SDGI GLOBAL UNIVERSITY

PROGRAMMESTRUCTURE

SCHOOLOFCOMPUTER APPLICATIONS

Scheme & Syllabus

of

MasterofComputerApplications(MCA) 2

Years Full time program

Specialization-ArtificialIntelligenceandDataScience

Academic Program

W.E.F 2025

SDGI Global University, Ghaziabad(U P)

VisionoftheUniversity

To be recognized as an Institution of excellence, facilitating learning, fostering creativity, knowledge creation, innovations, consultancy and leadership in multiple areas to build a conscious community that will positively impact living beings for a sustainable future.

MissionoftheUniversity

1. To Create conducive environment for an interactive and application oriented experiential learning making the Institute a preferred destination for work and study.
2. To Foster creativity, research and innovation orientation in students and faculty in basic and applied areas in all of its disciplines, provide cost effective solutions and nurture entrepreneurial capabilities to accelerate growth.
3. To act as a catalyst in social change by developing academic, social, political, technological, scientific, industrial and business leadership in the spirit "Think Globally and Act Locally"; by providing ample opportunities to develop team spirit, sportsmanship and love for culture and national heritage.

Core Values

1. **Integrity**
2. **Honesty**
3. **Transparency**
4. **Empathy**

School of Computer Applications

Vision of School

To be a premier institution in computing sciences, recognized for pioneering research, transformative education, and impactful contributions to society, shaping the future of technology and driving positive.

Mission of School

To empower students with cutting-edge knowledge and skills in computing sciences, foster a culture of innovation, and prepare them to address the challenges of a rapidly evolving digital world through rigorous academics, experiential learning, and interdisciplinary collaboration.

CoreValues

1. Excellence
2. Innovation
3. Sustainability
4. Global Perspective

Program Educational Objectives (PEOs)

1. Technical Excellence: Prepare graduates with a strong foundation in computing principles, advanced technical skills, and the ability to apply these skills to solve complex problems in various domains of computing.

2. Professional Development: Equip graduates with the necessary skills to succeed in professional careers, including software development, systems analysis, and IT management, while fostering lifelong learning and adaptability to emerging technologies**.**

3. Research and Innovation: Encourage graduates to engage in research and innovation, contributing to the advancement of technology and knowledge in the field of computer science.

4. Ethical and Social Responsibility: Instill a sense of professional ethics, responsibility, and awareness of the social and environmental impact of technology, ensuring that graduates make informed and ethical decisions in their careers**.**

5. Communication and Teamwork: Develop graduates' abilities to communicate effectively and work collaboratively in multidisciplinary teams, enhancing their effectiveness in professional and academic environments.

Program Outcomes (POs)

1. Knowledge and Understanding: Demonstrate a comprehensive understanding of computing fundamentals, including programming, software engineering, database management, and network systems.

2. Problem-Solving Skills: Apply analytical and problem-solving techniques to design, develop, and implement solutions for complex computing problems.

3. Research and Development: Engage in research and development activities, including the ability to conduct experiments, analyze data, and contribute to technological advancements.

4. Professional Ethics and Responsibility: Adhere to professional ethics and legal standards in the practice of computing, with an understanding of the broader impact of technology on society**.**

5. Communication and Teamwork: Exhibit effective communication skills and the ability to collaborate with peers, stakeholders, and clients in professional and academic settings.

Program Specific Outcomes (PSOs)

1. Advanced Computing Techniques: Demonstrate proficiency in advanced computing techniques, including the development and implementation of complex software systems, data analytics, and machine learning.

2. Software Development Lifecycle: Apply knowledge of the software development lifecycle to manage projects, from requirements gathering and design to implementation, testing, and maintenance.

3. Emerging Technologies: Stay updated with emerging technologies and industry trends, including cloud computing, cybersecurity, and artificial intelligence, and apply this knowledge to solve real-world problems.

4. Research and Innovation Skills: Develop and apply research skills to contribute to innovative solutions in computing, including the ability to design and conduct experiments, analyze results, and publish findings.

5. Professional and Ethical Practice: Exhibit professional conduct and ethical practices in all aspects of computing, including adherence to industry standards, legal regulations, and societal responsibilities.

These PEOs, POs, and PSOs align with UGC norms by ensuring that the MCA program provides a well-rounded education that prepares students for various aspects of their professional and academic careers.

