Uttar Pradesh Rajarshi Tandon Open University, Prayagraj

Assignment Paper for M.Sc. (Computer Science)
School of Sciences

Session: **2026 -27**	Max. Marks: **30**
Program Name: **Master of Computer Science (M.Sc. CS)**	
Course Code: **MCS 101N**	Course Name: **Discrete Mathematical**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Differentiate between tautology, contradiction, and contingency using suitable truth tables. Explain the role of quantifiers in mathematical logic with an example.	2
2.	Convert the Boolean function $F(A,B,C)=A\bar{B}+BC+\bar{A}C$ into Disjunctive Normal Form (DNF). Explain the importance of Boolean algebra in simplifying logic expressions.	2
3.	Explain the relationship between equivalence relations, equivalence classes, and partitions of a set with a suitable example.	2
4.	A department has 8 faculty members and wants to select a Chairperson, Coordinator, and Treasurer. Explain whether permutations or combinations should be used and justify your answer.	2
5.	State the Inclusion–Exclusion Principle. Explain its difference from the Addition Rule of Counting with one practical application.	2
6.	Explain the importance of switching circuits in digital systems. Describe how Boolean expressions help in circuit simplification.	2
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain the principles of Object-Oriented Programming related to mathematical logic applications: classes, objects, abstraction, encapsulation, inheritance, and polymorphism. Discuss their importance in software design. **(b)** Minimize the Boolean function using Karnaugh Map: $F(A,B,C,D)=\Sigma m(0,2,5,7,8,10,13,15)$ Draw the simplified logic circuit.	6

8.	**(a)** Let A={1,2,3,4,5,6} and define relation: R={(a,b): a ≡ b (mod 2)} Determine whether R is an equivalence relation. Find equivalence classes and explain their relation with partitions. **(b)** Prove using Mathematical Induction: $1+3+5+...+(2n-1)=n^2$ Also determine the number of ways to select 4 electives from 10 subjects and justify the use of combinations.	6
9.	**(a)** Expand $\mathbf{(2x-3)^8}$ using Binomial Theorem. Find the general term and middle term(s). Discuss applications of binomial expansion in computer science. **(b)** In a survey of 150 students: Python users = 80, Java users = 65, C++ users = 50 Python & Java = 30, Java & C++ = 20, Python & C++ = 18, All three = 10. Using the Inclusion–Exclusion Principle: (i) Find students studying at least one language. (ii) Find students studying none. (iii) Explain the application of Pigeonhole Principle in large datasets.	6

Session: **2025 -26**	Max. Marks: **30**
Program Name: **Master of Computer Science (M.Sc. CS)**	
Course Code: **MCS 102N**	Course Name: **C++ and Object-oriented programming**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Compare procedural programming and object-oriented programming. Explain any four advantages of OOP with suitable examples.	**2**
2.	Explain encapsulation, abstraction, and information hiding. How do these concepts improve software quality and maintainability?	**2**
3.	Differentiate between friend functions and member functions in C++. Explain one suitable application of each.	**2**
4.	Explain the role of constructors and destructors in object initialization and cleanup. Why is constructor overloading useful?	**2**
5.	Explain operator overloading in C++. State the rules governing operator overloading and discuss one practical application involving user-defined data types.	**2**
6.	Why are virtual functions essential for achieving runtime polymorphism? Explain with a suitable real-world example.	**2**
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	(a) Discuss the principles of Object-Oriented Programming—Classes & Objects, Encapsulation, Abstraction, Inheritance, and Polymorphism. Compare OOP with Procedural Programming and explain why C++ is suitable for implementing OOP concepts. (b) A software company is developing an Employee Management System. Design suitable classes and objects. Explain how inheritance, dynamic binding, late binding, and reusability improve the software design. Illustrate your answer with a suitable class diagram.	**6**
8.	(a) Write a C++ program to create a Student class containing Roll Number, Name, and Marks. Implement a default constructor, parameterized constructor, copy constructor, and destructor. Create objects using different constructors and explain the order of constructor and destructor invocation. (b) Differentiate among Single, Multiple, Multilevel, Hierarchical, and Hybrid Inheritance with suitable diagrams and C++ syntax. Also explain the purpose of virtual base classes	**6**

	and abstract classes with appropriate examples.	
9.	(a) A university wishes to develop a Student Academic Record System. Write a C++ program using base and derived classes, virtual functions, runtime polymorphism, the this pointer, and pointers to objects. Explain how runtime polymorphism improves software extensibility. (b) Design an Object-Oriented Library Management System. Your answer should include: • Identification of objects and classes. • Class and object diagrams. • C++ file handling for storing and retrieving records. • Opening, closing, reading, writing, and error handling during file operations. • Use of command-line arguments for processing library database files.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 103N**	Course Name: **Data Structures**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Distinguish between Linear and Non-Linear Data Structures across four attributes: definition, memory arrangement, traversal method, and example. Present your answer as a comparison table.	**2**
2.	A two-dimensional array A[1..4][1..5] is stored in row-major order. The base address is 100 and each element occupies 2 bytes. Calculate the address of element A[3][4] using the row-major address formula. Show all steps clearly.	**2**
3.	Convert the following infix expression to postfix using a stack. Show the stack contents and output at each step in a table: Infix: A + B * C − D / E	**2**
4.	A Circular Queue has a maximum size of 6 and currently contains elements: Front = 1, Rear = 4, elements = [_, 10, 20, 30, 40, _]. Perform the following operations in sequence and show the queue state after each: *(i)* Enqueue 50 *(ii)* Enqueue 60 *(iii)* Dequeue *(iv)* Enqueue 70 State the condition for queue full and queue empty in a circular queue.	**2**
5.	Insert the following keys one by one into an initially empty Binary Search Tree (BST): 50, 30, 70, 20, 40, 60, 80 Draw the resulting BST. Then perform an Inorder Traversal and write the output sequence. State one property of inorder traversal of a BST.	**2**
6.	Apply Dijkstra's Shortest Path Algorithm on the following weighted graph starting from vertex A: Edges: A–B (4), A–C (2), B–C (1), B–D (5), C–D (8), C–E (10), D–E (2) Show the distance table at each step and state the shortest path from A to E with total cost.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain Algorithm Complexity and Arrays: *(i)* Define Time Complexity and Space Complexity of an algorithm. Explain the following asymptotic notations in one sentence each with a graphical sketch: Big-O (O), Omega (Ω), and Theta (Θ). State the time complexity of linear search and binary search using Big-O notation.	**6**

	(ii) Explain Sparse Matrix representation — define a sparse matrix and state why storing it in a standard 2-D array is wasteful. Show the triplet representation of the following sparse matrix:
	0 0 3 0
	0 0 0 0
	1 0 0 5
	0 2 0 0
	State how many non-zero elements exist and how the triplet table is structured (row, column, value).
	
	(b) Explain Stacks and Recursion:
	(i) Define a Stack and state its LIFO property. Show the array representation of a stack of size 5. Trace the following sequence of operations on an initially empty stack and show the stack state after each operation:
	Push(10), Push(20), Push(30), Pop(), Push(40), Pop(), Pop()
	(ii) Evaluate the following postfix expression using a stack. Show the stack state at each step in a table:
	6 2 3 + − 3 8 2 / + * 2 ↑ 3 +
	(iii) Explain Tail Recursion — define it, state one advantage over non-tail recursion, and convert the following recursive factorial function to a tail-recursive version:
	int factorial(int n) {
	if (n == 0) return 1;
	return n * factorial(n − 1);
	}	

8.	**(a)** Explain Linked Lists: *(i)* Draw the structure of a Singly Linked List node showing the data and next pointer fields. Write pseudocode for the following operations on a singly linked list: • Insert a node at the beginning • Delete a node with a given value *(ii)* Distinguish between Singly Linked List, Doubly Linked List, and Circular Linked List across three attributes: pointer fields per node, traversal direction, and deletion ease. Draw a 3-node example of each. *(iii)* Explain Polynomial Representation using Linked List — show how the polynomial $P(x) = 4x^3 + 3x^2 + 2x + 1$ is represented as a linked list where each node stores coefficient, exponent, and next pointer. Draw the linked list. **(b)** Explain Trees and AVL Trees: *(i)* Define the following binary tree terms in one sentence each: Root, Leaf, Height, Depth, Complete Binary Tree, Extended Binary Tree. *(ii)* Explain AVL Tree — define the Balance Factor (BF = Height of Left Subtree − Height of Right Subtree). Insert the following keys one by one into an empty AVL tree and perform necessary rotations to maintain balance: 10, 20, 30 Show the tree after each insertion, identify the imbalance type (LL, RR, LR, RL), and draw the tree after rotation. *(iii)* State the difference between a B-Tree and a Binary Search Tree across two attributes: number of keys per node and use case. State the order of a B-Tree and what it implies.	**6**
9.	**(a)** Explain Graph Algorithms: *(i)* Represent the following undirected graph using both an Adjacency Matrix and an Adjacency List. State one advantage of each representation. Vertices: {A, B, C, D}, Edges: {A–B, A–C, B–C, B–D, C–D} *(ii)* Apply Breadth First Search (BFS) and Depth First Search (DFS) on the above	**6**

