# Air University

**(Mid-Term Examination: Fall 2022)**

| | | | |
|---|---|---|---|
| Subject: | **Data Structures and Algorithms Lab** | Total Marks: | **50** |
| Course Code: | **CS-216L** | Date: | |
| Class: | **BS-CYS** | Time: | |
| Semester: | **III** | Duration: | **2 Hours** |
| Section: | **A** | FM Name: | Mr. Hassan Mazhar |

HoD Signatures:_____ FM Signatures:_____

**Note:**

- **All questions must be attempted.**
- **This examination carries 15% weight towards the final grade.**
- **Return the question paper with the answer sheet**

<table>
<tr><th></th><th>Q. No. 1 (CLO 1)</th><th>10 Marks</th></tr>
<tr><td>a</td><td>1. Instructions: For the following questions, analyze the provided code snippets in terms of their time complexity. Write the C++ code where indicated and describe the time complexity for each part.<br><br>Reverse an Array<br><table><tr><td><pre>#include <iostream>
using namespace std;

void reverseArray(int arr[], int n) {
    for (int i = 0; i < n / 2; i++) {
        int temp = arr[i];
        arr[i] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }
}</pre></td><td><pre>int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    reverseArray(arr, n);
    cout << "Reversed Array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}</pre></td></tr></table>Calulate its Time Complexity?</td><td>3</td></tr>
<tr><td>b</td><td>Write a C++ function that calculates the Fibonacci series up to the nth term. Calculate Time complexity as well</td><td>4</td></tr>
<tr><td>c</td><td>Implement a function to find the maximum element in an array. Calculate Time Complexity.</td><td>3</td></tr>
<tr><th></th><th>Q. No. 2 (CLO 2)</th><th>20 Marks</th></tr>
<tr><td>a</td><td>Scenario:<br>You are part of a team developing a basic management system for a library. The library needs various data structures to handle operations for books, records, and</td><td>4</td></tr>
</table>

| | | |
|---|---|---|
| | users. Choose the most suitable data structures for the following tasks and implement them to meet the system's requirements.<br><br>a. **Library Book Records Management:**<br>The library has numerous book records, each with a unique ID. You need to design and implement a system that:<br><br>• Allows adding a new book record at the beginning and end of the collection.<br>• Supports deleting a book record from a specified position.<br>• Can display all book records efficiently.<br><br>**Task:** Choose and implement the most suitable data structure for managing book records, ensuring that it meets the specified requirements.<br><br>• | |
| b | b. **User Borrowing History Management:**<br>The system also needs to manage the borrowing history of users. It should be able to:<br><br>• Add a new user record at a specified position.<br>• Delete a user record from a given position.<br>• Display the list of users in both forward and reverse order.<br><br>**Task:** Select and implement a data structure that best meets these requirements for managing user records. | 10 |
| c | c. **Checkout Request Processing:**<br>To handle book checkout requests in the library, you need a structure that:<br><br>• Allows adding a new checkout request.<br>• Supports removing the most recent checkout request.<br>• Displays all current checkout requests.<br><br>**Task:** Identify and implement the data structure that would be most effective for processing these checkout requests. | 6 |
| | **Q. No. 3 (CLO 3)** | **20 Marks** |
| a | You are tasked with designing a **Customer Service Queue** for a tech support center. Customers are served on a **first-come, first-served** basis. Implement a **Queue** using both an **array-based** and a **linked list-based** approach to handle the following operations:<br><br>1. **Add a customer to the queue**: When a new customer arrives, their name is added to the queue.<br>2. **Serve a customer**: When a customer is served, they are removed from the queue.<br>3. **Display the queue**: Show the current list of customers in the queue. | 20 |

| | | |
|---|---|---|
| | **Requirements:**<br><br>• Implement the **array-based queue** with the following constraints:<br>  ○ The maximum number of customers that can be in the queue at any given time is 5.<br>  ○ If a customer tries to join the queue when it's full, print a message indicating that the queue is full.<br>• Implement the **linked list-based queue** with the following constraints:<br>  ○ There is no fixed size limit for the queue.<br>  ○ Customers can be dynamically added and removed as needed.<br><br>**Marking Scheme:**<br>• Array-based implementation (Basic operations: enqueue, dequeue, display) – 5 Marks<br>• Linked list-based implementation (Basic operations: enqueue, dequeue, display) – 10 Marks<br>• VIP customer prioritization in both implementations – 5 Marks | |

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