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. You are required to develop a small stock-trading system. Customers place buy and sell orders and securities transfer with service representative; then, the service representative enter those orders into the system. The system is closed, meaning that all buy orders are matched with sell orders within the system. (This system does not use orders placed with any other system.)
A security transfer moves stock or cash into or out of the system. Each transfer includes the customer's account number and either the amount of cash, or stock name and number of shares that is being transferred. A service representative also creates the accounts at customers' requests.
Each order includes the customer's account number, the name of the stock and the number of shares to be bought or sold, whether the order is buy or sell and the price customer is willing to pay or receive. (Orders must include a price because this system does not employ market prices.) Optionally, an order may include an expiration time indicating the date and time after which the order is no longer valid.
When an order is entered in the system, the service representative is given a unique order number for that order. Orders have a status. When they are placed, buy and sell orders are registered as open. When a buy order is matched with one or more sell orders, the involved buy and sell orders are registered as executed. An order that expires without being exercised is registered as expired. An order that is cancelled is registered as cancelled. By supplying an order number, a service representative may request the status of an order and any other relevant information about the order.
This system does not employ settlement period or short sales. To execute the buy or sell order, the customer's account must have sufficient fund or stock to cover the trade. You may match any number of buy orders to any number of sell orders. The only requirements are that in any match: the total number of buy orders shares equals the total number of sell order shares (That is, you do not match partial orders), and the asking price of any sell orders is no higher than offering price of any buy order. (When the two prices do not match, the final price may be anything between asking and the offering prices.) Design the Use-case, Class and Sequence Diagram for the above stock trading system along with code snippets in Java/C++. [15]

Q2. (A) 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]

P.T.O

(B) Are private members of super class are available in subclasses, if yes, why they are there if these being private can not be accessed in subclasses. [03]

(C) Show with an example code how the messages gets bound to appropriate methods. Explain also how it is decided what messages can be received by an object. [03]

(D) Describe various types of data members and method members of a class along with the accessibility of one member from the other. [03]

(E) Enumerate different ways to overload a function. Why are functions not overloaded through return types? [03]

Q3. (A) Consider a method being used in an application class as Serve(Serviceable x). We have an interface Serviceable which has methods f() and g(). A totally unrelated class X has methods m() and n() already implemented which has behavior of f() and g(). Design all solutions which make use of m() and n() methods for Serviceable and let method call Serve(x) working without any error in application class. [03]

(B) Describe how with the help of hooks/primitive methods a template pattern is implemented. Define template and primitive methods and differentiate between them. [03]

(C) List various ways to add operations in a class along with the constraints which necessitates them, if you are aware about that some operations need to be added in the class before designing the class but you don't know how may and which operations. [03]

(D) Identify the basic principles of object orientation used in creating the design pattern. Justify your answer. [03]

(E) Define and classify the design pattern along with the examples of each class. [03]

Q4. (A) Describe two kinds of reuse in object oriented modeling, which supports higher reuse and why? [05]

(B) Why java supports both abstract classes as well as interfaces. Can we do away with one, justify your answer. [02]

(C) Consider an order processing system for an international e-commerce company. This system must be able to process sales orders in many countries. The functions of the sales order object is allow to fill out the order with GUI, Handle tax calculations, Process the order and Print the sales receipt. Supposing after writing the applications I receive a request to change the way of handling the tax. Earlier the application has been written to calculate the tax using Indian tax rules. Now it is required that it should be able to handle the tax calculations according to rules in United States or United Kingdom and many more. For handling these new rules which design pattern is most appropriate and why, also draw its class diagram. [04]

(D) Describe the need of multiple interfaces and single implementation with the example code illustrating the various design principles used in the above example. [04]