Department of Computer Science and Engineering
Motilal Nehru National Institute of Technology
Allahabad, Uttar Pradesh, India

Object Oriented Modeling (CS1504) + Object Based Modeling (CA3305)
B. Tech. (V Semester(CS+IT) and MCA-III Semester.

End Semester Exam   **Max Marks:60**   **Max Time: Three Hours**

- If you need to make any assumptions, state them clearly.
- **Questions carry marks shown against them.**

Q1. An application for creating an Online Ordering System (OOS.COM) is required to be developed. System should facilitate placing of orders for goods and perform various transactions as follows:

- A single order can be for multiple line items. (For example, with a single order a customer can purchase a book, bags, and umbrella.)
- Customers can cancel both entire orders and individual items in orders. Cancelling an item entails cancelling all items in the order.
- An item can be cancelled if it has not already been shipped or previously cancelled.
- Customers can also return items they do not want.

A customer may have any number of orders pending at any time. The OOS.COM system must generate a unique order number for each order. This number serves as a key the customer can use to identify the order. OOS maintains an inventory of goods. Order items are allocated from that inventory. When its inventory is exhausted, an item will be back-ordered from a supplier. Customer orders can be paid through credit or debit cards. Hence, each order must include a credit or debit card number. OOS.COM obtains funds from the card when the order is placed. OOS.COM ships only complete orders. If one or more items in an order are currently unavailable, the shipments of the other part of the order will be delayed until those items become available. The order processing system must also interact with legacy software. OOS.COM has a product warehouse legacy system that handles inventory management (including ordering goods from suppliers). It also has a legacy shipping system that handles shipping (including tracking the progress of the shipment). Develop Usecases and Usecase, Class, Sequence Diagrams for the above identifying system boundaries, actors (different types) and other artifacts of the modeling.   **[20]**

Q2.(A) How adding different levels of abstraction helps an application developer in writing flexible, evolvable and maintainable application code. Explain with class diagram having at least three levels of hierarchy for the same along with an example code in Java.   **[07]**

   (B) How can we support different levels of access to the same application code to different users using object oriented paradigm.   **[03]**

<div align="center">

P.T.O.

</div>

Q3.(A) Explain the concept of dependency injection, inversion of control, and autowiring in the design of application frameworks. You are also required to list the object oriented principles that facilitate the above concept in framework design. **[06]**

(B) Distinguish between static and dynamic binding. The compiler has the whole code, why it is not possible for compiler to decide the binding of messages to methods in some cases statically, by performing analysis of the code. **[03]**

(C) Describe various types of data members and method members of a class along with the accessibility amongst themselves. **[03]**

(D) Why Interfaces are required? Define Explicit and Implicit interfaces along with the description of their uses in object oriented software development supporting through example codes in Java/C++. **[03]**

Q4.(A) Consider a method *serve(Serviceable obj)*, as it can be seen from its signature, accepts an object exporting interface *Serviceable*. The interface *Serviceable* exports two methods *f()* and *g()* which are accessed by method *serve*. An existing class called *AlreadyExists* that has no relation with interface *Serviceable* exports two methods called *m()* and *n()*. It is found that these two methods implement exactly the functionality that is intended by *f()* and *g()* respectively. Devise a way of getting *m()* and *n()* of *AlreadyExists* served through *serve* without making a change to method *serve*, interface *Serviceable* and class *AlreadyExists*. **[05]**

(B) A class *X* has methods *f()* and *g()*. Due to security reasons the class *X* cannot be extended. The designer of the class *X* is already aware of the requirement of the addition of some methods which are not known yet. You are requested to provide design solution of above problem in a way so that new methods *m()*, *n()* and others in future may be added to the class *X* without extending it. Explain why your solution should work. **[04]**

(C) In visual modeling tools you can select an icon of objects or relationship among them and then create actual objects over the canvas. You can think of working of any tools like dia, paint brush or Microsoft office word for understanding the requirement of the design. You are required to provide the flexible design solution/s for the development as well as effective extensions of the above kind of tools. **[03]**

(D) You have to design support for 3 levels of embedding of the objects. There are some objects that cannot embed any other object. Provide a class diagram for efficient solution of the above problem that is designed to be used once. **[03]**