

85

Reg. No. :

Question Paper Code : 52906

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2019.

Third Semester

Electronics and Communication Engineering

EC 6301 — OBJECT ORIENTED PROGRAMMING AND DATA STRUCTURES

(Common to: Biomedical Engineering/Medical Electronics/Robotics and Automation Engineering)

(Regulation 2013)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is meant by encapsulation?
2. List the C++ operators that cannot be overloaded.
3. How can member functions of the derived class access members of the base class?
4. Distinguish between static and dynamic binding.
5. What is a list ADT? How is it represented using linked lists?
6. Convert the given infix expression $10 * 3 + 15 / (8 + 2) * 2 - 4$ to its postfix form.
7. Define Trees.
8. Give two applications for which breadth first search is well suited.
9. What is meant by comparison based sorting?
10. How many times the iterative loops in the source code of insertion sort executes, if the input is {1 2 4 7 13 15 18}?

PART B — (5 × 13 = 65 marks)

11. (a) (i) Discuss the importance of static class members using an example. (6)
- (ii) Define function overloading. Write a C++ program using function overloading to compute the area of a triangle, rectangle, square and a circle. (7)

Or

- (b) (i) Write a note on the container classes. (6)
- (ii) Write a C++ program to add two complex numbers by overloading binary plus operator. (7)
12. (a) (i) Create a class called STUDENT with data members such as roll number, name and branch. Create a class called Exam with data members such as roll number and a six subject marks. Derive a Result class from Student and Exam and have its own data members such as total mark and result. Write a C++ program to get the marks of the subjects and display the total. (9)
- (ii) What is meant by overriding? Explain with an example. (4)

Or

- (b) (i) Define the class base as given below
- ```
class base {
 public: virtual void iam () { cout<< "base\n";}
};
```
- Derive two classes from base and for each, define iam () to write out the name of the class. Create objects of these classes and invoke iam() for them. Assign pointers to objects of the derived classes to base\* pointers and call iam() through those pointers.
- Display the actions carried out using member functions. (9)
- (ii) Explain concrete classes with an example. (4)

13. (a) Write C++ routines for the following operations in an array implementation of list ADT.
- (i) insert an element at the beginning. (3)
- (ii) insert an element at the end of the list. (3)
- (iii) delete an element Y after an element X. (3)
- (iv) to test if the given element X is present in the list or not. (4)

Or

- (b) (i) What are the operations that can be performed on a stack? Write routines to perform them using an array implementation of stack ADT. (7)
- (ii) Write routines to perform enqueue and dequeue of elements from the linked list implementation of Queue ADT. (6)

14. (a) Explain the binary tree traversals with their recursive routines. Trace the routines using suitable examples. (13)

Or

- (b) (i) Explain the ways to represent a graph mentioning the advantages of each. (7)
- (ii) Compare and contrast breadth first and depth first traversals. (6)
15. (a) Explain merge sort algorithm and use it to sort the integers {2, 6, 13, 18, 25, 27, 30, 35, 50, 100}. Show the intermediate results after each iteration. (13)

Or

- (b) (i) Write an iterative binary search algorithm and explain with an example. Show the intermediate results after each iteration. (9)
- (ii) Compare and contrast linear search and binary search algorithms. (4)

PART C — (1 × 15 = 15 marks)

16. (a) Create a base class called SHAPE. Use this to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base SHAPE. Add to the base class, a member function get\_data() to initialize base class data members and another member function display\_area() to compute and display the area of figures. Make display\_area() as a virtual function and redefine this function in the derived classes to suit their requirement. Design a C++ program to get the dimensions of triangle and rectangle and display the area.

Or

- (b) Design an algorithm to check if the parentheses are balanced in an arithmetic expression using stack ADT and trace it with an example.