	graph starting from vertex A. Show the order of visited vertices for each traversal and state the data structure used internally by each algorithm. *(iii)* Apply Kruskal's Algorithm to find the Minimum Cost Spanning Tree of the following weighted graph. Show the edges sorted by weight, the edge selection steps, and the final MST with total cost. Edges: A–B(1), A–C(3), B–C(2), B–D(4), C–D(5) **(b)** Explain Sorting, Searching, and Hashing: *(i)* Trace Quick Sort on the array: [38, 27, 43, 3, 9, 82, 10] using the first element as pivot. Show the array state after each partition step and state the best-case and worst-case time complexity of Quick Sort. *(ii)* Trace Binary Search on the sorted array [5, 12, 23, 35, 47, 58, 69, 78] for the target value 47. Show the low, mid, and high values at each step. Compare Binary Search with Sequential Search across three attributes: prerequisite, time complexity, and suitability. *(iii)* Explain Collision Resolution in Hashing — define collision and explain any two strategies: Linear Probing and Chaining. Show how the keys {12, 25, 38, 6, 19} are inserted into a hash table of size 7 using the hash function h(k) = k mod 7 with linear probing. Show the final hash table.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **PGBR-01N**	Course Name: **Basics in Research**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Write down the meaning and objective of research.	**2**
2.	Discuss the need for reviewing literature in brief. What are the types of literature review?	**2**
3.	What do you understand by google scholar, science direct?	**2**
4.	What do you understand by google Scopus, web of science?	**2**
5.	write short notes on the following: a) Journal abstracts b) SciFinder	**2**
6.	Write short notes on the following: a) Citation index b) Peer review and revision.	**2**
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	Discuss in detail about the intellectual property and intellectual property rights (IPR).	**6**
8.	Write short notes on the following- a) Citation index b) Peer review and revision.	**6**
9.	What are the various kinds of report writing in academics and research. Explain in detail.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 106N**	Course Name: **Computer Organization**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Explain the significance of different number systems (Binary, Octal and Hexadecimal) in digital computers. Convert $(7A9)_{16}$ into Binary and Decimal, and briefly describe one practical application of hexadecimal representation.	**2**
2.	State De Morgan's Theorems and explain their importance in simplifying Boolean expressions. Verify the theorem (A+B)' = A'B' using a truth table and identify the equivalent logic gate implementation.	**2**
3.	Differentiate between combinational and sequential circuits with suitable examples. Draw the logic diagram and truth table of a Full Adder, and explain its role in binary arithmetic.	**2**
4.	Explain the working of SR and JK Flip-Flops with neat diagrams and characteristic tables. Why is the JK Flip-Flop preferred over the SR Flip-Flop in digital circuit design?	**2**
5.	Compare Static RAM (SRAM) and Dynamic RAM (DRAM) with respect to structure, speed, cost and applications. Why is cache memory essential for improving processor performance?	**2**
6.	Differentiate between a microprocessor and a microcontroller. Explain any four features of the 8085 microprocessor, and identify the addressing mode used in the instruction: MVI A, 45H	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	(a) Explain the standard representation of Boolean expressions using SOP (Sum of Products) and POS (Product of Sums) forms. Discuss the concepts of Minterms, Maxterms, and Karnaugh Map (K-Map) simplification with suitable examples. (b) Simplify the following Boolean expression using a 4-variable Karnaugh Map, and implement the simplified expression using only NAND gates. $F(A,B,C,D)=\Sigma m(0,2,3,5,7,8,10,11,15)$ Draw the K-Map, show grouping clearly, obtain the minimized expression, and sketch the corresponding logic circuit.	**6**

8.	(a) Explain the architecture of a digital computer by describing the functions of the Arithmetic Logic Unit (ALU), Control Unit, Input Unit, Output Unit, and Memory Unit. Illustrate the Instruction Execution Cycle with a neat block diagram and explain the role of the Program Counter and Instruction Register. (b) Compare the following addressing modes with suitable examples: • Direct • Immediate • Indirect • Indexed • Relative For each addressing mode: • Explain how the effective address is obtained. • Identify one suitable application. • Determine the effective address for the following data: Base Register = 3000H Index Register = 0020H Instruction Address = 2500H Operand Address = 4000H	**6**
9.	(a) Explain the memory hierarchy of a computer system with a neat diagram. Compare RAM, ROM, Cache Memory, Virtual Memory, EEPROM and EPROM in terms of speed, volatility, storage technology and applications. Discuss how cache memory improves CPU performance. (b) Design a conceptual computer system for a college examination management system. Your design should include: • Selection of suitable processor • Memory organization • Input devices • Output devices • System buses • Storage devices Justify your design choices based on cost, performance and reliability. Also explain how data moves between the CPU, memory and I/O devices during result generation.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 108N**	Course Name: **Data Communication and Computer Networks**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Explain the layered architecture of computer networks with reference to the ISO-OSI model. Describe the primary function of any four layers and illustrate how data moves from the sender to the receiver.	**2**
2.	Differentiate between Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM). A communication channel has a bandwidth of 12 MHz and is divided equally among 6 users using FDM. Calculate the bandwidth allocated to each user and state one advantage of TDM over FDM.	**2**
3.	Compare Pure ALOHA and Slotted ALOHA in terms of channel access mechanism, collision probability and throughput. Calculate the maximum theoretical throughput of both protocols and explain why Slotted ALOHA performs better.	**2**
4.	Explain the working of the Stop-and-Wait ARQ protocol over a noisy communication channel using a flow diagram. Predict the sequence of events if an acknowledgment (ACK) is lost during transmission.	**2**
5.	Differentiate between Bridges, Routers and Gateways based on functionality, OSI layer, addressing mechanism and practical applications. Give one real-world example where each device is used.	**2**
6.	Explain the purpose of Quality of Service (QoS) in the Transport Layer. Discuss any four QoS parameters and identify two real-time applications where QoS is essential.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Discuss the architecture of the ISO-OSI Reference Model. Explain the responsibilities of each layer and compare the OSI model with the TCP/IP model in terms of architecture, protocols, advantages and limitations. Illustrate your answer with neat diagrams. **(b)** A company plans to connect five branch offices located in different cities through a high-speed communication network. Design a suitable communication system by selecting: • Appropriate transmission media • Multiplexing technique • Network topology • Suitable communication devices	**6**

	• Error control mechanism Justify each choice considering bandwidth, cost, reliability and scalability.	
8.	**(a**) Explain the operation of the Data Link Layer with special reference to: • Stop-and-Wait Protocol • HDLC Protocol • IEEE 802.4 Token Bus • IEEE 802.5 Token Ring • FDDI Compare their advantages, limitations and suitable application areas. **(b)** A network transmits 1000-bit frames over a 1 Mbps link. Propagation delay = 20 ms ACK size is negligible. Calculate: 1. Transmission time of one frame. 2. Channel utilization using the Stop-and-Wait protocol. 3. Suggest one protocol that would improve utilization and justify your answer. 4. Draw the sender and receiver state transition diagram for the Stop-and-Wait protocol.	**6**
9.	**(a)** Explain the working principles of the following routing algorithms: • Dijkstra's Shortest Path Algorithm • Bellman-Ford Algorithm • Floyd-Warshall Algorithm Compare them based on routing strategy, computational complexity, advantages, limitations and suitable network environments. **(b)** Consider the following weighted network. Link Cost A–B 4 A–C 2 B–C 1 B–D 5 C–D 8 C–E 10 D–E 2 D–F 6 E–F 3 Using Dijkstra's Algorithm: 1. Determine the shortest path from node A to every other node. 2. Draw the shortest path tree. 3. Calculate the total cost to reach node F. 4. Comment on how routing tables are updated when link costs change.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 109N**	Course Name: **Database Management System**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Explain the characteristics of the Database Approach and distinguish it from the traditional file processing system. Illustrate your answer with a suitable real-world example.	**2**
2.	Construct an Entity-Relationship (ER) diagram for a Library Management System consisting of Book, Member, Librarian and Issue entities. Identify the primary keys, cardinality constraints and participation constraints, and explain how the ER model can be converted into relational tables.	**2**
3.	Differentiate between procedural and non-procedural query languages. Write SQL statements to: • Create a Student table with appropriate constraints. • Insert one record. • Retrieve students scoring more than 75 marks.	**2**
4.	Consider the relation: EMPLOYEE(EmpID, Name, Dept, Salary) Write suitable Relational Algebra expressions to: • Retrieve employees belonging to the IT department. • Display employee names with salary greater than ₹60,000. • Project only EmpID and Name.	**2**
5.	Explain the concept of Functional Dependency (FD) with suitable examples. Determine whether the following dependencies satisfy Second Normal Form (2NF): STUDENT(RollNo, Name, Course, CourseFee) FDs: • RollNo → Name, Course • Course → CourseFee Justify your answer.	**2**
6.	State the ACID properties of a database transaction. Explain how concurrency control maintains database consistency when multiple users update the same database simultaneously.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**

7.	**(a)** Explain the architecture of a Database Management System (DBMS). Discuss the three-schema architecture, data independence, data models, database languages (DDL, DML, DCL, TCL), and classification of DBMS. Illustrate your answer with neat diagrams. **(b)** A university wants to automate its Student Information System. Design an ER model containing at least the following entities: • Student • Department • Faculty • Course • Enrollment Your answer should include: • ER diagram • Entity attributes • Primary and foreign keys • Cardinality and participation constraints • Conversion of the ER diagram into relational tables	**6**
8.	**(a)** Explain the Relational Data Model and discuss relational constraints, update operations, integrity constraints and functional dependencies. Compare Primary Key, Candidate Key, Alternate Key, Foreign Key, and Composite Key with suitable examples. **(b)** Consider the relation: ORDER(OrderID, CustomerID, CustomerName, ProductID, ProductName, SupplierID, SupplierName, Quantity) The following functional dependencies hold: • OrderID → CustomerID • CustomerID → CustomerName • ProductID → ProductName, SupplierID • SupplierID → SupplierName Perform normalization up to **BCNF**. Your answer should include: • Identification of candidate keys • Detection of partial and transitive dependencies • Step-by-step decomposition into 1NF, 2NF, 3NF and BCNF • Final normalized relations with keys	**6**
9.	**(a)** Explain the concepts of transaction processing, serializability, recoverability, concurrency control, timestamp protocol, validation protocol, deadlock handling, and database recovery techniques using suitable examples and diagrams. **(b)** A banking database supports concurrent transactions on customer accounts. Consider the following transactions: T1 • Read(A) • A = A − 500 • Write(A) • Read(B) • B = B + 500	**6**