Multiple Exit and Multiple Entry Rule:

According to the UGC (University Grants Commission) guidelines for a PG program with multiple entry and exit points, the credit system typically includes:

1. Total Credits for a PG Program: Generally ranges from 60 to 120 creditsRegards for a full Master's program, depending on the duration (usually 2 years).

2. Credits for Different Entry/Exit Points.

Exit after 1 Year (or 30-40 credits): Often allows students to earn a Postgraduate Diploma or a similar qualification.

Exit after 2 Years (or 60-80 credits): Typically results in the completion of the full Master's degree.

Complete the Program (or 90-120 credits): This might be for programs that have extended components or specializations.

These credits are structured to ensure that students can enter, exit, or re-enter the program at different stages while still gaining recognized qualifications. The exact credit requirements can vary based on the specific UGC guidelines adopted by the institution.

TotalCredits forthe Batch 2024= 81Credits

SEMESTERWISECREDITS				CREDITS
I	II	III	IV	Total
19	21	23	18	81

CC - Core Course
OE - Open Elective/ Multidisciplinary
AE - Ability Enhancement Compulsory Course
VAC - Value Added Course
SE - Skill Enhancement
DSC-Discipline specific course

AI-DSSpecializationsSubjects(MCA)							
Semester	Code	CourseName	L (Hr.)	T (Hr.)	P (Hr.)	Credits	Type
I	M020224102	Introduction to Artificial Intelligence Machine LearningandDataScience	4	0	2	5	DSC (Minor)
II	M020224205	Python for Artificial IntelligenceandData Science	4	0	2	5	DSC (Minor)
II	M020224204	AdvanceDataScience	4	0	2	5	DSC (Minor)
III	M020224303	Advance Linear Algebra ProbabilityandStatistics for Machine Learning	4	0	0	4	DSC (Minor)
III	M020224302	Deep Learning	4	0	2	5	DSC Minor
Total			20	0	8	24	

Bridge Course(1st Semester)					
I	MSGUBR2401	Fundamental of Information Technology	30 Hrs		Bridge
I	MSGUBR2402	Programming in C	30Hrs		Bridge

SDGI GLOBAL UNIVERSITY, GHAZIABAD (SGU)

Semester–Wise Teaching Scheme PG Program

Programme: MCA with AI - ML

Semester – I

Batch – 2025 – 27

Academic year: 2025 – 26

S. No	Status	Paper Code	Subjects	Study Scheme Lec / Week			Hours	Credits	CIE	ESE	Total	Pass Marks
				L	T	P						
1	CC1	**M020224101**	CompetitiveProgramming (Using C++)	**4**	**0**	**0**	**4**	**4**	**50**	**50**	**100**	**40**
3	CC2	**M020224102**	AdvanceData Structures and Algorithms	**4**	**0**	**0**	**4**	**4**	**50**	**50**	**100**	**40**
4	CC3	**M020224103**	AdvanceOperatingSystem	**4**	**0**	**0**	**4**	**4**	**50**	**50**	**100**	**40**
5	CC Lab	**M020224151**	CompetitiveProgramming (Using C++) Lab	**0**	**0**	**2**	**2**	**1**	**60**	**40**	**100**	**40**
6	CC Lab	**M020224152**	AdvanceDataStructures and Algorithm Lab	**0**	**0**	**2**	**2**	**1**	**60**	**40**	**100**	**40**
7	**CC Lab**	**M020224153**	AdvanceOperatingSystem Lab	**0**	**0**	**2**	**2**	**1**	**60**	**40**	**100**	**40**
7	**DSC-1**	**M020224104**	**IntroductiontoArtificial Intelligence Machine Learning and Data Science (Minor)**	**4**	**0**	**0**	**4**	**4**	**50**	**50**	**100**	**40**
8	DSC Lab	**M020224154**	**IntroductiontoArtificial Intelligence Machine Learning and Data Science Lab (Minor)**	**0**	**0**	**2**	**2**	**1**	**60**	**40**	**100**	**40**
Total				**16**	**0**	**8**	**24**	**20**	**440**	**360**	**800**	**320**

CC - Core Course
OE - Open Elective/ Multidisciplinary
AE - Ability Enhancement Compulsory Course
VAC - Value Added Course
SE - Skill Enhancement
DSC-Discipline specific course

