



**Daffodil International University**  
**Department of Computer Science and Engineering**  
**Faculty of Science & Information Technology**  
**Midterm Examination, Spring 2025**  
**Course Code: CSE123, Course Title: Data Structures**  
**Level:1    Term:2    Batch: ALL**

**Time: 1.5 Hours**

**Marks: 25**

**Answer ALL Questions**

*[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]*

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                   |                       |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----------------------|
| 1. | <p>a) Explain the concept of a self-referential structure in C. Provide an example.</p> <p>b) Write necessary code using <b>C</b> language to implement a <b>Linear Linked List</b> with the following operations:</p> <ol style="list-style-type: none"> <li>1. <b>Insert at End</b></li> <li>2. <b>Delete from Beginning</b></li> <li>3. <b>Display the List</b></li> </ol> <p><b>Requirements:</b></p> <ul style="list-style-type: none"> <li>• Define node containing <i>data</i> (int) and a <i>pointer</i> (self referential).</li> <li>• Use dynamic memory allocation (malloc and free) for node creation and deletion.</li> <li>• Provide reasoning for using a linked list in applications like data storage for Library.</li> </ul> <p><i>(Full correct implementation: 7 marks, Partial correctness: 4-5 marks, Minor issues: 2-3 marks)</i></p> | <p>3</p> <p>7</p> | <p>CO1</p> <p>CO2</p> |
| 2. | <p>a) Why is dynamic memory allocation preferred when implementing linked lists? Explain with an example.</p> <p>b) Implement a <b>Stack</b> using a <b>Linked List</b> in C with the following operations:</p> <ol style="list-style-type: none"> <li>1. <b>Push</b> (Insert an element)</li> <li>2. <b>Pop</b> (Remove the top element)</li> <li>3. <b>Peek</b> (View the top element)</li> <li>4. <b>isEmpty</b> (Check if the stack is empty)</li> </ol> <p><b>Requirements:</b></p> <ul style="list-style-type: none"> <li>• Explain the time complexity of each operation.</li> </ul>                                                                                                                                                                                                                                                                  | <p>2</p> <p>8</p> | <p>CO1</p> <p>CO2</p> |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |     |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----|
|    | <ul style="list-style-type: none"> <li>• Provide appropriate output for sample inputs.</li> </ul> <p><i>(Full correct implementation: 8 marks, Partial correctness: 5-6 marks, Minor issues: 3-4 marks)</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |     |
| 3. | <p><b>Problem Solving (Analysis &amp; Synthesis Level)</b></p> <p><b>Problem Scenario:</b><br/>A printer management system requires efficient handling of print jobs. Each print job has a priority value. The system should process:</p> <ul style="list-style-type: none"> <li>• <b>High-priority jobs</b> immediately (Stack behavior: <u>LIFO</u>).</li> <li>• <b>Normal jobs</b> in the order they were received (Queue behavior: <u>FIFO</u>).</li> </ul> <p><b>Task:</b></p> <ol style="list-style-type: none"> <li>1. Propose a combination of Stack and Queue to design this system.</li> <li>2. Draw a flowchart or diagram to represent the process.</li> <li>3. Explain how the system handles multiple job requests with varying priorities.</li> </ol> <p><i>(Solution approach: 3 marks, Diagram: 1 mark, Explanation: 1 mark)</i></p> | 5 | CO2 |

Good Luck