	• Write(B) T2 • Read(A) • A = A × 1.05 • Write(A) Answer the following: 1. Identify possible concurrency problems. 2. Draw a precedence graph for one possible schedule. 3. Determine whether the schedule is conflict serializable. 4. Explain how Two-Phase Locking (2PL) and Timestamp Ordering resolve the problem. 5. Describe the recovery procedure if the system crashes immediately after Write(A) of transaction T1.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **PGED-02**	Course Name: **Entrepreneurship development**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	What is entrepreneurship? What are the main characteristics of an entrepreneur to do entrepreneurship?	**2**
2.	Differentiate between Entrepreneurs, Intrapreneur &Manager.	**2**
3.	What are the problems faced by women entrepreneurs in doing business?	**2**
4.	Explain the characteristics of projects with various types of project?	**2**
5.	Highlight the need of technology for entrepreneurs	**2**
6.	What is the various assistance of financial Institutions to entrepreneurs?	**2**
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	Explain various type of entrepreneurship. Explain eight stages of Entrepreneurship Development Cycle.	**6**
8.	What are the needs and Objectives of Entrepreneurship Development Programs.	**6**
9.	Explain any four all India Financial Institutions who aid to entrepreneur.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 111N**	Course Name: **Design And Analysis Of Algorithms**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Define an algorithm and explain the importance of time complexity and space complexity in algorithm analysis. Using Big-O notation, determine the time complexity of the following pseudocode: for i = 1 to n for j = 1 to n print(i + j)	**2**
2.	Differentiate between Breadth First Search (BFS) and Depth First Search (DFS) with respect to traversal strategy, data structures used, time complexity and applications. Draw a suitable graph and illustrate both traversals.	**2**
3.	Explain the Divide and Conquer strategy. Trace the execution of the Binary Search algorithm for the sorted array: {5, 12, 18, 25, 32, 40, 47, 55, 63} Search Key = 40 Show each iteration and identify the time complexity.	**2**
4.	Differentiate between Counting Sort, Radix Sort, and Bucket Sort with suitable examples. Under what conditions is each algorithm preferred over comparison-based sorting algorithms?	**2**
5.	Explain the Greedy Method and discuss the conditions under which a greedy algorithm produces an optimal solution. Illustrate the construction of a Huffman Coding Tree for the characters: Character Frequency A 5 B 9 C 12 D 13	**2**
6.	Define NP, NP-Hard, and NP-Complete problems. Explain why the Boolean Satisfiability (SAT) problem is considered NP-Complete and discuss one real-world application of NP-Complete problems.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**

7.	(a) Explain asymptotic notations (Big-O, Big-Ω and Big-Θ) with suitable examples. Discuss the Substitution Method and Master Method for solving recurrence relations, and explain their importance in the analysis of recursive algorithms. (b) Solve the following recurrence relation using the Master Method: $T(n)=2T(n/2)+n$ Also analyze the time complexity of Merge Sort and compare it with Quick Sort and Heap Sort in terms of: • Best, average and worst-case complexities • Space requirements • Stability • Practical applications	**6**
8.	(a) Compare the Greedy Method and Dynamic Programming approaches. Explain their design principles, advantages, limitations and suitable applications. Discuss the solutions of the following problems: • Fractional Knapsack • 0/1 Knapsack • Matrix Chain Multiplication • Travelling Salesperson Problem (TSP) Highlight why some problems are better solved using Dynamic Programming instead of the Greedy approach. (b) Given the following set of items: Item Weight Profit A 2 20 B 3 30 C 4 40 D 5 50 Knapsack Capacity = 8 1. Solve the problem using the Greedy (Fractional Knapsack) approach. 2. Solve the same problem using the 0/1 Knapsack Dynamic Programming algorithm. 3. Compare the obtained solutions and justify the difference in total profit.	**6**
9.	(a) Explain the Backtracking and Branch-and-Bound strategies. Discuss their state-space tree representation, pruning techniques and applications in solving: • 8-Queen Problem • Graph Coloring • Hamiltonian Cycle • Travelling Salesperson Problem Compare both techniques with suitable examples. (b) Consider the following weighted graph. Edge Weight A–B 4 A–C 2 B–C 1 B–D 5 C–D 8	**6**

	C–E 10 D–E 2 D–F 6 E–F 3 Perform the following tasks: 1. Apply Dijkstra's Algorithm to determine the shortest path from A to all other vertices. 2. Construct the Minimum Spanning Tree (MST) using Kruskal's Algorithm. 3. Calculate the total weight of the MST. 4. Compare the objectives of shortest-path algorithms and minimum spanning tree algorithms.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 112N**	Course Name: **Java Programming**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Differentiate between Object-Oriented Programming (OOP) and Procedure-Oriented Programming (POP) with respect to data handling, modularity, and code reusability. Support your answer with a brief real-world analogy — for example, how a *Bank Account* would be modelled differently in each paradigm.	**2**

2.	Predict the exact output of the following Java code snippet and justify each result by explaining integer overflow, implicit type promotion, and operator precedence:
	`int a = 2147483647; // Integer.MAX_VALUE`
	`int b = a + 1;`
	`byte c = (byte)(a + 1);`
	`int d = 10 + 3 * 4 - 6 / 2;`
	`System.out.println(b + " " + c + " " + d);`
		2

3.	Trace the following Java program step by step and determine the exact output. Identify the sorting algorithm being used, state its worst-case time complexity, and explain what would happen if the inner loop condition were changed to `j < arr.length`:
	`int[] arr = {5, 3, 8, 1, 9};`
	`for (int i = 0; i < arr.length - 1; i++)`
	`for (int j = 0; j < arr.length - i - 1; j++)`
	`if (arr[j] > arr[j+1]) {`
	`int t = arr[j]; arr[j] = arr[j+1]; arr[j+1] = t;`
	`}`
	`for (int x : arr) System.out.print(x + " ");`
		2

4.	Explain the concept of method overloading in Java with the help of a short example. How does the Java compiler resolve which overloaded method to invoke at compile time? Also state two rules that must be followed for valid method overloading.	**2**
5.	A Java program defines the following class hierarchy: Animal → Mammal → Dog Explain multilevel inheritance using this hierarchy. Write a short Java code snippet (8–10 lines) demonstrating how the `Dog` class inherits and overrides a method `sound()` from `Animal`. Also state the role of the `super` keyword in this context.	**2**
6.	Explain the Java Thread Life Cycle with a neat labelled diagram showing all states: *New, Runnable, Running, Blocked/Waiting,* and *Terminated*. Briefly describe the method or event that causes a transition between each pair of adjacent states.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**

7.	**(a)** Explain the four fundamental principles of Object-Oriented Programming — Abstraction, Encapsulation, Inheritance, and Polymorphism — with reference to Java. For each principle, provide one practical example drawn from a real-world domain (e.g., banking system, hospital management, e-commerce). Discuss how these principles collectively contribute to better software design, maintainability, and code reuse. Also briefly describe any three applications of OOPS in modern software development. **(b)** A university stores data about its students. Design a Java class `Student` with the following specifications: • Instance variables: `rollNo` (int), `name` (String), `marks` (double array of size 5) • Methods: o `inputData()` — assign values directly within the method o `computeResult()` — calculate total and percentage o `displayResult()` — print all details Write the complete Java program. In `main()`, create two `Student` objects, call all three methods on each, and display the results. Demonstrate how encapsulation is achieved in your design by using appropriate access modifiers.	**6**
8.	**(a)** Explain Packages and Interfaces in Java. Discuss: *(i)* The purpose and structure of a Java package; how to create a user-defined package and import it into another class. *(ii)* The concept of an interface — how it differs from an abstract class, and when to prefer one over the other. *(iii)* How a class can use both `extends` and `implements` simultaneously, and what constraints apply. Support your explanation with appropriate syntax examples and a comparison table between *interface* and *abstract class* covering at least four attributes (e.g., method types, constructors, multiple inheritance support, default methods). **(b)** Consider the following scenario: A payments application processes transactions. When a user enters an invalid amount (negative or zero), the system must throw a custom exception `InvalidAmountException`. If the account balance is insufficient, a `InsufficientFundsException` must be raised. (i) Write Java code to define both custom exception classes by extending the `Exception` class. (ii) Write a `PaymentProcessor` class with a method `processPayment(double amount, double balance)` that throws these exceptions using `throw`, handles them in `main()` using `try-catch-finally`, and prints appropriate messages. (iii) Identify whether each exception is *checked* or *unchecked* and justify your classification.	**6**
9.	**(a)** Explain Multithreaded Programming in Java covering the following: *(i)* The concept of multithreading and how it differs from multiprocessing. State at least three advantages of multithreading in application development. *(ii)* Two ways to create a thread in Java (extending `Thread` class vs. implementing `Runnable` interface) — compare both approaches with pros and cons. *(iii)* The concept of synchronization — why it is needed, and how the `synchronized` keyword prevents race conditions. Illustrate with a brief example	**6**

	involving a shared counter variable accessed by two threads. **(b)** Consider the following practical scenario involving Java Networking: A client application needs to connect to a server running on `localhost` at port `5000`, send a string message "`HELLO SERVER`", and receive an acknowledgement response from the server. (i) Write Java code for the Server side using `ServerSocket` and `Socket` to accept the connection, read the client message, and send back "`ACK: MESSAGE RECEIVED`". (ii) Write Java code for the Client side using `Socket`, `PrintWriter`, and `BufferedReader` to establish the connection, send the message, and print the server's response. (iii) Briefly explain the difference between TCP/IP Sockets and Datagrams (UDP) in Java networking. State one use case where each is preferred.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 113N**	Course Name: **Operating System**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Distinguish between Multiprogramming, Time-Sharing, and Multiprocessing operating system environments. For each, state the primary objective, the number of CPUs involved, and one real-world example of an OS or application that uses that model. Present your answer in a structured comparison table followed by a brief explanatory note.	**2**
2.	Draw a neat labelled diagram of the Process State Transition Model showing all five states: *New*, *Ready*, *Running*, *Waiting*, and *Terminated*. For each transition (arrow), identify the specific event or system call that causes it (e.g., scheduler dispatch, I/O request, I/O completion).	**2**
3.	Consider the following set of processes with their arrival times and burst times: Process Arrival Time (ms) Burst Time (ms) P1 0 8 P2 1 4 P3 2 2 P4 3 1 Apply the Shortest Job First (SJF) Non-Preemptive scheduling algorithm. Draw the Gantt chart and calculate the Average Waiting Time and Average Turnaround Time.	**2**
4.	Explain the concept of a Critical Section and state the three necessary conditions (Mutual Exclusion, Progress, Bounded Waiting) that any valid solution to the critical section problem must satisfy. Describe briefly how a Semaphore enforces mutual exclusion using wait() and signal() operations with a short pseudocode illustration.	**2**
5.	A system has the following resource-allocation state with 3 processes and 2 resource types: Allocation (A, B) Max (A, B) Available (A, B) P0 1, 0 7, 5 3, 3 P1 2, 0 3, 2 P2 3, 0 9, 0 Compute the Need matrix and determine whether the system is in a safe state using the Banker's Algorithm. If safe, state the safe sequence.	**2**
6.	Explain the concept of Thrashing in virtual memory systems. Draw a graph showing the relationship between the degree of multiprogramming and CPU utilisation that illustrates the onset of thrashing. State two techniques used by operating systems to detect and recover from thrashing.	**2**
	SECTION -B	**6*3=18**

