



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.]

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

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

Good Luck