

MASTER OF COMPUTER APPLICATIONS (MCA_NEW/MCAOL)

SEMESTER-2 ASSIGNMENTS OF MCA_NEW/MCAOL PROGRAMME S

(July - 2026 & January - 2027)

MCS-218, MCS-219, MCS-220, MCS-221

MCSL-222, MCSL-223

**SCHOOL OF COMPUTER AND INFORMATION SCIENCES
INDIRA GANDHI NATIONAL OPEN UNIVERSITY
MAIDAN GARHI, NEW DELHI – 110 068**

CONTENTS

Course Code	Assignment No.	Submission-Schedule		Page No.
		For July-December Session	For January-June Session	
MCS-218	MCA_NEW/MCAOL(II)/218/Assign/26-27	31 st October, 2026	15 th April, 2027	3
MCS-219	MCA_NEW/MCAOL(II)/219/Assign/26-27	31 st October, 2026	15 th April, 2027	5
MCS-220	MCA_NEW/MCAOL(II)/220/Assign/26-27	31 st October, 2026	15 th April, 2027	7
MCS-221	MCA_NEW/MCAOL(II)/221/Assign/26-27	31 st October, 2026	15 th April, 2027	9
MCSL-222	MCA_NEW/MCAOL(II)/L-222/Assign/26-27	31 st October, 2026	15 th April, 2027	10
MCSL-223	MCA_NEW/MCAOL(II)/L-223/Assign/26-27	31 st October, 2026	15 th April, 2027	12

Important Notes

1. MCA_NEW Student need to submit the assignments to the Coordinator of your Study Centre on or before the due date.
2. MCAOL Student need to submit the assignments on LMS Portal on or before the due date.
3. Assignment submission before due dates is compulsory to become eligible for appearing in corresponding Term End Examinations. For further details, please refer to Programme Guide.
4. To become eligible for appearing the Term End Practical Examination for the lab courses, it is essential to fulfill the minimum attendance requirements as well as submission of assignments (on or before the due date). For further details, please refer to the Programme Guide.
5. The viva voce is compulsory for the assignments. For any course, if a student submitted the assignment and not attended the viva-voce, then the assignment is treated as not successfully completed and would be marked as ZERO.

Course Code	:	MCS-219
Course Title	:	Object Oriented Analysis and Design
Assignment Number	:	MCA_NEW/MCAOL(II)/219/Assign/2026-27
Maximum Marks	:	100
Weightage	:	30%
Last date of Submission	:	31st October, 2026 (For July, 2026 Session) 15th April, 2027 (For January, 2027 Session)

Note: This assignment has eight questions of 80 Marks. Answer all questions. Rest 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding assignments given in the Programme Guide for the format of presentation.

- Q1:**
- a.** Analyse the basic philosophy of the Object-Oriented (OO) paradigm. Contrast it from the classical Structured Analysis and Design approach, specifically focusing your evaluation on data-control coupling, computational encapsulation, and long-term system maintainability. **(4 Marks)**
 - b.** Explain the four foundational architectural pillars of object orientation: Abstraction, Encapsulation, Inheritance, and Polymorphism with the help of example(s). **(6 Marks)**
- Q2:**
- a.** Evaluate the structural and semantic variations between Aggregation (weak “has-a” relationship) and Composition (strong “part-of” ownership lifecycle). Support your analysis with concrete software engineering examples. **(5 Marks)**
 - b.** Critically examine the distinct roles of Abstract Classes and Interfaces as explicit design contracts in system architecture. Analyze the trade-offs of choosing one over the other when designing decoupling boundaries. **(5 Marks)**
- Q3:** Draw a Class Diagram for a cloud-native Learning Management System (LMS). The structural model must robustly handle multi-tiered authentication (Instructors, Students, Administrators), hierarchical course architectures (Modules, Lessons, Assessments), and asynchronous enrollment tracking. (Note: Your diagram must explicitly utilize correct UML notation for multiplicities, visibility modifiers (+, -, #), directional associations, generalization hierarchies, and dependencies). State all underlying domain assumptions. **(10 Marks)**
- Q4:** Draw a comprehensive State Machine Diagram tracking the lifecycle of an order transaction within an Enterprise e-Commerce Portal. The state machine must process complex internal/external events, explicit guard conditions, and alternate transactional path routings (including: Inventory_Reserved, Payment_Processing, Authorization_Failure, Order_Dispatched, and Refund_Initiated). Make necessary assumptions **(10 Marks)**
- Q5:** Explain the distinct utility and interplay of Object Modeling (Structural), Dynamic Modeling (Behavioral), and Functional Modeling (Dataflow) in modern software engineering. Using an Automated Banking Terminal (ATM) as a continuous reference point, outline how each of these three modeling dimensions captures a unique, non-overlapping aspect of the execution runtime environment. **(10 Marks)**
- Q6:**
- a.** Explain features of synchronization in UML. **(3 Marks)**
 - b.** Systematically detail the major architectural execution tasks performed during system design, specifically addressing: **(7 Marks)**
 - 1.Partitioning monolithic architectures into decoupled, autonomous subsystems.
 - 2.Concurrency design and process allocation strategies.
 - 3.Global resource allocation and data store management paradigm selection.
 - 4.Boundary condition handling (initialization, termination, and failure-recovery modes).

- Q7:** Map the complete object class model designed for the Learning Management System (LMS) in Question 3 into a 3NF-compliant Relational Database Management System (RDBMS) table schema. Clearly document the mapping patterns applied to transform object characteristics into flat relational structures. Explicitly specify your strategy for representing inheritance hierarchies (e.g., Table-Per-Class vs. Single-Table inheritance), resolving many-to-many (M:N) associations, and preserving referential integrity constraints via Primary Key (PK) and Foreign Key (FK) mechanics. **(10 Marks)**
- Q8:** **a.** Critically evaluate the practical hurdles of translating abstract UML structures into concrete code within modern programming languages (like C++, Java, C#). Address mismatch phenomena such as resolving structural associations and preserving encapsulation boundaries. **(5 Marks)**
- b.** Draw component diagram for the Online Shopping System. **(5 Marks)**