CompetitiveProgramming (Using C++)

Program:MCA-I **L T P C**

Course Code:M020224101 **4 0 0 4**

Course Objective:
Introduces Object Oriented Programming concepts using the C++ language.
Introduces the principles of data abstraction, inheritance and polymorphism
Introduces the principles of virtual functions and polymorphism
Introduces handling formatted I/O and unformatted I/O
Introduces exception handling

CO	Course Outcome	Bloom's Level
CO1	Introduces Object Oriented Programming concepts using the C++ language.	**K1**
CO2	Understanding the principles of data abstraction, inheritance, and polymorphism.	**K2**
CO3	Apply the principles of virtual functions and polymorphism.	**K3**
CO4	Understanding the concept of Memory management.	**K4**
CO5	Understand the Exception Handling	**K2**

Detailed Syllabus

Unit I

Object-Oriented Thinking: Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP conceptsAbstraction, Encapsulation, Inheritance and Polymorphism. C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statement- if, switch, while, for, do, break, continue, goto statements. Functions - Scope of variables, Parameter passing, Default arguments, inline functions, Recursive functions, Pointers to functions. Dynamic memory allocation and de-allocation operators-new and delete, Preprocessor directives.

Unit II

C++ Classes and Data Abstraction: Class definition, Class structure, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction, ADT and information hiding.

Unit III

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class. Virtual Functions and Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.

Unit IV

C++ I/O: I/O using C functions, Stream classes hierarchy, Stream I/O, File streams and String streams, Overloading operators, Error handling during file operations, Formatted I/O.

Unit V

Exception Handling: Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

Text Books:

1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India)
2. Turbo C++ by Robert Lafore, Pearson Education

Reference Books:

1. Big C++ - Wiley India

2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India)

CompetitiveProgramming (Using C++) Lab

Program:MCA-I **L T P C**

Course Code:M020224151 **0 0 2 1**

Course Objective:
Students will demonstrate the ability to utilize variables and control flow statements, design functions with parameter passing techniques and default arguments, implement classes with constructors and destructors, develop class hierarchies using inheritance and polymorphism, and manage file handling along with exception management in C++.

CO	Course Outcome	Bloom's Level
CO1	Demonstrate the use of variables, arithmetic operations, and control flow statements inC++.	**K3**
CO2	Illustrate parameter passing techniques and default arguments in function design.	**K2**
CO3	Implement classes with constructors and destructors to model real-world entities.	**K3**
CO4	Create class hierarchies and use virtual functions for dynamic behavior..	**K2**
CO5	Utilize file handling and exception management for robust program development..	**K3**

List of experiments.

1. Write a simple C++ program that declares variables of different data types, performs arithmetic operations, and outputs the results.
2. Create a program that uses `if`, `switch`, `for`, `while`, and `do-while` statements to perform calculations based on user input.
3. Write a program with various functions, demonstrating different parameter passing techniques (by value, by reference) and the use of default arguments.
4. Define a class representing a simple bank account, including member functions for deposit, withdrawal, and checking the balance. Implement constructors and destructors.
5. Create a base class `Shape` and derived classes `Circle` and `Rectangle`. Implement area and perimeter calculations using overridden functions.
6. Use virtual functions to implement a system where derived classes (`Circle`, `Rectangle`) can override a base class function for calculating area.
7. Write a program that reads data from a file, processes it (e.g., calculates the average of numbers), and writes the result to another file. Include error handling for file operations.
8. Implement a program that performs division operations and handles potential division-by-zero exceptions. Demonstrate throwing and catching exceptions with proper error messages
9. Write a program that dynamically allocates memory for an array of integers, allows user input, and displays the array. Ensure proper memory deallocation using `delete`

IntroductiontoArtificial Intelligence Machine Learning and Data Science

Program:MCA-I **L T P C**
Course Code:M020224104 **4 0 0 4**

Course Objective:
Understand the scope, branches, and history of AI.
Learn the fundamentals of machine learning and key algorithms
Gain practical skills in data preprocessing and advanced ML techniques
Explore data science concepts and tools for data analysis
Study advanced AI applications like NLP, big data, and current trends.

