(Pages : 2) **2019**

Reg. No. : .....

Name : .....

# Fourth Semester B.Tech. Degree Examination, May 2014
## (2008 Scheme)
## 08.404 : OBJECT ORIENTED TECHNIQUES (RF)

Time: 3 Hours Max. Marks: 100

### PART – A

Answer **all** questions :

1. What is data hiding ? How is it implemented in C++ ?
2. What is function overloading ?
3. Define the term sensability.
4. What are constructors and destructors ?
5. What is an abstract class ?
6. How is dynamic memory allocation is done in C++ ?
7. What are function templates ?
8. What are nested classes ?
9. What do you mean by generic functions ?
10. What is multi level inheritance ?



### PART – B

Answer **any one** from **each** module :

#### Module – 1

11. a) With suitable example explain static members and protected members. **(10)**

b) Write a program to overload a function named volume () to find the volume of a cube, sphere and a rectangular box. **(10)**

OR

12. a) Explain in detail the features of object oriented programming. **(12)**

b) With suitable example explain functions with default arguments and inline functions. **(8 )**

## Module – 2

13. a) Describe the following with example :

i) Copy constructors.

ii) This pointers. **(10)**

b) Illustrate with example how constructors are overloaded ? **(10)**

OR

14. a) Explain the concept of classes and objects with a suitable example . **(10)**

b) Explain 'const' data members and member functions with example. **(10)**

## Module – 3

15. a) How do you overload unary and binary operators in C++ ? Explain. **(10)**

b) Explain virtual functions with an example. **(10)**

OR

16. a) A company has many departments. An employee of a company is associated with any one of the departments. Each department has a manager. The manager is one of the employees. Create necessary inheritance hierarchy and polymorphic functions for printing the details of employees. Classes should have constructors and destructors. Write a separate function to list the details of employees of a given department. **(12)**

b) Define class template and function template with suitable examples. **(8)**