Reg. No. : | | | | | | | | | | | | |

Question Paper Code : X 60389

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2020
Sixth Semester
Computer Science and Engineering
CS 2353/CS 63/10144 CS 603 – OBJECT ORIENTED ANALYSIS AND DESIGN
(Common to Information Technology)
(Regulations 2008/2010)
(Also common to PTCS 2353 – Object Oriented Analysis and Design for B.E. (Part-Time) Fifth Semester – Computer Science and Engineering – Regulations 2009)

Time : Three Hours Maximum : 100 Marks

Answer ALL questions

PART – A **(10×2=20 Marks)**

1. Why do we need object oriented systems development ?
2. List out the steps for finding use cases.
3. What is a Domain Model ?
4. Define Aggregation and Composition.
5. List the relationship used in class diagram.
6. What is the use of system sequence diagram ?
7. What is Design Pattern ?
8. What is meant by Low Coupling ?
9. What is the use of component diagram ?
10. Give the meaning of Event, State and Transition.

PART – B **(5×16=80 Marks)**

11. a) What do you mean by Unified Process in OOAD ? Explain the phases with suitable diagrams. **(16)**

(OR)

b) By considering your own application, perform the Object Oriented System Development and give the use case model for the same (use include, extend and generalization). **(16)**

12. a) Describe the strategies used to identify conceptual classes. Describe the steps to create a domain model used for representing conceptual classes. **(16)**

(OR)

b) Write briefly about elaboration and discuss the differences between Elaboration and Inception with examples. **(16)**

13. a) Briefly explain about UML Sequence diagrams. **(16)**

(OR)

b) Describe the UML notation for class diagram with an example. Explain the concept of association and inheritance. **(16)**

14. a) Explain about GRASP Patterns. **(16)**

(OR)

b) Write short notes on adapter, singleton, factory and observer patterns. **(16)**

15. a) Explain UML State Machine Diagrams and Modeling. **(16)**

(OR)

b) Write short notes about the following :

i) Operation contacts. **(6)**

ii) Implementation model (mapping design to code). **(10)**

———————