

***B.Tech. Degree VI Semester (Supplementary) Examination in  
Computer Science and Engineering, December 2002***

**CS 602 OBJECT ORIENTED PROGRAMMING AND DESIGN  
(1998 Admissions)**

Time: 3 Hours

Maximum Marks: 100

- I. (a) What are the two main programming paradigms? How do they differ from each other? (10)  
(b) Explain the meaning of Abstraction with an example. (10)
- OR**
- II. Define Object Oriented Programming. How is generic programming achieved in C++? (20)
- III. (a) What are the various design methodologies used in programming? Which type of Methodology is objected oriented programming? (10)  
(b) Give the various design issues to be considered during the development of Object Oriented System. (10)
- OR**
- IV. (a) Describe the steps used in the development cycle of an Object Oriented System. (10)  
(b) What are associations? How will you represent the associations using class diagram? (10)
- V. (a) Explain how reusability is achieved in Object Oriented Programming. (10)  
(b) Give specific reasons choosing an Object Oriented Language for developing an application. (10)
- OR**
- VI. (a) Give the merits of using an Object Oriented Programming. (10)  
(b) What are interfaces? Explain the use of interfaces in Object Oriented Programming. (10)
- VII. (a) How is pointer and array related to each other? How is array allocation and deallocation done in C++? (10)  
(b) What are constructors and destructors? What is overloading? How do constructors help in type conversion? (10)
- OR**
- VIII. (a) What are structures? How do they differ from a class? Define an array? Distinguish between stater memory allocation and dynamic memory allocation. (10)  
(b) What are different types of inheritance? Define a virtual base class. (10)
- IX. (a) What is operator overloading? How do overloading help in type conversion? (10)  
(b) Design a class Matrix. Define operator \* to multiply two matrices. Do necessary initialisations of matrixes also. (10)
- OR**
- X. (a) Define  

```

Class base {
    Public : virtual void iam( ){iout < c"base\n";}
};

```

Derive two classes from base and for each define iam( ) to write out the name of the class. Create objects of these classes and call iam( ) for them. Assign pointers to objects of the derived clauses to base \* pointers and call iam( ) through those pointers. (10)  
(b) What are templates and how do they help in generic programming? (10)

