Q. Code:170821

Reg. No. | | | | | | | | | | | | | |

B.E./ B.TECH. DEGREE EXAMINATIONS, MAY 2024

Third Semester

EC22301 – OBJECT ORIENTED PROGRAMMING AND DATA STRUCTURES

(Electronics and Communication Engineering)

(Regulation 2022)

TIME: 3 HOURS **MAX. MARKS: 100**

COURSE OUTCOMES	STATEMENT	RBT LEVEL
CO 1	Read, Write and Execute simple C++ programs.	**2**
CO 2	Choose appropriate object-oriented programming principles and propose novel solution to solve computational problem.	**3**
CO 3	Understand the core data structures like lists, stack and queue using C++.	**2**
CO 4	Design and implement non-linear data structures using C++ programs.	**3**
CO 5	Discuss different sorting and searching techniques to organizing the large amount of data.	**3**

PART- A (20 x 2 = 40 Marks)

(Answer all Questions)

		CO	RBT LEVEL
1.	How a ‘class’ differ from a ‘struct’ in C++?	**1**	**2**
2.	How is the 'this' pointer used in member functions?	**1**	**2**
3.	What is a friend function in C++?	**1**	**2**
4.	Mention the use of ‘new’ and ‘delete’ operators.	**1**	**2**
5.	List the operators that cannot be overloaded in C++.	**2**	**2**
6.	What are virtual functions in C++?	**2**	**2**
7.	List the different types of inheritance in C++.	**2**	**2**
8.	Define Inheritance and Polymorphism.	**2**	**2**
9.	Differentiate virtual function and pure virtual function.	**2**	**2**
10.	What is an Abstract Data Type (ADT) and why is it important in programming?	**3**	**2**
11.	What are the advantages of using linked lists over arrays for implementing lists?	**3**	**2**
12.	Classify the different types of queues.	**3**	**2**

13.	Give the difference between singly linked lists and doubly linked lists.	**3**	**2**
14.	Describe how the following "infix" expression is evaluated with the help of stack: 3 * (6 + 4) - 12 / 4.	**3**	**2**
15.	Suggest an expression tree for the expression. A*(B-C)-((D-E*F)*G).	**4**	**2**
16.	Provide examples of real-world applications where trees are used.	**4**	**3**
17.	Evaluate the postfix expression 5 8 6 + 4 * 2 + *	**4**	**2**
18.	List the applications of Searching.	**5**	**2**
19.	Distinguish between internal and external sorting.	**5**	**2**
20.	Recommend a simple algorithm for a binary search.	**5**	**2**

PART- B (5 x 10 = 50 Marks)

		Marks	CO	RBT LEVEL
21. (a)	Write a member function to perform matrix addition and subtraction by overloading '+' and '-' respectively.	**(10)**	**1**	**2**
	(OR)			
(b)	Explain constructor and destructor, detailing the various types and providing examples to demonstrate their usage in object-oriented programming.	**(10)**	**1**	**2**
22. (a)	Explain the concept of operator overloading and design a C++ program demonstrating the overloading of the arithmetic operator '-' for user-defined data types.	**(10)**	**2**	**3**
	(OR)			
(b)	Explain the advantages and disadvantages of using protected members compared to public or private members in various scenarios. Additionally, discuss strategies for effective member visibility and access control in object-oriented programming.	**(10)**	**2**	**3**
23. (a)	Examine the process of the Last-In-First-Out (LIFO) principle associated with stack.	**(10)**	**3**	**2**
	(OR)			
(b)	Design an algorithm for performing polynomial addition using linked lists.	**(10)**	**3**	**2**

24. (a)	Explain how you would insert a new node into a binary search tree (BST). Provide pseudocode for the insertion algorithm and discuss its time complexity.	**(10)**	**4**	**3**
	(OR)			
(b)	List the ways to represent a graph and discuss on Dijkstra's shortest path algorithm.	**(10)**	**4**	**3**
25. (a)	Recommend an algorithm for Merge Sort and explain with an example.	**(10)**	**5**	**3**
	(OR)			
(b)	Suggest a quick sort algorithm and explain with suitable example. Give its worst case, average case and best-case time complexities.	**(10)**	**5**	**3**

PART- C (1 x 10 = 10 Marks)

(Q.No.26 is compulsory)

		Marks	CO	RBT LEVEL
26.	Construct a C++ code to implement singly linked list, incorporating the insertion of data elements: 50, 100, 150, 200, 250, 300. Also perform the following operations: Delete data in the beginning, Delete data in the end, search for a particular data element.	**(10)**	**3**	**5**

****