242212

Air University

(Mid-Term Examination: Spring 2025)

Subject: **Object-Oriented Programming**
Course Code: **CS-112**
Class: **BS-CYS**
Semester: **IV**
Section: **Evening A/B**

HoD Signatures:

Total Marks: **50**
Date:
Time:
Duration: **2 Hours**
FM Name: **Khwaja Mansoor**

FM Signatures:

Note:

- **All questions must be attempted.**
- **This examination carries 25% weight towards the final grade.**
- **<u>The Exam is Open Book.</u> Students are allowed to use only hard copy materials, whether printed or handwritten. The use of any electronic devices, including laptops, mobile phones, tablets, or digital notes, is strictly prohibited. Additionally, students are not allowed to share or exchange any study materials during the exam.**
- **Answers must be precise and to the point—focus on writing clean, well-structured code and providing a clear rationale for your implementation rather than simply restating concepts from the question.**

	Q. No. 1 (CLO 1)	**10 Marks**
	Differentiate the following pairs of concepts	
✓ a	Const Member Functions vs. Const Objects	5
✓ b	Composition vs. Aggregation in Object-Oriented Programming	5
	Q. No. 2 (CLO 2)	**20 Marks**
a	**A Library Management System allows users to borrow and return books while maintaining data integrity. The system should ensure that:** • A user cannot borrow more books than are available. • A user cannot return more books than they have borrowed. • The *displayStatus()* function should be a const function to prevent unintended modifications. However, the given implementation has logical and syntax errors that violate these rules. **Code:**	10

```
#include <iostream>
using namespace std;

class Library {
   string bookTitle;
   int totalCopies;
   int borrowedBooks;

public:
   Library(string title, int copies) {
      bookTitle = title;
      totalCopies = copies;
      borrowedBooks = 0;
    }

    void borrowBook(int count) {
      if (count <= totalCopies) {
        borrowedBooks += count;
        totalCopies -= count;
      }
      cout << count << " books borrowed." << endl;
    }

    void returnBook(int count) {
      totalCopies += count;
      borrowedBooks -= count;
      cout << count << " books returned." << endl;
    }

    void displayStatus() {
      cout << "Book: " << bookTitle << endl;
      cout << "Available Copies: " << totalCopies << endl;
      cout << "Borrowed Copies: " << borrowedBooks << endl;
    }
};

int main() {
    Library lib("Data Structures", 5);

    lib.borrowBook(3);
    lib.borrowBook(4);
    lib.returnBook(2);

    lib.displayStatus();
    return 0;
}
```

Your Task:

Dry run the code with the following test case:

- The library starts with 5 books.
- A user borrows 3 books.
- The user tries to borrow 4 more books (which should not be allowed).
- The user returns 2 books.

Identify and correct errors in the code while ensuring:

- Borrowing and returning logic works correctly.
- The *displayStatus()* function is correctly marked as const.
- Proper error messages are displayed for invalid operations.

10

Q. No. 3 (CLO 3)

20 Marks

a

You are tasked with designing a secure banking system where a customer's account details must be protected from unauthorized modifications. The system should allow users to view their account balance but not modify it directly. Only authorized bank operations should be able to update the balance, ensuring data integrity.

To achieve this:

- Implement a *BankAccount* class with private attributes: *accountNumber*, *accountHolderName*, and *balance*.
- Provide const member functions to allow customers to safely retrieve their balance and account details without modifying them.
- Implement a secure method (deposit and withdraw) that allows only the banking system to modify the balance under controlled conditions.
- Ensure that withdrawal transactions do not allow an overdraft (negative balance).

Write a main function that creates an account object, displays its details, and demonstrates both valid and invalid operations.

20

***************** End of Question Paper *****************