Ref No: EX/IEE/PC/B/IT/T/221/2025

B.E in INSTRUMENTATION AND ELECTRONICS ENGINEERING

2ND YEAR, 2nd SEMESTER EXAMINATION, 2025

DATA STRUCTURE, ALGORITHMS & OOPS

Time: 3 hours **Full Marks:** 100

Answer all parts of a question together in one place. Do not scatter the answers.

CO1 [20 MARKS]	**1. Multiple Choice Questions (1 mark × 4 = 4 marks)** i. Which of the following best describes an Abstract Data Type (ADT)? a) A data structure with fixed memory size b) A model that defines data and operations without implementation c) A programming language feature d) None of the above ii. The time complexity of linear search in the worst case is: a) O(log n) b) $O(n^2)$ c) O(n) d) O(1) iii. Which of the following is true about pseudo-code? a) It is executed directly by the computer b) It uses strict programming language syntax c) It provides a human-readable representation of algorithm logic d) It cannot include loops or conditionals iv. What does the notation O(1) signify in algorithm analysis? a) The algorithm runs in constant time regardless of input size b) The algorithm depends on the square of input size c) The algorithm cannot be analyzed d) The algorithm runs in logarithmic time **2. Fill in the blanks (1 mark × 4 = 4 marks)** i. A data structure that follows Last In First Out (LIFO) principle is called a _____. ii. Big-Oh notation represents the _____ bound of an algorithm's running time. iii. A _____ is a step-by-step procedure used to solve a specific problem. iv. The time complexity of binary search in the worst case is _____. **3. Short Answer Questions (2 marks × 4 = 8 marks)** i. Define a Data Structure. Mention any two linear data structures. ii. Write a simple pseudo-code to find the sum of elements in an array. iii. Differentiate between primitive data types and abstract data types with one example each. iv. Give an example to illustrate the difference between worst-case and average-case time complexity. **4.** Explain Big Oh, Omega, and Theta notations with suitable graphical representations. How do these notations help in algorithm analysis? **[4 marks]**

CO2 **[20 MARKS]**	**5. Justify the truth/falsity of each of the following statements. Provide suitable arguments in support of your answers along with suitable code snippets where/if required.** i) In row-major order, the elements of a 2D array are stored row-wise in memory. ii) A circular linked list allows traversal to the previous node without using extra pointers. iii) A queue implemented using an array can never be made circular. iv) The in-order traversal of a binary search tree always gives sorted data. v) Insertion in an AVL tree can require more than one rotation. **(1.5 marks × 5 = 7.5 marks)** **6.** Given two arrays that represent pre-order and post-order traversals of a **full binary tree**, construct the binary tree with required steps. **(2.5 marks)** **pre[]** = [1, 2, 4, 8, 9, 5, 3, 6, 7] **post[]** = [8, 9, 4, 5, 2, 6, 7, 3, 1] **7.** Write pseudocodes for insertion and deletion in a **Binary Search Tree. (5 marks)** **8.** Write pseudocodes for insertion and deletion in a **doubly linked list. (5 marks)**
CO3 **[20 MARKS]**	**9. Fill in the blanks (1 mark × 4 = 4 marks)** i. _____ sort works by repeatedly swapping adjacent elements if they are in the wrong order. ii. In a priority queue, the underlying data structure used in Heap Sort is a _____. iii. The best case time complexity of Insertion Sort is _____. iv. The number of comparisons in Selection Sort for an array of n elements is _____. **10.** Explain with pseudocode a search algorithm that can improve upon Binary Search in certain scenarios. What happens **if all the elements in the array are equal? (6 marks)** **11.** Explain with pseudocode a sorting algorithm that requires extra space and has **O (n log n)** time complexity in all cases (best, worst, and average). **(4 marks)** **12.** Consider an array containing data elements 8, 2, 4, 6, 9, 7, 10, 1, 5, 13. Create the max heap from the array and hence show the array contents after 1st iteration of Heap sort. **(3 marks)** **13.** Consider an array containing data elements 5, 4, 15, 20, 6, 2, 8, 9, 10. Now set **5** as the pivot element and show where it will be placed after 1st iteration of Quick Sort. Hence show the contents of the array. **(3 marks)**

CO4

[20 MARKS]

14. Discuss the four basic pillars of Object-Oriented Programming. **(2 marks)**

15. Differentiate Function Overloading and Function Overriding with code snippets. **(3 marks)**

16. What are the differences between private, protected, and public? How to access a private member outside of a class? **(3 + 2 = 5 marks)**

17. Consider the following code snippet with the definition of class *Complex* with two member variables *real* and *imag* representing the real part and imaginary part of a complex number respectively.

Define all the constructors and methods within or outside the class definition and show how they are invoked from *main()*.

```
class Complex {
        float real, imag;
        public:
        Complex (float real, float imag);
        friend Complex sum (Complex, Complex);
        void subtract (Complex);
        Complex multiply (Complex);
        void show();
```

(6 marks)

18. Assume two classes *ABC* and *XYZ* have a private data member each of their own. Initialize their values with suitable constructors. Show (with proper code snippet) how the values of those two member variables can be swapped with the help of only one function. **(4 marks)**

CO5

[20 MARKS]

19. Distinguish among each of the following triples. Use suitable code snippet in support of your answer.

i) Private, protected, and public mode of inheritance

ii) Abstract class, derived class, and friend class

(3+3 = 6 marks)

20. Why should a base class destructor be declared as virtual in C++, and what could go wrong if it is not? Illustrate with an example. **(3 marks)**

23. Explain the significance of static data members in C++. Give the output of the below C++ code.

```
#include <bits/stdc++.h>
using namespace std;
void f() {
  static int count = 0;
  count++;
  cout << count << " ";
}

int main() {
  for (int i = 0; i < 5; i++) {
    f();
  }
  return 0;
}
```

(3 marks)

	21. What is **diamond Problem in C++** and how will you over come this? Discuss. **(3 marks)** **22.** What is operator overloading in C++. Write a program to add two complex numbers by overloading binary (+) operator. **(5 marks)**

Course outcomes:

CO1: Understand data structures their advantages, drawbacks its types and analyze algorithms

CO2: Explain, apply and analyze different types of linear and non-linear data structures

CO3: Explain and illustrate different techniques of searching and sorting and differentiate them in terms of performance

CO4: Explain, illustrate and recognize the basic features of classes, objects and encapsulation mechanisms.

CO5: Illustrate the extended features of OOPs (Inheritance, Polymorphism, Operator overloading) and apply them to solve practical problems.