25MCPC514

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)
Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

Model Question Paper

Second Semester MCA Degree Examination

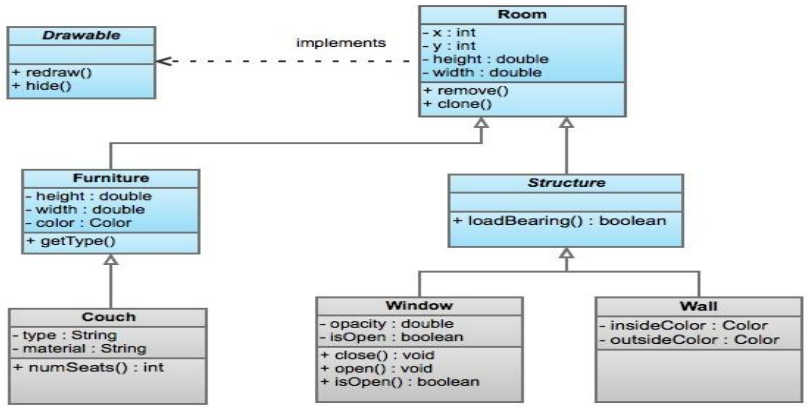
Object Oriented Modeling and Design

Time: 3 Hours **Max. Marks: 100**

Note: *1. Answer any FIVE full questions, choosing ONE full question from each module.*
2. M: Marks, L: RBT (Revised Bloom's Taxonomy) level, C: Course outcomes.

		Module -1	M	L	C
Q1	a.	Model a Library Management System by identifying and defining key classes, their attributes, and methods. You have to include classes such as Books, Members, Librarians, and Transactions. Define appropriate associations, generalizations, and dependencies.	10	L3	CO1
	b.	Design an object diagram where there are different classes such as, Organization (with attributes Name, Address and Phone number) which can have 1 object as College, Department (with attributes DeptId, DeptName and Venue) which can have multiple objects as department1, department2….. and Staff (with attributes StaffId and StaffName) / Students (with attributes StudentId and StudentName, venue) with various staffs and students as objects.	10	L3	CO1
		OR			
Q2	a.	In a healthcare system, a Patient receives a Treatment from a Doctor during a specific Visit, and the outcome is recorded as a Diagnosis. Create a UML class diagram showing an N-ary association between Patient, Treatment, Doctor, Visit, and Diagnosis. How would you represent this association to capture all five elements together? Justify your modeling approach.	10	L3	CO1
	b.	Create a class diagram for a project management system. A project consists of several tasks, but tasks can exist independently of the project and can be part of other projects. Use aggregation to represent the relationship between Project and Task.	10	L3	CO1
		Module- 2			
Q3	a.	Create a state diagram to represent the checkout process in an online shopping system. You have to include states such as browsing, adding items to the cart, viewing cart, entering shipping details, processing payment, and order confirmation. Use guard conditions like "if payment is successful" or "if stock is available" to control the transitions.	10	L3	CO2

	b.	Create a state diagram to represent the different stages in an ATM withdrawal process. The state diagram should include states like card insertion, PIN verification, selecting withdrawal amount, checking account balance, dispensing cash, and receipt printing. Use guard conditions such as "if sufficient balance" or "if ATM is not out of order" for transitioning between states.	10	L3	CO2
OR					
Q4	a.	Create a one-shot state diagram to represent the process of submitting and reviewing a document in an organization. The stages include document submission, initial review, feedback generation, document revision, final approval, and submission closure. Show the linear progression and transitions between these states.	10	L3	CO2
	b.	Design a UML State Machine Diagram for a bank loan application process using nested (composite) states. The states that have to be included are Application Submission, Document Verification, Credit Evaluation, Loan Approval, Loan Rejection, and Loan Disbursement. Model Document Verification and Credit Evaluation as composite states with appropriate substates, and use guard conditions such as [documents verified] and [credit score > threshold] to control state transitions. Show the complete flow from application submission to either loan disbursement or rejection.	10	L3	CO2
Module- 3					
Q5	a.	Develop a comprehensive UML Use Case Diagram for a Food Delivery Application. Identify and represent the primary actors such as Customer, Delivery Personnel, and Restaurant. Include major use cases such as User Registration/Login, Browse Restaurants, Search Food Items, Place Order, Make Payment, Track Order, Manage Menu, Accept/Reject Orders, Assign Delivery, Update Delivery Status, and View Order History. Clearly show the relationships between actors and use cases, and use appropriate <<include>> and <<extend>> relationships wherever applicable. Explain how the actors interact with the system throughout the food ordering and delivery process.	10	L3	CO3
	b.	Design a detailed UML Sequence Diagram for an Online Food Ordering System that illustrates the complete process of a customer placing an order. The key participants should include Customer, Food Ordering Application, Restaurant System, and Payment Gateway. Show the sequence of interactions involved in browsing the menu, selecting food items, placing the order, processing payment, confirming the order, notifying the restaurant, and sending order confirmation to the customer. Clearly represent the messages exchanged between objects, activation lifelines, and return messages. You should include alternative scenarios for successful payment and payment failure using appropriate UML notations.	10	L3	CO3
OR					

