INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

Time : 3 Hours *Maximum Marks : 90*

Note : Question no. 1 is compulsory. Attempt Five questions in all, selecting ONE question from each Unit.

1. i) Explain the difference between structure and class in terms of Access Modifier.
   ii) What is difference between pointers to constant and constant to pointers?
   iii) Compare constructor conversion and operator conversion.
   iv) Can you use this pointer to a friend function? Comment on it.
   v) What are the benefits of inline functions over macros?
   vi) What is the difference between the statements ;
   cin >> ch; & ch = cin.get ( );

UNIT-I

2. (a) What are class and object in C++? How does a class implement data hiding and encapsulation?
   b) Explain concept of state member function with example program in C++.
3. (a) List the features of OOPS. Discuss them with examples.
   (b) Illustrate with examples, Nested and Local class.

UNIT-II

4. Discuss the need of constructor in C++. How is it different from normal member function? Explain its types with examples.

5. (a) Justify the need of object cleanup and initialization facility for creating live objects.
   (b) Illustrate with an example, how the endl, ws and setw, setfill manipulator works?

Unit-III

6. (a) What is a friend function? What are the merits and demerits of using friend function?
   (b) Write a C++ program to read two strings S1 and S2 perform the task S3 = S1+S2 and print the string S3. Consider overloading of the '+' operator using friend function.
7. (a) What do you understand by dynamic memory allocation/deallocation ? Write a program for illustrating dynamic memory management.
   (b) What is this pointer? What is your reaction to the statement : delete this? Write a program demonstrating the use of this pointer.

UNIV-IV

8. (a) What are the limitations of overloading unary increment? decrement operator? How are they overcome?
9. a) What are the limitations of in-line function?
   b) What is static Polymorphism? Discuss its merits and demerits.