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

| Seat No. | |
|---|---|

# [4857]-1077

## S.E. (Computer) (Second Semester) EXAMINATION, 2015
## OBJECT ORIENTED AND MULTICORE PROGRAMMING
## (2012 PATTERN)

**Time : Two Hours** **Maximum Marks : 50**

***N.B.*** **:—** (*i*) Neat diagrams must be drawn wherever necessary.

(*ii*) Figures to the right indicate full marks.

(*iii*) Use of calculator is allowed.

(*iv*) Assume suitable data, if necessary.

**1.** (*a*) Explain the following terms with example : [8]

(*i*) Copy constructor

(*ii*) Manipulators

(*iii*) Static data members

(*iv*) This pointer.

(*b*) With suitable examples, demonstrate the benefits of Operator Overloading. [4]

*Or*

**2.** (*a*) Explain why and when do we use protected Instead of Private ? With suitable examples, explain different types of inheritance. [8]

(*b*) Explain run-time polymorphism with a suitable example. [4]

P.T.O.

**3.** (*a*) What are the core services of an OS ? Describe process state transition diagram. [8]

(*b*) What do you mean by Unformatted Console I/O Functions ? What is the use of the following Unformatted Console I/O Functions ? [4]

(*i*) Getch( )

(*ii*) Putchar( )

(*iii*) Get( )

(*iv*) Put( )

*Or*

**4.** (*a*) What is C++ template ? Describe type template parameters and non-type template parameters. [8]

(*b*) What is Process and Thread ? How mapping of multiple threads on multiple cores takes place ? [4]

**5.** (*a*) What is deadlock ? What are different conditions that must be true for deadlock to happen ? [9]

(*b*) Differentiate between preemptive scheduling and time slicing. [4]

*Or*

**6.** (*a*) Explain the different attributes of the pthread_attr_t object which can be modified by the creator of the thread. [9]

(*b*) Write a short note on thread interface classes. [4]

**7.** (*a*) With suitable terminologies explain the following terminologies : [8]

(*i*) Task synchronization

(*ii*) Critical section

(*iii*) Semaphore

(*iv*) Message Passing.

(*b*) What do you mean by Thread safety ? With reference to Thread safety, what do you mean by conditionally safe and Not thread safe code ? [5]

*Or*

**8.** (*a*) Explain Interprocess Communication (IPC) and explain any *two* ways of implementing IPC. [9]

(*b*) Explain the use of read-write-locks to prevent race conditions and deadlocks. [4]