Q6	a.	Develop an activity diagram for purchasing a product on an e-commerce website. You diagram should include browsing products, adding to cart, checkout, payment, and order confirmation.	10	L3	CO3
	b.	Develop a detailed UML Collaboration (Communication) Diagram to represent the interactions among the objects involved in a customer withdrawing cash from an ATM system. Objects such as Customer, ATM Machine, Card Reader, Authentication Module, Bank Server, Account Database, Cash Dispenser, and Receipt Printer must be included. Show the sequence of messages exchanged for activities including card insertion, PIN authentication, account verification, balance checking, cash withdrawal request, transaction authorization, cash dispensing, account balance update, and receipt generation.	10	L3	CO3
Module- 4					
Q7	a.	You are reviewing a codebase for a payment processing system that currently supports credit card payments. The team wants to add support for digital wallets like PayPal and Apple Pay without modifying existing classes. Which SOLID principle is primarily violated if adding a new payment method requires changing the existing PaymentProcessor class? Propose a design using interfaces or abstract classes that adheres to this principle. Illustrate your solution with a simple UML diagram or pseudocode.	10	L3	CO4
	b.	Make use of the given class diagram below and write the appropriate C++/JAVA code snippet for the same. 	10	L3	CO4
OR					
Q8	a.	You are given a UML class diagram representing a university course registration system. It includes classes like Student, Course, Registration, and shows relationships such as associations, aggregations, and generalizations. Illustrate how you would translate this UML diagram into object-oriented code (e.g., Java or C++). Provide code snippets showing how you would implement inheritance, association, and aggregation based on the model. What challenges might arise during translation, and how can they be addressed?	10	L3	CO4
	b.	A company has a ShippingCalculator class that calculates shipping cost using a flat-rate formula. Now, they want to support multiple shipping	10	L3	CO4

		strategies (e.g., Standard, Express, International). How would you redesign the system to follow the Open-Closed Principle, allowing new shipping types to be added without modifying the existing class?			
Module- 5					
Q9	a.	You are developing a game engine where different types of enemies (e.g., Zombie, Robot, Dragon) can appear in various levels. You want to avoid using multiple if-else conditions to instantiate enemy types based on level difficulty. Which creational pattern would you use to dynamically create enemy objects at runtime? Justify your choice and illustrate how this pattern helps decouple object creation from the client code. Provide a simplified UML sketch or pseudocode to illustrate your implementation.	10	L3	CO5
	b.	An enterprise application is being upgraded to use a new logging framework, but several existing modules rely on an outdated logging API. How can the Adapter design pattern help ensure that existing modules continue to function without modifying their source code?	10	L3	CO5
OR					
Q10	a.	Your team is integrating a third-party payment gateway API into an existing e-commerce system. However, the interface provided by the third-party (NewPaymentAPI) doesn't match the expected interface (LegacyPaymentProcessor) used by your system. Which structural design pattern would help bridge this mismatch? Explain how it allows your system to work with incompatible interfaces. Provide a simple example showing how the adapter translates calls between the two interfaces.	10	L3	CO5
	b.	You are working on a document viewer application that loads large image files from a remote server. To improve performance, you want to delay loading images until they are actually needed (on-demand loading). How would you apply the Proxy pattern to implement lazy loading of images? Describe the structure of your solution, including the real subject, proxy, and common interface. What are the benefits of using this pattern in terms of memory usage and responsiveness?	10	L3	CO5