No. of Printed Pages : **8**

A5-R5.1 : Data Structure Through Object Oriented Programming Language

DURATION : 03 Hours **MAXIMUM MARKS : 100**

OMR Sheet No. :					

Roll No. : | | | | | | |

Answer Sheet No. : | | | | | | |

Name of Candidate : _____ **Signature of Candidate :** _____

INSTRUCTIONS FOR CANDIDATES

- Carefully read the instructions given on Question Paper, OMR Sheet and Answer Sheet.
- Question Paper is in English language. Candidate has to answer in English language only.
- There are **TWO PARTS** in this Module/Paper. **PART ONE** contains **FOUR** questions and **PART TWO** contains **FIVE** questions.
- **PART ONE** is Objective type and carries **40** Marks. **PART TWO** is Subjective type and carries **60** Marks.
- **PART ONE** is to be answered in the **OMR ANSWER SHEET** only, supplied with the question paper, as per the instructions contained therein. **PART ONE** is **NOT** to be answered in the answer book for **PART TWO**.
- Maximum time allotted for **PART ONE** is **ONE HOUR**. Answer book for **PART TWO** will be supplied at the table when the Answer Sheet for **PART ONE** is returned. However, Candidates who complete **PART ONE** earlier than one hour, can collect the answer book for **PART TWO** immediately after handing over the Answer Sheet for **PART ONE** to the Invigilator.
- **Candidate cannot leave the examination hall/room without signing on the attendance sheet and handing over his/her Answer Sheet to the invigilator. Failing in doing so, will amount to disqualification of Candidate in this Module/Paper.**
- After receiving the instruction to open the booklet and before answering the questions, the candidate should ensure that the Question Booklet is complete in all respects.

DO NOT OPEN THE QUESTION BOOKLET UNTIL YOU ARE TOLD TO DO SO.

PART - ONE

(Answer all the questions; each question carries ONE mark)

1. Each question below gives a multiple choice of answers. Choose the most appropriate one and enter in the "OMR" answer sheet supplied with the question paper, following the instructions therein. **(1x10)**

1.1 What is the difference between the virtual function and the pure virtual function ?

(A) A virtual function can be defined in the base class or overridden in the defined class, while a pure virtual function must be overridden in a derived class.

(B) A pure virtual function can be defined in the base class or overridden in a derived class, while a virtual function must be overridden in a derived class.

(C) There is no difference between a virtual function and a pure virtual function.

(D) A virtual function is used for polymorphism, while a pure virtual function is used for inheritance.

1.2 What is the time complexity of push and pop operations in a stack implemented using an array ?

(A) O(log n)

(B) O(n)

(C) O(1)

(D) Depends on the implementation

1.3 In which data structure does insertion and deletion take place from opposite ends?

(A) Stack

(B) Queue

(C) Deque

(D) Priority queue

1.4 Which of the following statements accurately compares the space complexity of Insertion Sort and Merge Sort ?

(A) Insertion Sort has a space complexity of O(n), while Merge Sort has O(1).

(B) Insertion Sort has a space complexity of O(1), while Merge Sort has O(n).

(C) Both Insertion Sort and Merge Sort have a space complexity of O(n).

(D) Both sorting algorithms have the same space complexity of O(log n).

1.5 What is a major advantage of using a threaded binary tree compared to a regular binary tree ?

(A) It ensures that all nodes have two children.

(B) It reduces the memory required for storing nodes.

(C) It allows for faster insertion and deletion operations.

(D) It allows faster traversal without using a stack or recursion.

1.6 What is the maximum number of edges in a complete directed graph with n vertices ?

(A) $n(n-1)$

(B) $n(n-1)/2$

(C) $n(n+1)$

(D) n^2

1.7 What is the main advantage of using a circular linked list in a situation where elements need to be accessed in a continuous loop, such as in a round-robin scheduling algorithm ?

(A) It simplifies memory management compared to arrays.

(B) It allows for O(1) access time to any element.

(C) It requires less memory overhead than a doubly linked list.

(D) It prevents the need to traverse the entire list to return to the starting point.

1.8 What are the advantages of passing arguments by reference ?

(A) Modifications to parameter values inside the function directly impact the original arguments.

(B) It eliminates the need to copy parameter values, resulting in reduced memory usage.

(C) Constructors for parameters do not need to be called, leading to improved performance.

(D) All of the above advantages are applicable.

1.9 Which of the following pairs of tree traversal sequences is **not sufficient** to uniquely reconstruct a binary tree ?

(A) Preorder and Inorder

(B) Preorder and Postorder

(C) Inorder and Postorder

(D) All of the above pairs are sufficient

1.10 Consider a graph represented as an adjacency list. You need to perform a traversal to find a specific node. Which of the following statements is true regarding the characteristics and applications of Depth First Search (DFS) and Breadth First Search (BFS)?

(A) DFS is more memory-efficient than BFS when exploring deep graphs due to its use of a stack, while BFS may require significant memory to store all the nodes at the current level.

(B) BFS always finds the shortest path in an unweighted graph, while DFS does not guarantee finding the shortest path.

(C) DFS is inherently recursive, while BFS is iterative and uses a queue for its implementation.

(D) All of the above statements are correct.

2. Each statement below is either TRUE or FALSE. Choose the most appropriate one and enter in the "OMR" answer sheet supplied with the question paper, following the instructions therein. (1x10)

2.1 In encapsulation, it allows for restricting access to certain components of an object.

2.2 A constructor is not a special member function, and it is implicitly not called by the compiler when an object of the class is created.