CO	Course Outcome	Bloom's Level
CO1	**Understand the history and societal impacts of AI..**	**K2**
CO2	**Apply and evaluate machine learning algorithms**	**K3**
CO3	**Use data preprocessing and optimize models effectively**	**K3**
CO4	**Analyze and visualize data using popular libraries**	**K4**
CO5	**Solve real-world problems using advanced ML and AI techniques**	**K6**

Detailed Syllabus:

Unit-1

IntroductiontoArtificialIntelligence

Definition and scope of Artificial Intelligence, Historical background and milestonesin AI development, Various branches of AI: symbolic AI, statistical AI, etc., ApplicationsofAIindifferentfieldslikehealthcare,finance,gaming,etc.,Ethical considerationsandsocietalimpactofAI

Unit-2

FundamentalsofMachineLearning

Introduction to Machine Learning and its importance, Types of Machine Learning: Supervised, Unsupervised, and Reinforcement Learning, Basic concepts: features, labels, training data, etc., Popular Machine Learning algorithms: Linear Regression, LogisticRegression,DecisionTrees,k-NearestNeighbors,etc.,Evaluationmetricsfor MachineLearningmodels:accuracy,precision,recall,F1-score,etc.

Unit-3

MachineLearning Techniques

Data preprocessing techniques: handling missing data, feature scaling, feature encoding, etc., Model selection and hyperparameter tuning, Cross-validation techniques,Ensemblemethods:Bagging,Boosting,RandomForests,etc.,Introduction todeeplearningandneuralnetworks

Unit-4

IntroductiontoDataScience

What is Data Science and why it is important?, Role of Data Scientist and skills required, Data acquisition: sources of data, data formats, data cleaning, etc., ExploratoryDataAnalysis(EDA):statisticalanalysis,datavisualizationtechniques, Introductiontolibraries/tools:NumPy,Pandas,Matplotlib,Seaborn,etc

Unit-5

AdvancedTopicsandApplications

Advanced Machine Learning techniques: Support Vector Machines (SVM), Neural Networks, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), etc., Natural Language Processing (NLP) and its applications, Introduction to BigDatatechnologies:Hadoop,Spark,etc.,Casestudiesandreal-worldapplications

invariousdomains,Futuretrendsandcareerprospectsin AI,ML,andData Science

Text Books:

1. "Artificial Intelligence:AModernApproach"byStuartRussellandPeterNorvig

2. "IntroductiontoMachineLearning"byEthemAlpaydin

<table>
<tr><td colspan="4">School Name- School of Computer Applications</td></tr>
<tr><td colspan="3">Program- MCA</td><td>Semester-1st</td></tr>
<tr><td colspan="4">Course Name- Advance Data Structures and Algorithms</td></tr>
<tr><td>A.Y 2025-26</td><td>Course Code- M020224102</td><td>Batch- 2023-27</td><td>CIE Marks- 50 (MM)</td></tr>
<tr><td>Total Teaching Hours 50</td><td colspan="2">Total Credits- 4</td><td>ESE Marks- 50 (MM)</td></tr>
<tr><td colspan="3">Type of Course- Theory</td><td>Total Marks 100 (MM)</td></tr>
<tr><td colspan="4">Course Objectives/Course Description

1. Understand fundamental concepts of data structures, abstract data types, and algorithm analysis.
2. Implement linear data structures like arrays, linked lists, stacks, and queues.
3. Apply and manipulate non-linear structures such as trees and graphs in problem solving.
4. Analyze and apply algorithm design techniques like Divide-and-Conquer, Greedy, and Dynamic Programming</td></tr>
<tr><td>UNIT</td><td colspan="2">Topics</td><td>No. of Teaching hours/ (Lecture)</td></tr>
<tr><td>1</td><td colspan="2">Introduction to data structure: Data, Entity, Information, Difference between Data and Information, Data type , Build in data type,Abstractdatatype, Definition of data structures,Types ofData Structures, Introduction to Algorithms: Definition of Algorithms, Difference between algorithm andprograms,propertiesofalgorithm,AlgorithmDesignTechniques, Performance Analysis of Algorithms, Complexity of various code structures, Order of Growth, Asymptotic Notations.
Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order,DerivationofIndexFormulae for 1-D,2-D Array, Application ofarrays,SparseMatricesand their representations.
Linked lists: Array Implementation and Pointer Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, PolynomialRepresentationandAdditionSubtraction& MultiplicationsofSingle variable.</td><td>10</td></tr>
<tr><td>2</td><td colspan="2">Stacks:AbstractDataType,PrimitiveStackoperations:Push&Pop, ArrayandLinkedImplementationofStackinC,Applicationofstack: Prefix and Postfix Expressions, Evaluation of postfix expression, Iteration and Recursion- Principles of recursion, Tail recursion, Removal of recursion Problem solving using iteration and recursion with examples such as</td><td>10</td></tr>
</table>

