Total No. of Questions—8]   [Total No. of Printed Pages—3

Seat No.	

[5152]-167

S.E. (Comp.) (II Sem.) EXAMINATION, 2017
OBJECT ORIENTED AND MULTICORE PROGRAMMING
(2012 PATTERN)

Time : Two Hours   Maximum Marks : 50

N.B. :— (i) Answer Q. No. **1** or Q. No. **2**, Q. No. **3** or Q. No. **4** or Q. No. **5** or Q. No. **6**, Q. No. **7** or Q. No. **8**.

(ii) Neat diagrams must be drawn wherever necessary.

(iii) Figures to the right indicate full marks.

(iv) Assume suitable data, if necessary.

1. (a) Define the terms :   [8]

    (i) Member Access Control

    (ii) RTTI

    (iii) Constructor and Destructor

    (iv) Static member function.

  (b) Explain the virtual base class with suitable example.   [4]

Or

2. (a) Explain the following terms :   [9]

    (i) Generic programming

    (ii) New and delete keyword

    (iii) Pure virtual function.

  (b) Explain the concept of copy constructor in detail.   [3]

P.T.O.

3. (a) What is a scope resolution operator ? Explain with suitable example. [4]

(b) Explain the free store operators and memory management operators. [4]

(c) What do you mean by dynamic initialization of variables ? Explain with example. [4]

Or

4. (a) Explain with suitable example, call by value and call by reference. [8]

(b) How to handle multiple exceptions occurred in a program ? [4]

5. (a) Explain difference between function overloading and function template. [6]

(b) Explain STL in detail with its components. [6]

Or

6. (a) Explain try-catch-throw-rethrow in detail with example. [6]

(b) Distinguish between error and exception [3]

(c) Explain any *two* unformatted I/O functions. [3]

7. (a) Explain threads in terms of creating, compiling and linking. [6]

(*b*)   Explain concept of setting thread and scheduling and priorities.   [4]

(*c*)   What are thread interface classes ? Give example.   [4]

Or

8.   (*a*)   Explain with suitable diagram, decomposition of tasks in multiprocessing environment.   [8]

(*b*)   Explain sequential and concurrency models.   [3]

(*c*)   Explain message queue and semaphore.   [3]