		marks
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain the concept of Threads in operating systems. Discuss the Thread Model covering user-level threads and kernel-level threads — compare both with respect to implementation, scheduling, blocking behaviour, and performance overhead. Also explain the three classic multithreading models: *Many-to-One*, *One-to-One*, and *Many-to-Many*, with a neat diagram for each. State at least three practical benefits of using threads over processes in modern application development (e.g., web servers, database systems). **(b)** Consider the following five processes arriving at a CPU scheduler: Process Arrival Time (ms) Burst Time (ms) Priority (Lower = Higher) P1 0 10 3 P2 1 5 1 P3 2 8 4 P4 3 2 2 P5 4 4 2 (i) Apply Preemptive Priority Scheduling and draw the Gantt chart. (ii) Apply Round Robin Scheduling with Time Quantum = 3 ms and draw the Gantt chart. (iii) For both algorithms, calculate the Average Waiting Time and Average Turnaround Time. (iv) Compare the two results and state which algorithm is more suitable for an interactive time-sharing system and why.	**6**
8.	**(a)** Explain Deadlock in operating systems. Discuss: *(i)* The four necessary conditions for deadlock (Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait) with an intuitive real-world analogy for each. *(ii)* The concept of a Resource Allocation Graph (RAG) — explain how a cycle in the RAG indicates deadlock when each resource has a single instance. Draw a sample RAG with 3 processes (P1, P2, P3) and 3 resources (R1, R2, R3) that demonstrates a deadlock condition. *(iii)* Briefly compare the three strategies — Deadlock Prevention, Deadlock Avoidance, and Deadlock Detection — in terms of approach, overhead, and applicability. **(b)** Consider a system with the following page reference string and a memory frame capacity of 3 frames (initially empty): Reference String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 (i) Apply the FIFO Page Replacement algorithm. Show the frame state at each step and count the total number of page faults. (ii) Apply the Optimal Page Replacement algorithm on the same string. Show the frame state at each step and count the total page faults. (iii) Apply the LRU (Least Recently Used) page replacement algorithm and count page faults. (iv) Tabulate the page fault counts for all three algorithms and comment on which performs best and why. Also explain Belady's Anomaly and state which algorithm is immune to it.	**6**

9.	**(a)** Explain Disk Scheduling in secondary memory management. Describe the following disk scheduling algorithms with their working principle and one advantage and one disadvantage each: *(i)* FCFS (First Come First Served) *(ii)* SSTF (Shortest Seek Time First) *(iii)* SCAN (Elevator Algorithm) *(iv)* C-SCAN (Circular SCAN) For each algorithm, apply it to the following disk request queue and calculate the total head movement (in cylinders). Assume initial head position = 53, disk cylinders range from 0–199, and head is moving towards higher cylinders. Request Queue: 98, 183, 37, 122, 14, 124, 65, 67 **(b)** Write a structured case study of the UNIX Operating System covering the following aspects: *(i)* Architecture of UNIX — draw a neat layered diagram showing Hardware, Kernel, Shell, and User Programs. Briefly explain the role of each layer. *(ii)* UNIX Process Management — explain how processes are created using fork() and exec() system calls. State what a zombie process and an orphan process are. *(iii)* UNIX File System — explain the inode structure, the directory hierarchy (/, /home, /bin, /etc, /tmp), and the concept of hard links vs. symbolic links. *(iv)* UNIX Memory Management — briefly describe how UNIX handles virtual memory and swapping. *(v)* State three distinguishing features of UNIX that influenced the design of modern operating systems such as Linux and macOS.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **PGRT-03N**	Course Name: **Research Tools and Practices**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Write down the types and methods of data collection.	**2**
2.	What are the different types of sample design?	**2**
3.	What do you understand by data presentation?	**2**
4.	Briefly describe errors in hypothesis testing.	**2**
5.	Discuss the application of ICT in research.	**2**
6.	Discuss the application of MS office in research.	**2**
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	Discuss sample designing detail. What are the characteristics of a good sample design?	**6**
8.	What is research hypothesis and formulation of hypothesis? Discuss the concept of hypothesis testing.	**6**
9.	What do you understand by Data classification and tabulation?	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 117N**	Course Name: **Soft Computing**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Distinguish between Hard Computing and Soft Computing paradigms. State the three principal constituents of soft computing (Fuzzy Logic, Neural Networks, Genetic Algorithms) and for each, identify one real-world problem domain where it is preferred over traditional AI techniques. Also state two key limitations of conventional AI that motivated the development of soft computing.	**2**
2.	A spam filter uses Bayesian reasoning to classify emails. Given the following probabilities: • P(Spam) = 0.40, P(Not Spam) = 0.60 • P(word "offer" | Spam) = 0.80, P(word "offer" | Not Spam) = 0.10 Using Bayes' Theorem, calculate the probability that an email containing the word "offer" is spam. Show all steps clearly and interpret the result in the context of the spam filter decision.	**2**
3.	Let Universe of Discourse X = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Two fuzzy sets are defined as: A = "Small Numbers" = {1/1, 0.8/2, 0.6/3, 0.4/4, 0.2/5, 0/6, 0/7, 0/8, 0/9, 0/10} B = "Medium Numbers" = {0/1, 0.2/2, 0.5/3, 0.8/4, 1/5, 0.8/6, 0.5/7, 0.2/8, 0/9, 0/10} Compute: (i) A ∪ B (ii) A ∩ B (iii) Complement of A. Show results in standard fuzzy set notation and briefly explain the set-theoretic operations used.	**2**
4.	Explain the Perceptron Training Algorithm with a neat diagram of a single-layer perceptron having two inputs (x_1, x_2), one bias, and one output. Trace one complete training step using the following values: • Inputs: $x_1 = 1$, $x_2 = 0$; Target output: $t = 1$ • Initial weights: $w_1 = 0.3$, $w_2 = 0.5$; Bias: $b = 0.2$; Learning rate $\eta = 0.1$; Threshold = 0.5 Calculate the net input, actual output, error, and updated weights.	**2**
5.	Consider a Genetic Algorithm applied to maximise the function $f(x) = x^2$ where x is encoded as a 4-bit binary string (x ranges from 0 to 15). Given the initial population: Chromosome Binary String C1 1010 C2 0101 C3 1100 C4 0011 Calculate the decimal value, fitness $f(x) = x^2$, and selection probability for each chromosome. Identify which chromosome has the highest selection probability and explain why fitness-proportionate selection favours it.	**2**

6.	Differentiate between a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) with respect to architecture, type of input data, memory capability, and typical applications. Present your answer as a structured comparison table (at least four attributes) followed by one real-world application example for each network type.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain monotonic vs. non-monotonic reasoning with one example each. Draw a simple Bayesian Network for the scenario: Rain → Wet Grass ← Sprinkler, label the conditional probability tables, and explain how inference flows through the network. **(b)** A Mamdani Fuzzy Inference System evaluates student performance using inputs Study Hours (0–10) and Attendance (0–100%) with rules: • IF Study Hours is High AND Attendance is Good → Grade is Excellent • IF Study Hours is Low OR Attendance is Poor → Grade is Poor For Study Hours = 7, Attendance = 75%: (i) Define triangular membership functions and compute membership degrees. (ii) Fire rules and compute firing strengths. (iii) Defuzzify using the Centroid Method to obtain a crisp grade.	**6**
8.	**(a)** Draw an MLP with 2 inputs, 2 hidden neurons, and 1 output. Explain the forward pass using the sigmoid function. Derive the weight update rule for the output layer using gradient descent. State the momentum update equation and explain how momentum helps avoid local minima. List three limitations of EBPA. **(b)** For the network below, perform one complete forward and backward pass: • Inputs: $x_1 = 0.6$, $x_2 = 0.9$ • Hidden weights: $w_{11} = 0.4$, $w_{12} = 0.7$, $w_{21} = 0.3$, $w_{22} = 0.5$; Biases: $b_1 = 0.1$, $b_2 = 0.2$ • Output weights: $v_1 = 0.6$, $v_2 = 0.8$; Bias: $b_3 = 0.3$ • Target: $t = 1.0$; $\eta = 0.5$; Activation: $\sigma(x) = 1/(1 + e^{-x})$ *(i)* Compute h_1, h_2, and final output $\hat{y}$. *(ii)* Calculate MSE = $\frac{1}{2}(t - \hat{y})^2$. *(iii)* Compute the error gradient at the output layer (δ_output) and update output weights v_1 and v_2. *(iv)* Computethe error gradients at the hidden layer δ_h_1 and update w_{11}. Show all steps.	**6**
9.	**(a)** Draw the GA Generational Cycle flowchart. Illustrate single-point, two-point, and uniform crossover using 8-bit chromosomes. Explain bit-flip mutation and inversion operator. Compare GA with gradient descent and exhaustive search across four attributes: search strategy, derivative requirement, parallelism, and handling of multimodal functions. **(b)** Apply GA to the 0/1 Knapsack Problem (capacity = 10 kg): Item Weight (kg) Value (₹) I1 2 6 I2 3 10 I3 4 12 I4 5 13 I5 1 4	**6**

	Initial population (5-bit strings): C1 = 11001, C2 = 01100, C3 = 10011, C4 = 00110. Fitness = total value if total weight ≤ 10, else 0. (i) Evaluate fitness of all chromosomes. (ii) Compute roulette wheel selection probabilities. (iii) Apply single-point crossover at position 3 between C1–C2 and C3–C4. (iv) Apply bit-flip mutation at position 2 on offspring O1 and re-evaluate fitness. (v) Identify the best solution and list selected items, total weight, and total value.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 121D**	Course Name: **Dissertation with viva voce**

General Note:

