CO - VI (Rev)

OOSE

19/11/2014

QP Code : 15022

(3 Hours) **[Total Marks : 100**

N.B : (1) Question **No. 1** is **compulsory.**
(2) Attempt any **four** questions out of the remaining **six** questions.
(3) Assume **data if required**, and state clearly.

1. MSRTC owns a number of busses. Each bus is allocated to a particular route, although some routes may have several buses. Each route passes through a number of towns. One or more drivers are allocated to each stage of a route, which corresponds to a journey through some or all of the towns on a route. Some of the towns have a garage where busses are kept and each of the busses are identified by the registration number and can carry different numbers of passengers, since the vehicles vary in size and can be single or double-decked. Each route is identified by a route number and information is available on the average number of passengers carried per day for each route. Drivers have an employee number, name, address, and sometimes a telephone number. Develop Class and Use Case Diagram. **20**

2. (a) Explain the following with suitable examples Composition, Association, Generalization, Aggregation. **10**
   (b) Compare Forward Engineering, Reverse Engineering and Re-Engineering. **10**

3. (a) Develop an activity diagram for any one of the use cases of Airline reservation system. **10**
   (b) Explain COCOMO model used for software estimation. **10**

4. (a) Explain how to map different types of association and generalization relationship to code. **10**
   (b) Explain the object oriented testing strategies. **10**

5. (a) Explain coupling & cohesion. Explain different types of coupling and cohesion. **10**
   (b) What are the different types of maintenance and also explain steps for creating a maintenance log? **10**

6. (a) What do you mean by requirements? Explain Functional and Non Functional Requirements in detail. **10**
   (b) Explain Open Source Software Life Cycle. **10**

7. Write short notes on any **two** :— **20**
   (a) Risk Management
   (b) Software Architectural styles
   (c) Design Pattern.
