of Stack and Queue to simulate warehouse inventory and shipping flow	3
Menu-driven interface with clear user interaction	2
Proper use of functions, code organization, and readability	2
Error handling (e.g., underflow/overflow, invalid input)	2