	binary search, Fibonacci numbers, and Hanoi towers. **Queues:**OperationsonQueue:Create,Add,Delete,FullandEmpty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue. **Searching:** Concept of Searching, Sequential search, Index SequentialSearch,BinarySearch.ConceptofHashing&Collision resolution Techniques used in Hashing.	
3	**Sorting:** Insertion Sort, Selection Sort, Bubble Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time: Counting Sort and Bucket Sort. **Graphs:** Terminology used with Graph, Data Structure for Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. **Graph Traversal:** Depth First Search and Breadth First Search,Connected Component.	**10**
4	**Trees:** Basic terminology used with Tree, Binary Trees, Binary Tree **Representation:** Array Representation and Pointer (Linked List) Representation, Binary Search Tree, Complete Binary Tree, A Extended Binary Trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search Tree. Threaded Binary trees, Huffman coding using Binary Tree, AVL Tree and B Tree.	**10**
5	Divide and Conquer with Examples Such as Merge Sort, Quick Sort, Matrix Multiplication: Strassen's Algorithm Dynamic Programming: Dijikstra Algorithm, Bellman Ford Algorithm, All- pair Shortest Path: Warshal Algorithm, Longest Common Sub-sequence Greedy Programming: Prims and Kruskal algorithm.	**10**

Course Outcomes

CO1: Explaintheconceptof datastructure,abstractdatatypes, algorithms, analysis of algorithms and basic data organization schemes suchas arrays and linked lists.

CO2: Describetheapplicationsofstacksandqueuesandimplement various operations on them using arrays and linked lists.

CO3: Describe the propertiesof graphsandtreesandimplement various operations such as searching and traversal on them.

CO4: Compareincremental and divide-and-conquer approaches of designing algorithms for problems such as sorting and searching.

CO5: ApplyandanalyzevariousdesignapproachessuchasDivide-and- Conquer, greedy and dynamic for problem solving.

Text books:

1. CormenT.H.,LeisersonC.E.,RivestR.L.andSteinC.,"IntroductiontoAlgorithms",PHI.
2. HorowitzE.,SahniS.andRajasekharanS.,"FundamentalsofComputerAlgorithms",UniversitiesPress.
3. DaveP.H.andDaveH.B,"DesignandAnalysisofAlgorithms",PearsonEducation.
4. LipschutsS.,"TheoryandProblemsofDataStructuresSchaum'sSeries)",TataMcGraw-Hill.
5. GoyalK.K.,SharmaS.andGuptaA.,"DataStructuresandAnalysisofAlgorithms",HPHamilton.

Reference books:

1- SamantaD.,"ClassicDataStructures",PrenticeHallIndia.
2- GoodrichM.T.andTomassiaR.,"AlgorithmDesign:Foundations,AnalysisandInternetExamples"
3- SridharS.,"DesignandAnalysisofAlgorithms",OxfordUniv.Press.
4- AhoA.,UllmanJ.andHopcroftJ.,"DesignandAnalysisofalgorithms",PearsonEducation.
5- NeapolitanR.andK.Naimipour,"FoundationsofAlgorithms",JonesandBartlettStudent

Assessment method: (Continuous Internal Assessment = 50th%, Final Examination = 50th%)

School Name- School of Computer Applications			
Program- MCA			**Semester-1st**
Course Name- Advance Data Structures and Algorithms Lab			
A.Y 2025-26	**Course Code- M020224152**	**Batch- 2023-27**	**CIE Marks- 60 (MM)**
Total Teaching Hours	**Total Credits- 2**		**ESE Marks- 40 (MM)**
Type of Course- Theory			**Total Marks 100 (MM)**

Course Objectives/Course Description

1. Develop programming skills to implement basic searching and sorting algorithms in C/C++.
2. Apply array-based techniques for performing operations such as addition, multiplication, and transposition of 2D arrays.
3. Implement stack and queue operations using both arrays and linked lists.
4. Solve graph-related problems such as minimum spanning tree using appropriate algorithms and data structures.