1. Students are required to select **any one** project title from the above list, based on their individual interest, skill level, and available resources, in consultation with their project supervisor.
2. All datasets, tools, technologies, frameworks, and software mentioned above are suggested examples only. Students may use alternative or newer tools, as well as publicly available or self-created datasets, based on project needs. The final choice should be made in consultation with the project supervisor.
3. Guidelines for preparing Research Project/Dissertation is available at link: http://uprtou.ac.in/upload_pdf/01_02_2023_Guidelines_fo_Project_Lit_Survey_Dissertation.pdf

Project No.	Project Details
1.	**Title:** AI-Powered Early Detection of Diabetic Retinopathy Using Convolutional Neural Networks **Objective:** Develop a deep learning-based diagnostic support system that analyses retinal fundus images to detect diabetic retinopathy at an early stage using a Convolutional Neural Network. The model classifies images into severity grades (No DR, Mild, Moderate, Severe, Proliferative) and provides instant screening results through a web interface. This addresses the critical shortage of ophthalmologists in rural India, where diabetic retinopathy often goes undiagnosed until vision loss occurs. **Technologies / Tools:** • Deep Learning: TensorFlow/Keras, CNN (custom or transfer learning with InceptionV3/EfficientNet) • Dataset: APTOS 2019 Blindness Detection (Kaggle) / EyePACS • Image Processing: OpenCV, PIL (preprocessing, contrast enhancement) • Backend: Python, Flask • Frontend: HTML, Bootstrap • Evaluation: Accuracy, Confusion Matrix, Cohen's Kappa Score **Real-World Application:** Deployable in rural primary health centres and telemedicine platforms (e.g., eSanjeevani) where ophthalmology specialists are unavailable. Supports India's National Programme for Control of Blindness by enabling mass screening camps with instant AI-assisted results.
2.	**Title:** Fake News Detection on Social Media Using Natural Language Processing and Machine Learning **Objective:** Build a text classification system that distinguishes real news from fake news using NLP preprocessing techniques and a machine learning/deep learning classifier trained on labelled news datasets. The system extracts linguistic features (sentiment, subjectivity, source credibility indicators) and applies TF-IDF or word embeddings before classification. A browser extension or web dashboard allows users to paste a news snippet and receive an instant credibility score. **Technologies / Tools:** • NLP: Python, NLTK, spaCy, TF-IDF, Word2Vec/GloVe embeddings • ML/DL: scikit-learn (Logistic Regression, SVM), LSTM (TensorFlow/Keras), or fine-tuned BERT • Dataset: LIAR dataset, Kaggle Fake and Real News dataset • Backend: Flask/FastAPI • Frontend: HTML, Bootstrap, Chrome Extension (optional) • Evaluation: Precision, Recall, F1-Score, ROC-AUC

	Real-World Application: Directly applicable to social media platforms, WhatsApp forward verification tools, and government fact-checking initiatives (e.g., PIB Fact Check). Highly relevant given the scale of misinformation during Indian elections and public health crises.
3.	**Title:** Sentiment Analysis of Customer Reviews for E-Commerce Product Recommendation **Objective:** Develop a sentiment analysis pipeline that processes customer product reviews from e-commerce platforms to classify sentiment (positive, negative, neutral) and generate a product recommendation score. The system combines aspect-based sentiment analysis (identifying sentiment toward specific product features like "battery," "camera," "price") with a ranking algorithm to recommend top-rated products within a category. **Technologies / Tools:** • NLP/ML: Python, NLTK, TextBlob, scikit-learn, VADER Sentiment • Deep Learning (optional): LSTM/BERT for aspect-based sentiment • Dataset: Amazon Product Review Dataset / Flipkart scraped reviews • Web Scraping: BeautifulSoup, Selenium (if live data needed) • Backend: Flask/Django • Frontend: HTML, Bootstrap, Chart.js (visualisation of sentiment trends) **Real-World Application:** Directly applicable to e-commerce platforms (Amazon, Flipkart, Myntra) for improving product ranking algorithms and to brand manufacturers for monitoring customer satisfaction and identifying product defects through review mining.
4.	**Title:** Smart Traffic Management System Using Computer Vision and IoT **Objective:** Implement a real-time traffic density estimation and adaptive signal control system using computer vision object detection (YOLO) on live or recorded traffic camera feeds. The system counts vehicles per lane, estimates congestion levels, and dynamically recommends signal timing adjustments to reduce average wait time, simulated through a dashboard interface. **Technologies / Tools:** • Computer Vision: YOLOv8, OpenCV, Python • IoT (optional): Raspberry Pi with camera module for edge deployment • Backend: Flask/FastAPI • Frontend: HTML, Bootstrap, live video feed rendering • Simulation: SUMO (Simulation of Urban Mobility) for signal timing simulation • Database: SQLite (traffic log storage) **Real-World Application:** Aligns with India's Smart City Mission — deployable at busy urban intersections in cities like Bengaluru, Pune, and Delhi to reduce traffic congestion, fuel wastage, and pollution through adaptive signal timing instead of static timers. **Examiner Appeal:** Combines Computer Vision, IoT, and traffic simulation — a multidisciplinary project with strong visual demonstration potential (live vehicle counting on camera feed). CO1 alignment (environment and society — reducing urban emissions) is a compelling dissertation narrative.
5.	**Title:** Crop Disease Detection and Advisory System for Farmers Using Deep Learning and Mobile Application **Objective:** Build a deep learning-based image classification system using transfer learning (MobileNetV2) that identifies plant leaf diseases from smartphone photographs across multiple crop types. The system is deployed as a lightweight, mobile-responsive web application that works on low-bandwidth rural networks and provides treatment advisory in simple language. **Technologies / Tools:** • Deep Learning: TensorFlow Lite, Keras, MobileNetV2 (Transfer Learning) • Dataset: PlantVillage dataset (Kaggle) • Backend: Python, Flask • Frontend: HTML, Bootstrap (mobile-first responsive design), PWA (Progressive Web App) for offline access • Image Processing: OpenCV

	• Optional: Multilingual text-to-speech advisory (gTTS) for low-literacy farmers **Real-World Application:** Directly supports India's agricultural sector, where over 50% of the population depends on farming. Reduces dependency on agricultural extension officers by providing instant, AI-powered disease diagnosis and treatment guidance accessible via basic smartphones.
6.	**Title:** Intrusion Detection System for IoT Networks Using Anomaly-Based Machine Learning **Objective:** Develop a network intrusion detection system tailored for IoT environments that identifies anomalous traffic patterns using unsupervised and supervised machine learning models (e.g. Isolation Forest, Autoencoder, Random Forest). The system is trained and evaluated on the CICIDS2017 or NSL-KDD dataset and simulates real-time detection of common IoT attacks such as DDoS, port scanning, and botnet traffic. **Technologies / Tools:** • ML/DL: scikit-learn, TensorFlow/Keras (Autoencoder for anomaly detection) • Dataset: NSL-KDD, CICIDS2017, or Bot-IoT dataset • Network Simulation: Wireshark (packet capture reference), Scapy (traffic generation) • Backend: Python, Flask • Frontend: HTML, Bootstrap, real-time alert dashboard • Evaluation: Precision, Recall, F1-Score, False Positive Rate **Real-World Application:** Critical for securing smart home devices, industrial IoT (IIoT) systems, and connected infrastructure increasingly deployed in India's smart city and Digital India initiatives, where traditional signature-based firewalls are ineffective against evolving IoT-specific attacks.
7.	**Title:** Mental Health Chatbot Using Deep Learning and Sentiment Analysis **Objective:** Build a conversational AI chatbot that engages users in supportive dialogue, analyses emotional tone using sentiment analysis, and detects signs of distress through text patterns using an CNN/RNN/LSTM -based intent classification model. The chatbot provides empathetic responses, coping strategies, and directs users to professional mental health resources when distress indicators are detected. **Technologies / Tools:** • NLP/DL: Python, TensorFlow/Keras (LSTM), Rasa or Dialogflow (conversational framework), Hugging Face Transformers (optional) • Sentiment Analysis: VADER, TextBlob • Dataset: Mental health FAQ datasets, EmpatheticDialogues dataset • Backend: Flask/FastAPI • Frontend: HTML, CSS, JavaScript (chat interface) • Database: SQLite (conversation logging with privacy safeguards) **Real-World Application:** Addresses the growing mental health crisis among Indian youth and university students, providing an accessible, stigma-free first point of contact. Complements initiatives like India's Tele-MANAS mental health helpline by offering 24/7 preliminary text-based support.
8.	**Title:** Predicting Student Academic Performance Using Machine Learning for Early Intervention **Objective:** Develop a predictive analytics system that uses historical academic and behavioural data (attendance, assignment scores, engagement metrics) to identify students at risk of academic failure using classification algorithms (Decision Tree, Random Forest, Logistic Regression). The system provides a faculty dashboard with risk scores and visual analytics enabling early, targeted intervention. **Technologies / Tools:** • ML: scikit-learn (Decision Tree, Random Forest, Logistic Regression, XGBoost) • Dataset: UCI Student Performance dataset / custom institutional data (anonymised) • Backend: Flask/Django • Frontend: HTML, Bootstrap, Chart.js/Plotly (risk dashboard)

	• Database: MySQL/SQLite • Evaluation: Accuracy, Precision, Recall, ROC-AUC, Feature Importance analysis **Real-World Application:** Directly deployable within university academic monitoring systems to reduce dropout rates by enabling faculty and mentors to identify struggling students weeks before final examinations, allowing timely counselling and academic support interventions.
9.	**Title:** Handwritten Digit Recognition Using Convolutional Neural Network with Web Interface **Objective:** Implement a Convolutional Neural Network (CNN) trained on the MNIST dataset to recognise handwritten digits with high accuracy. The trained model is integrated into a web application where users can draw a digit on a canvas and receive instant prediction. The project explores deep learning architecture design, training optimisation, and model deployment. **Technologies / Tools:** • Deep Learning: TensorFlow / Keras (CNN architecture) • Dataset: MNIST (60,000 training images) • Backend: Python, Flask • Frontend: HTML5 Canvas, JavaScript, Bootstrap • Model Saving: HDF5 format (.h5) • Visualisation: Training accuracy/loss curves (Matplotlib) **Real-World Application:** Handwriting recognition is used in postal automation (pin code reading), bank cheque processing, digitisation of historical documents, and form data entry automation. Directly applicable in Indian postal and banking sectors.
10.	**Title:** Object Detection System for Campus Security Using Pre-Trained YOLO Model on Static Images **Objective:** Develop a web-based application that allows users to upload static campus images (e.g., photographs of gates, parking areas, corridors) and uses a pre-trained YOLOv8 model to detect and label objects such as people, bicycles, cars, and bags. The system displays the uploaded image with bounding boxes and object labels, along with a count summary of detected objects. This project applies an existing pre-trained model to single images, removing the complexity of real-time video processing. **Technologies / Tools:** • Deep Learning: YOLOv8 (Ultralytics) — pre-trained weights only, no custom training required • Image Processing: OpenCV, PIL, Python • Backend: Flask (image upload and processing route) • Frontend: HTML, Bootstrap, JavaScript (image upload form and result display) • Database: SQLite (optional — log of uploaded image name, detected objects, and timestamp) • Sample Data: Self-captured campus photographs or open-source images (Kaggle, Unsplash, COCO sample images) **Real-World Application:** Represents a simplified version of image-based security auditing tools used to analyse CCTV snapshots, entry-gate photographs, or parking area images for object counting and monitoring. Useful for periodic manual security reviews where continuous live monitoring is not required, such as verifying parking occupancy or checking crowd density from a single photograph.

