FACULTY OF ENGINEERING

BE I-Year (Backlog) Examination, November / December 2018

Subject: Programming in C & C++

Time: 3 Hours **Max. Marks: 75**

Note: Answer All Questions From Part-A & Any Five Questions From Part-B.

PART-A (25 Marks)

1. Define Compiler, Loader. (2)
2. Write an algorithm to find sum of 'n' natural numbers. (3)
3. Differentiate i) Global variable and Local variable ii) actual arguments and formal parameters. (3)
4. List out 'C' Preprocessor statements. (2)
5. What is a self referential structure? Give an example. (3)
6. What are enumerated data types? Give an example. (3)
7. What do you mean by static data members and static member functions? (2)
8. What is unary scope resolution operator? (2)
9. Define polymorphism. Give different types of polymorphism. (3)
10. What are I/O streams? (2)

PART-B (50 Marks)

11. (a) Give differences between while and do-while loops. Write a program to find the reverse of a given number . (6)
    (b) Draw a flowchart to find the roots of a quadratic equation. (4)
12. (a) Write a program in 'C' for Binary Search. (5)
    (b) Write a program in 'C' to find the maximum and minimum from the given array using pointers. (5)
13. (a) Write a 'C' program to add two complex numbers using structures. (5)
    (b) Write a program to read the contents of a text file and display them. (5)
14. (a) What is dynamic memory allocation? What operators are used in c++ to allocate memory dynamically. Write a program using dynamic memory allocation in c++. (6)
    (b) What are default arguments? (4)
15 (a) Define inheritance. Explain types of inheritance with examples. (5)
    (b) Define function templates with syntax. Write a program to find largest of 'n' numbers using function templates. (5)

16. (a) Write a program to add two numbers using command line arguments. (5)
    (b) Explain with an example how exception handling is done in c++. (5)

17. (a) Differentiate different parameter passing techniques in c++ with suitable examples. (5)
    (b) Write a program for overloading increment and decrement operators. (5)

******