



C-1102  
C-59

**Air University**  
(Mid-Term Examination : Fall 2025)

Subject: **Data Structures**  
Course Code: **CS-216**  
Class: **BS-CYS**  
Semester: **III**  
Section: **B**

Total Marks: **50**  
Date: \_\_\_\_\_  
Time: \_\_\_\_\_  
Duration: **2 Hours**  
FM Name: **Ms. Mehmoona Jabeen**

HoD Signatures: \_\_\_\_\_

FM Signatures: \_\_\_\_\_

**Note:**

- All questions must be attempted.
- This examination carries 25% weight towards the final grade.
- Return the question paper with the answer sheet

Q. No. 1 (CLO 1)		16 Marks
a.	Write an algorithm to find the minimum value stored in a Binary Search Tree (BST).	6
b.	For each of the following scenarios, choose the "best" data structure from the following list: an Array, Linked List, Doubly Linked List, Circular Linked List, Stack, Queue. In each case, briefly justify your answer.  i. A printer management system needs to manage print jobs in the order they are received, ensuring that the first job submitted is the first to be printed.  ii. A word processor must allow users to reverse their most recent actions, such as deleting text or formatting changes.  iii. A traffic signal controller cycles through a set of lights (red, yellow, green) repeatedly and must always continue to the next light in the loop.	6
c.	You are designing a music streaming application that stores recently played songs in a temporary buffer. The app can only remember a limited number of songs (fixed memory size). When a new song starts playing and the buffer is already full, the oldest song entry is automatically replaced by the newest one to keep the list up to date. You are required to: <ul style="list-style-type: none"><li>• Identify the most suitable data structure to manage this buffer.</li><li>• Write a complete algorithm that efficiently adds new songs, overwrites the oldest entry when full, and allows viewing the current song list.</li></ul>	4
Q. No. 2 (CLO 2)		16 Marks
d.	Assume a function MainLogin is currently executing, and it invokes another function ValidateUser. The MainLogin function must preserve its local variables in memory so they can be restored once ValidateUser finishes. Later, ValidateUser calls another function ConfirmOTP, which also needs to save its own local variables until execution completes. Identify and explain which data structure would be most suitable for managing such nested function calls and briefly describe how it works.	3+3

b	<p>In a program, you want to maintain the hierarchy of function calls using a tree structure, where each node represents a function call and stores its local variables. The root node represents the MainLogin function, which calls ValidateUser, and then ConfirmOTP.</p> <p>Required:</p> <ul style="list-style-type: none"> <li>• Design a tree-based structure that supports the following operations:</li> <li>• AddFunctionCall(child, parent): Add a new function call as a child of the given parent node.</li> <li>• EndFunctionCall(node): Remove a completed function call and update the hierarchy accordingly.</li> </ul> <p>Explain how these operations will maintain the relationship between functions in the call hierarchy.</p>	10
<b>Q. No. 3 (CLO 3)</b>		<b>18 Marks</b>
a	<p>i. Calculate the time complexity of the following code.</p> <pre> int x = 0, y = 0; for (int p = 0; p &lt; n; p++) {     if (p &lt; m) {         x = p + 1;         y = n - p;     } else {         for (int q = m; q &lt;= n; q++) {             x = x + q;             y = x - m + q;         }     } } </pre>	9
b	<p>i. How does the use of a circular queue solve the issue of 'back' reaching the end of the array? Does the time complexity of enqueue and dequeue operations remain <math>O(1)</math> with this solution? Why or why not?</p> <p>ii. In a Singly Linked List (SLL), removing the last node (tail) requires <math>O(n)</math> time complexity because you need to traverse the entire list to find the second-last node. What strategies or structural modifications can be employed to reduce the time complexity of deleting the last node from a list to <math>O(1)</math>? Please explain your approach in detail.</p> <p>iii. Find the time complexity of both functions used in the Q 02(a).</p>	3+3+3

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