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

Question Paper Code : 20407

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2018.

Third/Fourth Semester

Electronics and Communication Engineering

EC 6301 — OBJECT ORIENTED PROGRAMMING AND DATA STRUCTURES

(Common to Biomedical Engineering, Medical Electronics, Robotics and Automation Engineering)

(Regulations 2013)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is a class in object oriented programming? Illustrate with an example.

2. What is a friend function?

3. What is overriding?

4. Why there is need for operator overloading?

5. What is ADT?

6. Write short notes on queue.

7. What is a tree?

8. How a graph is represented?

9. What is meant by sorting?

10. What is space complexity?

PART B — (5 × 13 = 65 marks)

11. (a) Describe the major components of object oriented programming with illustrations. (13)

Or

(b) What is the purpose of constructor and destructor? Explain with suitable example the different types of constructors in C++. (13)

12. (a) What is inheritance? Discuss in detail about the various types of inheritances in C++ with suitable examples. (13)

Or

(b) What is virtual function? Explain with an example how late binding is achieved using virtual function. (13)

13. (a) Write a set of routines for implementing two stacks within a single array. (13)

Or

(b) Write a set of routines for implementing queue using linked lists. (13)

14. (a) Discuss the different methods for traversing a binary tree with algorithm. (13)

Or

(b) Illustrate the Depth First Search algorithm with a graph and explain. (13)

15. (a) Discuss the quick sort algorithm and apply the same for the following numbers 90, 77, 63, 99, 54, 88, 66. (13)

Or

(b) Explain in detail about binary search algorithm with an example. (13)

PART C — (1 × 15 = 15 marks)

16. (a) Develop program, which receives objects as arguments, and return objects as return values. Illustrate the above using complex number objects. Write a main( ) to test the above. (15)

Or

(b) Define a class String that could work as a user defined data type. Include constructors that will create un-initialized string and initialize an object with string constant at the time of creation of an object of string class. Include a function that adds two strings to make a third string. Write a main( ) to test your class. (15)