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 116N**	Course Name: **Computer Graphics**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Distinguish between Raster-Scan and Random-Scan display systems across any four attributes: image drawing method, resolution, cost, and typical application. Present your answer as a comparison table.	**2**
2.	Apply the DDA Line Drawing Algorithm to rasterize a line from P1(2, 2) to P2(5, 4). Compute the slope, x and y increments, and tabulate the calculated and rounded (x, y) pixel positions at each step.	**2**
3.	A line has endpoints A(−1, 2) and B(4, 6) and the clip window is: x_min = 0, x_max = 3, y_min = 0, y_max = 5. Using the Cohen-Sutherland Algorithm, compute the region codes for both endpoints and determine whether the line is accepted, rejected, or needs clipping. State the rule applied.	**2**
4.	A point P(3, 2) is rotated by 90° anticlockwise about the origin. Write the 3×3 homogeneous rotation matrix, apply it to P, and find the transformed coordinates P′. Show all matrix multiplication steps.	**2**
5.	Differentiate between Gouraud Shading and Phong Shading with respect to interpolation technique, specular highlight accuracy, and computational cost. Support with a one-line diagram showing normal vector usage in each method.	**2**
6.	Using the Midpoint Circle Drawing Algorithm for a circle centred at origin with radius r = 6, compute the first three pixel positions starting from (0, 6). Show the initial decision parameter p and its update at each step.	**2**
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain Bresenham's Line Drawing Algorithm. Derive the decision parameter p_k showing how integer arithmetic replaces floating-point computation. Write the pseudocode for the case where slope m is between 0 and 1. **(b)** Rasterize the line from P1(1, 1) to P2(7, 4) using Bresenham's Line Drawing Algorithm. *(i)* Compute Δx, Δy, initial decision parameter p_1, and increment constants. *(ii)* Tabulate pixel positions (x, y) and p_k at each step until endpoint is reached. *(iii)* Sketch the rasterized pixels on a clearly labelled grid.	**6**

8.	**(a)** Explain the following 2-D Geometric Transformations using homogeneous coordinates: (i) Derive the 3×3 matrices for Translation (Tx, Ty) and Scaling (Sx, Sy). (ii) Explain Rotation about an arbitrary point (h, k) as a three-step composite transformation. Write the sequence of steps and the resulting combined matrix form. (iii) State why homogeneous coordinates are essential for combining transformations multiplicatively. **(b)** A triangle with vertices A(1, 1), B(3, 1), C(2, 3) undergoes: • Step 1: Scale by Sx = 2, Sy = 2 • Step 2: Translate by Tx = 1, Ty = 2 (i) Write the 3×3 matrices for each transformation. (ii) Derive the composite matrix M = T_translation × T_scale. (iii) Apply M to vertices A, B, and C and tabulate the original and transformed coordinates.	**6**
9.	**(a)** Answer the following on Bezier Curves and Visible Surface Detection: *(i)* Write the parametric equation of a Cubic Bezier Curve using Bernstein polynomials for $t \in [0, 1]$. State any three properties of Bezier curves with one-line explanations. *(ii)* Explain the Z-Buffer (Depth Buffer) Algorithm for visible surface detection in pseudocode form. State one advantage and one limitation of this method. **(b)** Answer the following on Illumination and Ray Tracing: *(i)* Write the Phong Illumination Model equation: $I = I_a \cdot k_a + I_d \cdot k_d \cdot (\hat{L} \cdot \hat{N}) + I_s \cdot k_s \cdot (\hat{R} \cdot \hat{V})^n$ Define any four terms from the equation and explain the visual contribution of each reflection component (Ambient, Diffuse, Specular) in one sentence each. *(ii)* Draw a neat labelled diagram of the Basic Ray Tracing Algorithm showing the eye/camera, primary ray, shadow ray, and reflected ray. State one advantage of ray tracing over the Z-buffer method.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 114N**	Course Name: **Multimedia Technology**

SECTION -A		**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Define Multimedia and list any four core elements with one real-world application for each.	**2**
2.	A raw audio clip has: Sampling Rate = 44,100 Hz, Bit Depth = 16 bits, Channels = 2 (Stereo), Duration = 2 minutes. Calculate the uncompressed file size in MB. Show all steps.	**2**
3.	Distinguish between Real-Time Animation and Frame-by-Frame Animation across any three attributes: rendering method, hardware requirement, and typical application.	**2**
4.	Differentiate between MIDI and Digital Audio across four attributes: data representation, file size, editability, and hardware dependency.	**2**
5.	A digital video has: Resolution = 1280×720, Frame Rate = 25 fps, Colour Depth = 24 bits, Duration = 1 minute. Calculate the uncompressed video size in MB. Show all steps. State one reason why video compression is necessary.	**2**
6.	What is Video Conferencing? State any three technical requirements for a video conferencing system and name two protocols that support real-time audio-video transmission over the Internet.	**2**
SECTION -B		**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain the architecture of a Multimedia System covering: *(i)* Any three core elements of multimedia — define each and give one example of its use in an e-learning application. *(ii)* Any three multimedia file formats (e.g., MP3, AVI, JPEG) — for each, state the data type stored and one typical use case. *(iii)* State the role of RTP and RTSP protocols in multimedia streaming in one sentence each. **(b)** Explain Audio Compression in multimedia: *(i)* Draw a simple block diagram of the ADPCM encoder showing: Input PCM Signal → Difference Calculator → Adaptive Quantizer → Compressed Output. Explain in two sentences how ADPCM reduces file size compared to standard PCM. *(ii)* Compare FM Synthesis and Wavetable Synthesis across three attributes: sound quality, memory usage, and realism. State which is used in modern sound cards. *(iii)* Define Speech Recognition and state two real-world challenges it faces.	**6**

8.	**(a)** Explain Digital Video Capture and Processing: (i) Define the following terms in one sentence each: Frame Rate, Resolution, Aspect Ratio, Colour Depth. (ii) Distinguish between Frame Grabbing and Full Motion Video across three attributes: capture method, frame rate, and application. (iii) Name any two video compression standards (e.g., MPEG-2, H.264) and state one use case for each. **(b)** Explain 3-D Animation Techniques in multimedia: (i) Define a 3-D Mesh Model and explain how polygons, vertices, and edges represent a 3-D object. Give one example from game development. (ii) Explain any three of the following rendering concepts in two sentences each with a small sketch where appropriate: • Texture Mapping • Shadow Generation • Anti-Aliasing • Transparency (iii) State one difference between Automatic Motion Control and Manual Motion Control in 3-D character animation.	**6**
9.	**(a)** Explain Multimedia Authoring Tools: (i) Define a Multimedia Authoring Tool and name any two authoring tools identifying the authoring model each uses (icon-based, timeline-based, or card-based). (ii) Explain the role of any two of the following editors in one paragraph each: • Project Editor • Topic Editor • Hot-Spot Editor (iii) List four steps involved in developing a multimedia title using an authoring tool — from planning to delivery. State one advantage of authoring tools over manual programming. **(b)** Explain Multimedia on LANs and the Internet: (i) State two challenges of delivering multimedia content over a LAN — address bandwidth and latency — and name one technique used to manage multimedia traffic. (ii) Compare Fast Modems and High-Speed Digital Networks (e.g., DSL, Fibre) across three attributes: bandwidth, cost, and suitability for video streaming. (iii) Explain any two multimedia Internet applications (e.g., video-on-demand, IP telephony) — for each, state one network requirement and one enabling technology. State one future direction in multimedia networking in one sentence.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 119N**	Course Name: **Information and Network Security**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Explain the CIA Triad (Confidentiality, Integrity, Availability) as the three core characteristics of Information Security. For each characteristic, state one real-world example of a security threat that violates it.	**2**
2.	Encrypt the plaintext "SECURITY" using the Caesar Cipher with a key shift of 4. Show the encryption step for each letter. Then decrypt the resulting ciphertext back to the original plaintext.	**2**
3.	State any four design principles of Block Ciphers as proposed by Claude Shannon (e.g., confusion, diffusion). Explain each principle in one sentence and state how it contributes to the strength of DES.	**2**
4.	Apply the RSA Algorithm using p = 5, q = 11. *(i)* Calculate n and φ(n). *(ii)* Choose public key e = 3. Verify that gcd(e, φ(n)) = 1. *(iii)* Calculate the private key d using the extended Euclidean algorithm. *(iv)* Encrypt the message M = 7 and decrypt the resulting ciphertext. Show all steps.	**2**
5.	Distinguish between Intrusion Detection Systems (IDS) and Firewalls across four attributes: function, placement in network, type of threats addressed, and response mechanism.	**2**
6.	What is Steganography? Distinguish it from Cryptography across three attributes: objective, visibility of hidden data, and typical use case. Give one real-world example of steganography in digital media.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	(**a)** Explain the Model for Network Security and the X.800 Security Framework: *(i)* Draw and explain the Network Security Model showing the four components: Sender, Receiver, Trusted Third Party, and the communication channel. Explain the role of each component in securing a message in transit. *(ii)* Distinguish between Security Attacks, Security Mechanisms, and Security Services as defined in X.800. Classify the following as active or passive attacks with one-line justification: Eavesdropping, Replay Attack, Denial of Service, Traffic Analysis. *(iii)* State any two Security Services from X.800 (e.g., Authentication, Access Control) and for each, name one mechanism that implements it. **(b)** Explain Classical Encryption Techniques:	**6**

