KNT/KW/16/5303

Bachelor of Science (B.Sc.I.T.) Semester—II (C.B.S.) Examination

OBJECT ORIENTED PROGRAMMING USING "C++"

Paper—II

Time : Three Hours] [Maximum Marks : 50

N.B. :— (1) **All** questions are compulsory and carry equal marks.

(2) Draw neat and labeled diagram wherever necessary.

EITHER

1. (a) List different features of object oriented programming and explain any two. 5

(b) Write a program to implement class "Account" having following members :

Data member – Name of Account holder,
Account Number,
Balance Amount

Member function – get data ( )
deposit ( )
withdrawal ( )
put data ( ). 5

OR

(c) Explain the following access specifiers with example :

(i) Private

(ii) Public. 5

(d) Explain static data member giving suitable program. 5

EITHER

2. (a) What is operator overloading ? Explain overloading of binary operator giving suitable program. 5

(b) Explain parameterised constructor giving suitable program. 5

OR

(c) List rules for operator overloading. List the operators that cannot be overloaded. 5

(d) Write a program to demonstrate use of constructor and destructor. 5

EITHER

3. (a) Explain multiple inheritance giving suitable program. 5

(b) Explain the order of execution of constructors and destructors in derived classes giving suitable program in 'C++'. 5

OR

(c) Explain new and delete operator giving suitable program. 5

(d) How "this pointer" is used in programming ? Explain giving suitable program. 5

EITHER

4. (a) What do you mean by virtual function ? List rules for creating virtual functions. 5

(b) Explain Exception handling model giving suitable program. Write rules for handling exception successfully. 5

OR

(c) Why is there a need for pure virtual function ? Explain with suitable example. 5

(d) Write short note on abstract classes. 5

5. (a) Differentiate between :

(i) Defining member function inside the class and,

(ii) Defining member function outside the class. 2½

(b) Explain copy constructor giving a suitable program. 2½

(c) What is Inheritance ? Explain its advantages. 2½

(d) Explain handling of uncaught exceptions. 2½