Question Paper

Exam Date & Time: 30-Dec-2019 (09:30 AM - 12:30 PM)

MANIPAL ACADEMY OF HIGHER EDUCATION

INTERNATIONAL CENTRE FOR APPLIED SCIENCES
END-SEMESTER THEORY EXAMINATION- NOVEMBER 2019
III SEMESTER B.Sc. (Applied Sciences) - in Engg.
SOFTWARE DESIGN USING OBJECT ORIENTED PARADIGM [ICS 233 - S2]

Marks: 100 **Duration: 180 mins.**

Answer 5 out of 8 questions.

1) A) Along with a program, compare and explain why the object-oriented approach is better compared to the conventional approach of programming (8)

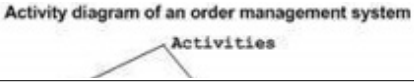
B) What are the essential notations used in the use case diagram? Write them through the diagram. List the guidelines to be followed for writing the use case diagram (8)

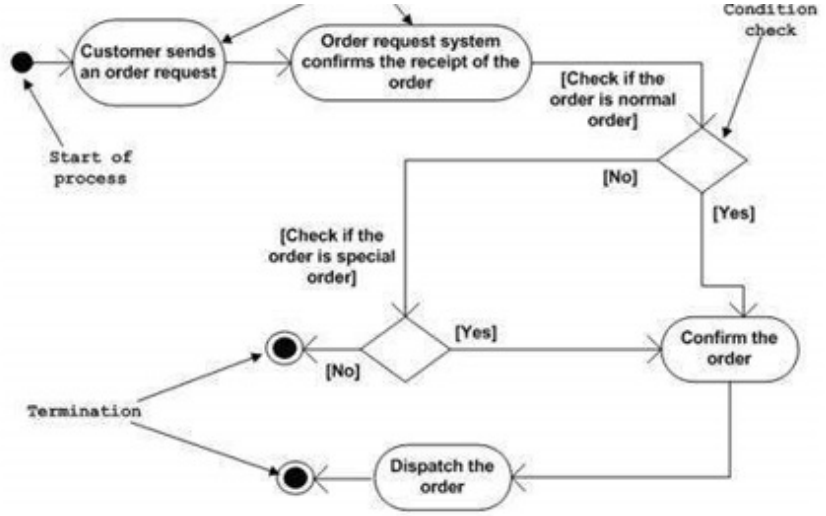
C) What is the meaning of following symbols and syntaxes in UML diagrams (4)
+, -, #, 1 to 1..*, [check = "true"], *[i = 1..n]

Write the UML syntax of method declaration in a class diagram for the following sentence "The method ReadBook is a protected method which takes book name, author name and book id as the parameters which are of type string, string, and integer respectively and the return type is a double"

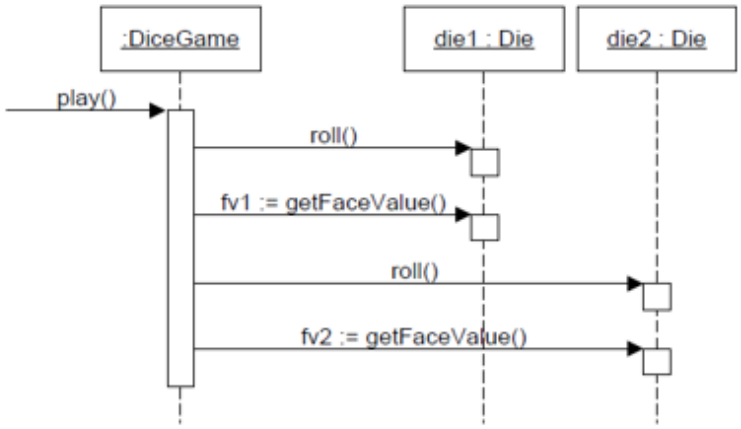
2) A) List at least four differences between the class diagram and the object diagram. Write an example to show these differences (8)

B) For the following activity diagram, write a simple program with the necessary classes in the programming language of your choice. Assume the suitable class member variables and methods (8)





C) Write the program for the following sequence diagram (4)



3) Explain the include, extend and generalization relationship w.r.t ProcessSale use case of NextGen POS system with a neat diagram (8)

A)

B) List the key points in identifying the conceptual classes (4)

C) Why do we need a description class in any POS system or e-commerce application? What problem does it solve? Explain this with a neat diagram (8)

4) Write the interaction diagram that shows the inter package interaction scenarios in case of a process sale use case. In this diagram, clearly specify the objects belonging to different packages with proper notation (hint: this is a system sequence diagram). (10)

A)

B) What is a design pattern (DP), and what are the four essential elements of a DP? List the primary classification of DP into three categories (5)

C) Explain designing objects with the responsibilities in design pattern (5)

5) Explain the concept of low coupling and high cohesion for the following case study through a program "The SLCM software wants to store the student record for 7 subjects along with subject name and marks. The calculation of the total, average, and GPA is expected". (8)

A)

B) Write the program for implementing the factory design pattern (10)

C) What are the disadvantages of the observer pattern and how to overcome these disadvantages (2)

6) Write the UML class diagram for the observer design pattern and the bridge pattern. (10)

A)

B) Explain the different phases of software development life cycle (10)

7) Write the component diagram by considering any big software project of your choice. Show the interfaces, port and the dependencies properly (10)

A)

B) Explain the V model of software development along with an example (10)

8) Explain different testing methods along with the order in which they (10)

A)

B) In question 8A, mention the place where these testing usually happens. For a developer / programmer, which testing is possible at the project development place? (2)

C) Identify and write the possible test cases (with proper syntax) for the following method. (8)

```
/**@param TheToken, @return Bill, For RQ_HAS_1 */
Bill HotelAutomationSystem::CheckoutRoom(int TheToken)
{
    Bill TheBill = null; List<Bill> ListOfBills;

    // First condition check to know whether it is valid number or not
    if(TheToken<0)
        throw exception ("Invalid token number");

    // Second condition to know whether the bill already exist for this bill nu
    else if(ListOfBills.Contains(TheToken))
        thrown exception ("There exists a bill already for this token number");

    // If not, then generate the bill object and return to the caller.
    else
    {
        TheBill = new Bill();
        TheBill.Customer = "Altaf";
        TheBill.Amount = 20,000;
        TheBill.NumberOfDays = 2;
    }
    ListOfBills.Add(TheBill); //Add to the list of bills.
    return TheBill;
}
```

-----End-----