	(i) Encrypt the plaintext "HELLO WORLD" using the Vigenère Cipher with keyword "KEY". Show the key repetition, the numerical addition (mod 26), and the final ciphertext letter by letter in a table. *(ii)* Apply the Rail Fence Transposition Cipher with 3 rails to the plaintext "INFORMATION". Draw the rail pattern, read off the ciphertext row by row, and write the final encrypted message. *(iii)* Define Steganography and state one technique used to hide data in a digital image (e.g., LSB substitution). State one advantage of steganography over cryptography in covert communication.	
8.	**(**a) Explain Public Key Cryptography and Key Management: *(i)* Explain the principle of Public Key Cryptography — define the key pair (public key, private key), explain how encryption and decryption work in asymmetric systems, and state two advantages over symmetric key systems. *(ii)* Explain the Diffie-Hellman Key Exchange Protocol with a step-by-step numerical example using: prime p = 23, primitive root g = 5, Alice's private key a = 6, Bob's private key b = 15. Show how both parties arrive at the same shared secret key. State one vulnerability of Diffie-Hellman and how it is addressed. *(iii)* State Fermat's Little Theorem and apply it to verify: 3^4 mod 5. State its relevance in RSA key generation. **(b)** Explain Message Authentication and Digital Signatures: *(i)* Distinguish between a Message Authentication Code (MAC) and a Hash Function across three attributes: key usage, purpose, and example algorithm. State one scenario where MAC is preferred over a simple hash. *(ii)* Explain the working of the SHA-1 Hash Function — state the input block size, output digest size, number of rounds, and the type of operations performed. State one known weakness of SHA-1 that led to its deprecation. *(iii)* Explain the Digital Signature Standard (DSS) — draw a simple diagram showing the signing process (message → hash → sign with private key → signature) and the verification process (signature → verify with public key → compare hash). State one real-world application of digital signatures.	**6**
9.	**(a)** Explain Network Security Applications: (i) Explain the Kerberos Authentication Protocol — draw a diagram showing the three entities: Client, Authentication Server (AS), and Ticket Granting Server (TGS). Describe in three steps how a client obtains a service ticket and authenticates to a server. (ii) Explain PGP (Pretty Good Privacy) for email security — state the four operations PGP performs on a message: Authentication, Confidentiality, Compression, and Email Compatibility. For each, name the algorithm or technique used. (iii) Explain SSL/TLS in one paragraph — state the two sub-protocols (Handshake Protocol and Record Protocol), their roles, and one difference between SSL 3.0 and TLS 1.0. **(b)** Explain Intruders, Malicious Programs, and Firewalls: (i) Classify intruders into three categories (Masquerader, Misfeasor, Clandestine User) and define each in one sentence. State two password selection strategies that reduce the risk of password-based intrusion. (ii) Explain any three types of viruses (e.g., Boot Sector Virus, Macro Virus, Polymorphic Virus) — for each, state how it spreads, what it infects, and one detection approach. State one limitation of signature-based antivirus detection. (iii) Distinguish between the following three types of firewalls across two attributes each — filtering method and limitation: • Packet Filtering Firewall	**6**

	• Stateful Inspection Firewall • Application-Level Gateway (Proxy Firewall) State one recommended firewall configuration for a corporate network and justify your choice in two sentences.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 104N**	Course Name: **Software Engineering**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Compare the Waterfall Model and the Spiral Model of software development across four attributes: development approach, handling of requirements change, risk management, and suitability. Present your answer as a comparison table.	**2**
2.	A software project has the following function point counts before complexity adjustment: Component — Count — Weighting Factor External Inputs — 10 — 4 External Outputs — 8 — 5 External Inquiries — 6 — 4 Internal Files — 4 — 10 External Interfaces — 3 — 7 Calculate the Unadjusted Function Point (UFP) count. If the Value Adjustment Factor (VAF) = 1.15, calculate the Adjusted Function Point count. Show all steps.	**2**
3.	Define Cohesion and Coupling in software design. State the most desirable level of cohesion and the least desirable level of coupling. Give one example each of a highly cohesive module and a tightly coupled module from a real software system.	**2**
4.	Distinguish between Validation Testing and Verification Testing in one sentence each. Then compare Alpha Testing and Beta Testing across three attributes: who performs it, where it is conducted, and its purpose.	**2**
5.	What is Software Reengineering? Distinguish between Reverse Engineering and Forward Engineering across three attributes: direction of process, objective, and output produced. Give one real-world scenario where reverse engineering is applied.	**2**
6.	State any four characteristics of good software (e.g., reliability, maintainability, portability, efficiency). For each characteristic, give one metric or indicator used to measure it in practice.	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**
7.	**(a)** Explain Software Project Planning and Risk Management: (i) Explain the COCOMO Model (Constructive Cost Model) — state the three modes of development (Organic, Semi-detached, Embedded) and write the effort estimation formula for the Basic COCOMO model: E = a × (KLOC)^b. For a project of size 50 KLOC in Organic mode (a = 2.4, b = 1.05), calculate the estimated effort in Person-	**6**

	Months. Show all steps. (ii) Explain Risk Identification and Risk Projection in software project management. Classify the following risks as either project, technical, or business risks with one-line justification: • Frequent change in requirements • New algorithm may not perform as expected • Competitor releases a similar product earlier • Key developer leaves the team (iii) State the difference between Problem-Based Estimation and Process-Based Estimation in two sentences each. **(b)** Explain Software Metrics: (i) Distinguish between Size-Oriented Metrics and Function-Oriented Metrics across three attributes: unit of measurement, what is measured, and ease of collection. Give one example of each from a real project context. (ii) A project team records the following data: Project KLOC Effort (Person-Months) Defects P1 12 24 36 P2 27 54 86 P3 8 16 24 For each project, calculate: (a) Productivity (KLOC/Person-Month) (b) Defect density (Defects/KLOC). Tabulate all results and identify the most productive project and the project with the lowest defect density. (iii) Define Software Metric Indicator and state how it differs from a raw measure.	
8.	**(**a) Explain Software Quality Assurance and Configuration Management: *(i)* Distinguish between Quality Control and Quality Assurance in one sentence each. Explain the concept of Cost of Quality — define Prevention Cost, Appraisal Cost, and Failure Cost with one example each. *(ii)* Explain Formal Technical Review (FTR) — state its objectives, the roles involved (Producer, Reviewer, Review Leader), and any three guidelines that make an FTR effective. *(iii)* Explain the SCM Process covering: Version Control, Change Control, and Configuration Audit — define each in two sentences and state the tool or mechanism commonly used for each (e.g., Git for version control). **(b)** Explain Requirements Analysis and SRS: *(i)* Explain any three Analysis Principles used during requirements elicitation (e.g., the information domain must be represented, models of the system must be built, partitioning). For each principle, state how it helps the analyst understand the system better. *(ii)* Explain the concept of Essential View vs. Implementation View in requirements analysis — state what each represents and why both are needed during system modelling. *(iii)* State any five standard sections of a Software Requirements Specification (SRS) document as per IEEE standards (e.g., Introduction, Overall Description, Specific Requirements). For each section, state in one sentence what it contains. State two characteristics of a good SRS.	**6**
9.	**(a)** Explain Software Design Concepts and Principles: (i) Explain the following design concepts in two sentences each with one illustrative example: • Abstraction (procedural and data abstraction) • Modularity (state the formula: M = n modules; explain why too few or too	**6**

	many modules both increase cost) • Information Hiding (state what is hidden and from whom) (ii) Explain Software Architecture and Control Hierarchy — define the terms: depth, width, fan-in, and fan-out in a software structure chart. Draw a small structure chart with 5 modules and identify the fan-in and fan-out of the root module. (iii) Classify the following modules by their level of cohesion (Coincidental, Logical, Temporal, Procedural, Communicational, Sequential, or Functional) with one-line justification: • A module that reads input, validates it, and returns the result • A module that performs all error-handling tasks grouped together • A module that initialises all variables at system startup **(b)** Explain Software Testing Strategies: (i) Distinguish between Top-Down Integration Testing and Bottom-Up Integration Testing across three attributes: order of module integration, use of stubs/drivers, and suitability. State one advantage of each approach. (ii) Explain the following types of System Testing in two sentences each: • Recovery Testing — with one example from a banking application • Stress Testing — with one example from an e-commerce website during a sale event • Security Testing — state two techniques used (iii) Explain Regression Testing — define it, state when it is performed in the development life cycle, and give one scenario from a software maintenance context where regression testing is essential.	

