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

Question Paper Code: 54824

B.E. / B.Tech. DEGREE EXAMINATION, NOV 2019

Fourth Semester

Electrical and Electronics Engineering

(Common to Electronics and Instrumentation Engineering and

Instrumentation and Control Engineering)

01UIT424 - DATA STRUCTURES AND ALGORITHMS

(Regulation 2013)

Duration: Three hours   Maximum: 100 Marks

Answer ALL Questions.

PART A - (10 x 2 = 20 Marks)

1. List the characteristics of Constructor

2. What is friend function?

3. What are objects and how they are created?

4. Name the various types of multiple inheritance.

5. Define ADT.

6. Define algorithm.

7. Define connected components of a graph. Write its uses.

8. Differentiate binary tree and binary search tree.

9. Arrange the values using bubble sort 4, 2, 1, 2, 0, 3, 2, 1, 4, 0, 2, 3, 0 .

10. Define divide and conquer algorithm?

PART - B (5 x 16 = 80 Marks)

11. (a) Explain overloading concept with unary and binary operators with examples. (16)

Or

(b) What is meant by overloading? How is operator overloading works? Write a program to add 2 complex numbers using operator overloading. What are the operators that cannot be overloaded. (16)

12. (a) Explain the following terms with respect to OOPS and give suitable examples.
   (i) Polymorphism
   (ii) Exception handling. (16)

Or

(b) What is virtual function? Explain with an example how late binding is achieved using virtual function. (16)

13. (a) (i) Write the algorithms for the operations of linked queues. (8)

(ii) Explain the representation of priority queue. (8)

Or

(b) Write a C++ code to perform addition of two polynomials using link list form of queue. (16)

14. (a) Define NP complete problem. Where it is applied? Discuss one application with example. (16)

Or

(b) Write C++ code for the implementation of different types of tree traversals. State few tree applications. (16)

15. (a) Compare merge sort and insertion sort algorithms with examples. (16)

Or

(b) Explain how the travelling salesman problem can be solved using greedy algorithm. (16)

54824