2.3 C++ supports both procedural and object-oriented programming paradigms.

2.4 Objects of the same class share the same memory for their data members.

2.5 Performing an in-order traversal on a binary search tree will yield the nodes in ascending order, regardless of the tree's balance or structure.

2.6 Polymorphism allows functions to work with objects of different classes at compile time only.

2.7 In row-major order, the elements of a two-dimensional array are stored in memory by filling each row completely before moving to the next row.

2.8 An adjacency matrix is typically more space-efficient than an adjacency list for representing sparse graphs.

2.9 In a circular linked list, the last node points to NULL, indicating the end of the list.

2.10 B-trees are designed to minimize the number of disk reads required during search operations, making them suitable for database indexing.

3. Match words and phrases in column X with the closest related meaning/ word(s) /phrase(s) in column Y. Enter your selection in the "OMR" answer sheet supplied with the question paper, following the instructions therein. (1x10)

X		Y	
3.1	The process of hiding the implementation details and showing only the essential features of an object	**A**	Notations
3.2	A collection of elements stored at contiguous memory locations, allowing for efficient indexing	**B**	Quick Sort
3.3	A collection of key-value pairs, allowing for efficient data retrieval	**C**	Hash
3.4	A tree structure where each node has at most two children	**D**	Pre-Order Traversal
3.5	In this traversal method, the root node is visited first, followed by the left subtree and then the right subtree	**E**	Binary Tree
3.6	An algorithm that sorts an array by dividing it into two partitions based on a pivot element	**F**	Array
3.7	A specialized tree data structure that allows for efficient priority management, where the highest or lowest priority element is always at the root	**G**	Destructor
3.8	Symbols used to represent operations, relationships, and properties in data structures and algorithms	**H**	Data Abstraction
3.9	A function that is automatically called when an object is destroyed, used for cleanup	**I**	In order
3.10	A class that extends or modifies the functionality of the base class, allowing for code reuse and the creation of a hierarchy of classes	**J**	Base Class
		K	Derived Class
		L	Heap
		M	Encapsulation

4. Each statement below has a blank space to fit one of the word(s) or phrase(s) in the list below. Enter your choice in the "OMR" answer sheet supplied with the question paper, following the instructions therein. (1x10)

A	Queue	**B**	Recursion	**C**	Priority Queue
D	Binary Search	**E**	Tree Traversal	**F**	Polymorphism
G	Destructor	**H**	Inheritance	**I**	Selection
J	Function Overloading	**K**	Complexity	**L**	Pointer
M	Constructor				

4.1 The principle of _____ allows a derived class to inherit the attributes and methods of its base class while also adding its unique features.

4.2 The _____ method is automatically called when an object goes out of scope and is used to release resources.

4.3 The _____ is a search algorithm that divides the array into two halves and recursively searches in the appropriate half, achieving O(log n) time complexity.

4.4 A _____ is an abstract data type that represents a collection of elements with a priority, allowing for efficient retrieval of the highest (or lowest) priority element.

4.5 In the context of algorithms, _____ refers to the theoretical measure of the time and space required for an algorithm to execute as a function of the length of the input.

4.6 A _____ sort is a comparison-based algorithm that builds the final sorted array one item at a time by repeatedly selecting the smallest (or largest) element from the unsorted portion.

4.7 _____ allows the same function name to be used for different functions, distinguished by the type or number of parameters.

4.8 A _____ is a variable that stores the memory address of another variable, allowing for dynamic memory management and efficient data manipulation.

4.9 _____ is a data structure that uses a function to map keys to their corresponding values, allowing for efficient data retrieval and insertion by converting the key into an index.

4.10 In object-oriented design, _____ allows for the creation of classes that share a common interface while implementing their functionality independently.

PART - TWO

(Answer any FOUR questions)

5. (a) Explain the concept of polymorphism in object-oriented programming. Provide examples of both compile-time polymorphism and run-time polymorphism.

(b) Discuss the principles of data encapsulation and data hiding. How do they contribute to the robustness of an application ?

(8+7)

6. (a) Write the importance of Depth First Search (DFS) and Breadth First Search (BFS) for graph traversal. Describe the algorithms for both methods.

(b) Define a graph and explain its key notations. What are the different types of graphs ? Give examples. What are the differences between adjacency matrix and adjacency list ?

(7+8)

7. (a) Design an algorithm to implement a circular queue using an array. Include operations for enqueueing and dequeuing elements.

(b) Given the following list of integers : 8, 3, 7, 4, 1, 5, sort them using the Quick Sort algorithm. Illustrate the partitioning process and the contents of the array at each step.

(8+7)

8. (a) Explain the concept of a stack overflow. What causes it, and how can it be prevented in a stack implementation ?

(b) Given the following sequence of operations on a stack : Push(5), Push(10), Pop(), Push(15), Push(20), Pop(), Pop(). What will be the final state of the stack after these operations ?

(c) Given the following postfix expression : 78+32+/10 - evaluate it using a stack. Show the contents of the stack and the output at each stage of the evaluation.

(3+5+7)

9. (a) What is hashing ? Discuss the essential properties of a good hash function. Explain four different hashing techniques with suitable examples.

(b) A hash table uses the hash function **H(key) = key** % 7 to insert keys into a table with 7 slots (indexed from 0 to 6). If linear probing is used to resolve collisions, insert the following keys into the table : 37, 38, 72, 48, 98, 11, 66. What will be the final location of key 11 after insertion ? How many total collisions occurred during the entire process ?

(6+9)

- o O o -

SPACE FOR ROUGH WORK

SPACE FOR ROUGH WORK