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 107N**	Course Name: **Theory of Computation**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	State the Chomsky Hierarchy of formal languages. List all four levels (Type 0 to Type 3) in a table showing: grammar type, language class, automaton that recognizes it, and one example language for each level.	**2**
2.	Consider the following NFA over alphabet $\Sigma = \{0, 1\}$ with states {q0, q1, q2}, start state q0, and final state q2: State 0 1 q0 {q0, q1} {q0} q1 φ {q2} q2 φ φ Trace the NFA on input string "010" by showing all possible state paths. State whether the string is accepted or rejected and justify.	**2**
3.	Convert the Regular Expression (a + b)*abb to a Non-Deterministic Finite Automaton (NFA) using Thompson's construction. Draw the NFA with clearly labelled states, transitions, start state, and final state.	**2**
4.	Apply the Pumping Lemma for Regular Languages to prove that the language $L = \{a^n b^n \mid n \geq 1\}$ is not regular. Clearly state the pumping length p, choose an appropriate string, and show that no valid decomposition satisfies all three pumping conditions.	**2**
5.	Consider the CFG with productions: $S \rightarrow aSb \mid ab$ *(i)* Derive the string "aaabbb" using leftmost derivation. Show each step. *(ii)* Draw the parse tree for the same string. *(iii)* State the language generated by this grammar in set notation.	**2**
6.	Design a Turing Machine that accepts the language $L = \{a^n b^n \mid n \geq 1\}$. Give the formal definition $(Q, \Sigma, \Gamma, \delta, q0, B, F)$ and draw the transition diagram. Trace the machine on input "aabb" showing each instantaneous description (ID).	**2**
	SECTION -B	**6*3=18 marks**
Q. No.	**Long answer type question (approx. 500 -800 words)**	**Marks**

7.	**(a)** Explain Finite Automata and the equivalence between DFA and NFA: (i) Define a DFA formally as a 5-tuple (Q, Σ, δ, q0, F). Construct a DFA over Σ = {a, b} that accepts all strings ending in "ab". Draw the transition diagram and give the transition table. (ii) Explain ε-NFA — define ε-closure of a state and explain the subset construction method to convert an ε-NFA to a DFA in three steps. State why DFA and NFA are equivalent in terms of language recognition power. (iii) State two limitations of Finite State Machines that prevent them from recognising context-free languages. **(b)** Explain Moore and Mealy Machines and DFA Minimisation: (i) Define a Moore Machine and a Mealy Machine as formal 6-tuples. State one structural difference between them. Draw a Moore Machine with three states over Σ = {0, 1} that outputs 1 whenever the input sequence ends in "01", and outputs 0 otherwise. (ii) Apply the Table Filling Algorithm (Myhill-Nerode) to minimise the following DFA with states {A, B, C, D, E}, alphabet {0, 1}, start state A, final states {C, E}: State 0 1 A B C B D C C B E D D C E B C Show the distinguishability table, identify equivalent state pairs, merge them, and draw the minimised DFA.	**6**
8.	**(a)** Explain Context Free Grammars and Normal Forms: (i) Define a CFG formally as a 4-tuple (V, T, P, S). Consider the following grammar: S → AB | ε A → aA | a B → bB | b State whether this grammar is ambiguous. Derive the string "ab" using both leftmost and rightmost derivations and draw the parse tree. (ii) Convert the following CFG to Chomsky Normal Form (CNF). Show each transformation step clearly — eliminate ε-productions, unit productions, and then convert remaining rules: S → aAB A → aA | ε B → bB | b (iii) State the Greibach Normal Form (GNF) — write its production rule format and state one advantage of GNF in designing Pushdown Automata. **(b)** Explain Closure and Decision Properties of Context Free Languages: (i) State which of the following operations preserve closure of CFLs and which do not. Justify each with one sentence: • Union of two CFLs • Intersection of two CFLs • Concatenation of two CFLs • Complement of a CFL • Intersection of a CFL with a Regular Language (ii) Apply the Pumping Lemma for CFLs to prove that L = $\{a^n b^n c^n \mid n \geq 1\}$ is not context-free. Clearly state the pumping length p, choose the string s = $a^p b^p c^p$, and show that no valid decomposition s = uvwxy satisfies all pumping conditions.	**6**

	(iii) State any two decidable and two undecidable decision problems for CFLs.	
9.	**(a)** Explain Pushdown Automata (PDA): (i) Define a PDA formally as a 7-tuple (Q, Σ, Γ, δ, q0, Z0, F). Distinguish between acceptance by final state and acceptance by empty stack — state one condition under which both are equivalent. (ii) Design a PDA that accepts the language $L = \{wcw^R \mid w \in \{a,b\}^*\}$ (strings of the form w followed by c followed by the reverse of w). Draw the transition diagram with clearly labelled states and stack operations. (iii) State the equivalence between CFG and PDA — write in one paragraph how every CFL has a corresponding PDA and vice versa, and why this equivalence is important in compiler design (e.g., syntax analysis using a push-down stack). **(b)** Explain Turing Machines and Undecidability: (i) Define a Turing Machine formally as a 7-tuple (Q, Σ, Γ, δ, q0, B, F). Explain the following variants of TM in two sentences each: • Multi-tape Turing Machine • Non-Deterministic Turing Machine State whether each variant increases the computational power of the standard TM. (ii) Explain the Halting Problem — state the problem formally, explain why it is undecidable using a proof by contradiction (diagonal argument) in four steps, and state its significance in the theory of computation. (iii) Distinguish between Recursive Languages and Recursively Enumerable Languages across three attributes: TM behaviour on acceptance, TM behaviour on rejection, and closure under complement. State one example language for each class.	**6**

Session: **2026 -27**	Max. Marks: **30**
Program Name:	**Master of Computer Science (M.Sc. CS)**
Course Code: **MCS 120N**	Course Name: **System Software**

	SECTION -A	**2*6=12 marks**
Q. No.	**Short answer type question (approx. 200 -300 words)**	**Marks**
1.	Distinguish between a Compiler, Interpreter, and Assembler across four attributes: input type, translation process, output produced, and execution speed. Present your answer as a comparison table.	**2**
2.	Consider the following assembly language program segment: START 100 LABEL1 LDA NUM1 ADD NUM2 STA RESULT NUM1 WORD 10 NUM2 WORD 20 RESULT RESW 1 END START Perform Pass 1 of a Two-Pass Assembler: construct the Symbol Table by assigning location counter values to each label. Show the Location Counter (LC) value at each line assuming each instruction and WORD occupies 3 bytes and RESW n occupies 3n bytes.	**2**
3.	Explain Macro Definition and Macro Call with a simple example. Write a macro named ADDTWO that adds two registers R1 and R2 and stores the result in R3. Show the macro definition, one macro call, and the expanded assembly code after macro expansion.	**2**
4.	Consider the grammar: $E \rightarrow E + T \mid T$ $T \rightarrow T * F \mid F$ $F \rightarrow (E) \mid id$ Compute FIRST and FOLLOW sets for each non-terminal E, T, and F. Show your working for at least two non-terminals.	**2**
5.	Convert the expression $(a + b) * (c - d)$ into: *(i)* A Three-Address Code sequence using temporary variables t1, t2, t3. *(ii)* A DAG (Directed Acyclic Graph) representation showing nodes for operators and leaves for operands.	**2**
6.	Distinguish between Static Linking and Dynamic Linking across four attributes: when linking occurs, memory usage, flexibility of updates, and example use case. State one advantage of dynamic linking in modern operating systems.	**2**
	SECTION -B	**6*3=18 marks**

Q. No.	Long answer type question (approx. 500 -800 words)	Marks
7.	**(a)** Explain Assembler Design and Macro Processing: (i) Explain the Two-Pass Assembler design — state the specific task performed in each pass. Draw a simple block diagram showing inputs, outputs, and internal data structures (Symbol Table, Opcode Table, Location Counter) for both passes. (ii) Explain Nested Macro Calls — define what happens when a macro calls another macro during expansion. Trace the expansion of the following nested macro sequence in three steps: *MACRO OUTER* *INNER ; calls macro INNER* *MOV R1, R2* *ENDM* *MACRO INNER* *ADD R3, R4* *ENDM* *OUTER ; macro call* Show the final expanded code after full expansion. (iii) State two Advanced Macro Facilities (e.g., conditional macro expansion, macro-time variables) and explain each in one sentence. **(b)** Explain Lexical Analysis: (i) State the role of a Lexical Analyser in a compiler — define the terms: token, lexeme, and pattern. For the statement result = value1 + 25; identify all tokens and their token types in a table. (ii) Explain Input Buffering in lexical analysis — draw the two-buffer scheme showing Buffer 1, Buffer 2, forward pointer, and beginning pointer. Explain in two sentences why double buffering is used instead of reading one character at a time. (iii) Explain how a DFA is used for token recognition — draw a simple DFA that recognises integer literals (one or more digits 0–9) and identifiers (letter followed by letters or digits). Label all states and transitions.	**6**
8.	**(a)** Explain Syntax Analysis and Parsing Techniques: *(i)* Distinguish between Top-Down Parsing and Bottom-Up Parsing across three attributes: direction of parse tree construction, stack usage, and example parser type. State one advantage of each. *(ii)* Explain Predictive Parsing (LL Parsing) — state the two conditions a grammar must satisfy to be LL(1): no left recursion and no left factoring. Show how to eliminate left recursion from the following grammar: $E \rightarrow E + T \mid T$ Write the transformed grammar after left recursion removal. *(iii)* Explain Operator Precedence Parsing — define the three operator precedence relations ($<$, $\doteq$, $>$) and state how they are used to shift or reduce during bottom-up parsing. Give one example showing precedence between operators + and *. **(b)** Explain Code Generation and Code Optimisation: *(i)* Explain Syntax-Directed Translation into Three-Address Code — define a translation scheme using semantic rules for the production $E \rightarrow E_1 + E_2$. Show the three-address code generated for the expression a = b * c + b * c before and after applying the Common Subexpression Elimination optimisation. State how many operations are saved.	**6**

	(ii) Classify Code Optimisation techniques into machine-dependent and machine-independent. Give one example of each type. Explain Dead Code Elimination and Constant Folding in two sentences each with a before-and-after code snippet. *(iii)* Distinguish between a Pure Interpreter and an Impure Interpreter in two sentences each. State one programming language that traditionally uses each approach.	
9.	**(a)** Explain Loaders and Linkers: *(i)* Explain the design of an Absolute Loader — state its function, draw a simple flowchart showing the steps: Read Header Record → Load Text Records → Jump to Starting Address. State one limitation of an absolute loader compared to a relocating loader. *(ii)* Explain Program Relocation and Program Linking — define a relocation bit map and explain how the loader uses it to adjust addresses when a program is loaded at a different memory address than its origin. Give one example showing address adjustment for a two-module program. *(iii)* Distinguish between a Linkage Editor and a Dynamic Linker across three attributes: time of linking, memory usage, and handling of shared libraries. State one advantage of dynamic linking in the context of the MS-DOS Linker. **(b)** Explain UNIX Device Drivers: *(i)* Define a Device Driver and state its role as an interface between the operating system kernel and hardware. Draw a simple layered diagram showing: User Process → System Call Interface → Device Driver → Hardware Device. Explain the role of each layer in two sentences. *(ii)* Distinguish between a Character Device Driver and a Block Device Driver across four attributes: data transfer unit, buffering, example device, and access method. State which type is used for a hard disk and which for a keyboard. *(iii)* Explain the UNIX Device Driver Installation process — describe in three steps how a new device driver is added to a UNIX system: writing the driver, adding entry points to the device switch table, and rebuilding the kernel. State one consequence of a faulty device driver on system stability.	**6**