PAPER-III

C++WITH INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

SECTION-A

1 (a) Differentiate between increment and decrement operator in C++ with the help of examples. 5

(b) Write short notes in C++ on the following :

(i) Enumerated data type and its advantage.

(ii) Type casting.

(iii) Redufining data types with typedef. 15'

2. (a) How do struct and union data type differ in C++ ? 7

(b) Distinguish between switch and if-else state,emt om C++. 6

(c) Differentiate between break and continue statement by writing any program in C++. 7

3. (a) Explain the hierarchy of operators in C++.

(b) Explain the various loop control structures in C++ along with examples. 14

4. (a) Write a program in C++ to implement bubble sorting. 10

(b) Write program in C++ to find transpose of a matrix. 10

5. (a) How the array is declared and initialized in C++ ? 6

(b) Explain the various library functions for manipulating strings in C++. 7

(c) Write short note on the various C++ streams. 7

SECTION-B

6. (a) Differentiate between class and object. Give examples in c++. 8

(b) Differentiate between constructors and destructors in C++. Write any program in C++. 8

7. (a) Explain the use of Scope Resolution Operator in C++ with the help of examples. 8

(b) Write short note on Static Data Member. 5

(c) Define this pointer. Explain its use in C++ with the help of examples. 7

8. Define Operator overloading. Write are the rules of operator overloading ? Explain overloading of Binary operators by writing program in C++. 20

9. (a) Explain the private, public and protected access specifiers in C++. How do they differ from each other ? Differentiate between these by writing any program in C++. 10

(b) Explain the concept of Virtual Base class and Abstract class in C++ with the help of examples. 10

10. (a) Explain the concept of Friend Function in C++. What are its characteristics ? 5

(b) Differentiate the following with the help of examples in C++ :

(i) Overridign vs. Overloading.

(ii) Multiple vs. Multilevel inheritance.

(iii) Hierarchical vs. Hybrid inheritance. 15

❐❐❐