FINAL EXAMINATION

SPECIMEN PAPER 1

Module: DSA-251: Data Structure & Algorithm

Time Allowed: 3 Hours

Total Marks: 100

Instructions:

Answer all four questions.

Each question is worth 25 marks.

Read each question carefully before answering.

QUESTION 1 [25 MARKS]

(a) Define what an Abstract Data Type (ADT) is and provide one example.

[5]

(b) Differentiate between static memory allocation and dynamic memory allocation in C++.

[5]

(c) Describe the key structural difference between a singly linked list node and a doubly linked list node.

[5]

(d) State one primary advantage and one primary disadvantage of using a doubly linked list compared to a singly linked list.

[5]

(e) The C++ code below creates a simple linked list node. Draw a simple diagram that illustrates what this code creates in memory, showing the pointer and the node's contents.

[5]

DSA - Exam

```
1  // Assume Node class is defined
2  Node head = new Node(10);
```

QUESTION 2 [25 MARKS]

(a) Define the Stack data structure, including its operating principle (LIFO/FIFO) and a suitable real-world application.

[5]

(b) Define the Queue data structure, including its operating principle (LIFO/FIFO) and a suitable real-world application.

[5]

(c) Explain the algorithm for evaluating a postfix (RPN) expression using a stack. Your explanation should cover how to handle both operands (numbers) and operators.

[5]

(d) Given a stack whose current state is `[10, 50, 20]` (with 20 at the top), what is the state of the stack after the following two operations are performed in sequence: `pop()` then `push(40)`?

[5]

(e) The C++ code for an array-based `push()` method is given below with one line missing. Complete the code by writing the missing line.

[5]

DSA - Exam

```
1  void Stack::push(int value) {
2      if (isFull()) {
3          return; // Error
4      }
5      // -- MISSING LINE --
6      arr[top] = value;
7  }
```

QUESTION 3 [25 MARKS]

(a) Define the two essential components of any valid recursive function: the Base Case and the Recursive Step.

[5]

(b) Explain the role of the Call Stack in managing recursive function calls.

[5]

(c) Clearly define the Binary Search Tree (BST) Property.

[5]

(d) You are given an empty Binary Search Tree. Draw the resulting BST after inserting the following sequence of values in order: `30, 15, 45`.

[5]

(e) Given the C++ code for the recursive `factorial` function, identify the single line of code that represents the Base Case.

[5]

DSA - Exam

```
1  int factorial(int n) {
2      if (n == 0) {
3          return 1;
4      } else {
5          return n  factorial(n - 1);
6      }
7  }
```

QUESTION 4 [25 MARKS]

(a) Explain the purpose of a hash function and state one property of a good hash function.

[5]

(b) Describe how the separate chaining method works to resolve collisions in a hash table.

[5]

(c) Describe the exploration strategy of Breadth-First Search (BFS) and state the data structure it uses.

[5]

(d) Describe the exploration strategy of Depth-First Search (DFS) and state the data structure it uses (either explicitly or implicitly).

[5]

(e) [Programming] Write a complete C++ function `void printArray(int arr[], int size)` that prints all elements of an integer array on a single line, separated by spaces.

[5]

DSA - Exam

```cpp
include <iostream>

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}
```

END OF SPECIMEN PAPER