	Topics	**No. of Teaching hours/ (Lecture)**
	PrograminC or C++forfollowing: 1. Toimplementadditionandmultiplicationoftwo2Darrays. 2. Totransposea 2D array. 3. Toimplementstackusingarray. 4. Toimplementqueueusingarray. 5. Toimplementcircularqueueusingarray. 6. Toimplementstackusinglinkedlist. 7. Toimplementqueueusinglinkedlist. 8. ToimplementBFSusinglinkedlist. 9. ToimplementDFSusinglinked list. 10. ToimplementLinearSearch. 11. ToimplementBinarySearch. 12. ToimplementBubble Sorting. 13. ToimplementSelectionSorting. 14. ToimplementInsertion Sorting. 15. ToimplementMergeSorting. 16. ToimplementHeapSorting. 17. ToimplementMatrixMultiplicationbyStrassen'salgorithm. FindMinimumSpanningTreeusingKruskal'sAlgorithm.	**10**

Course Outcomes

CO1: Writeandexecuteprogramstoimplementvarioussearchingand sorting algorithms.
CO2 Writeandexecuteprogramstoimplementvariousoperationson two-dimensional arrays.
CO3: ImplementvariousoperationsofStacksandQueuesusingbotharrays and linked lists data structures.
CO4: Implementgraphalgorithmtosolvetheproblemofminimumspanning tree.

Assessment method: (Continuous Internal Assessment = 60th%, Final Examination = 40th%)

School Name- School of Computer Applications			
Program-MCA			**Semester-Ist**
Course Name-Advance Operating System			
A.Y 2025-26	**Course Code-M020224103**	**Batch-2023-25**	**CIE Marks- 50 (MM)**
Total Teaching Hours 50	**Total Credits-04**		**ESE Marks- 50 (MM)**
Type of Course- Theory			**Total Marks- 100 (MM)**

Course Objectives/Course Description

•Understandadvancedconceptsofdistributedsystemsandarchitectures.

•Analyzeandapplyprocesssynchronizationanddeadlockhandlingtechniques.•Exploredistributedfilesystemsand memorymanagementindistributedenvironments.•Learnfaulttolerance,security,andrecoverymechanismsindistributedsystems.•Gainexpertiseincloudcomputing,virtualization,andthedesignofscalablesystems.

UNITS	Topics	No. of Teaching hours/ (Lecture)
1	Distributed Systems and Architectures Overview of Distributed Systems: Definition, goals, and typesofdistributedsystems.Architectures:Client-server,peer-to-peer,andmulti-tierarchitecture. Communication: Remote Procedure Calls (RPC), Remote Method Invocation (RMI), message- oriented communication. Synchronization in Distributed Systems. Clock synchronization.Case Studies: Distributed file systems, examples from Hadoop or Google File System.	**10**
2	ProcessSynchronizationandDeadlocksAdvancedProcessSynchronization: Mutualexclusion, synchronization primitives (semaphores, monitors), and classical synchronization problems. Deadlock Detection, Avoidance, and Prevention: Concepts of deadlock, conditions, deadlock handlingtechniques.ResourceAllocationGraphs:Detectionalgorithmsa ndBanker'salgorithm. Real-TimeOperatingSystems (RTOS): Schedulingalgorithms, real-timetask synchronization.	**10**
3	DistributedFileSystemsandMemoryManagementDistributedFileSystems(DFS):Designand implementation, consistency and replication, fault tolerance, NFS and HDFS. Memory Management in Distributed Systems. Advanced Paging Techniques: Demand paging, pre-paging, page replacement policies. Memory allocation techniques:-Worst Fit, Best Fit, First Fit. VirtualMemorySystems:Segmentationandpaging,translationlookasidebuff er(TLB).	**10**

4	Fault Tolerance, Security, and Recovery in Distributed Systems Fault Tolerance Mechanisms: Redundancy, replication, failover techniques. Consensus Algorithms: Paxos, Raft. Security in Distributed Systems: Authentication, encryption, secure communication protocols, Kerberos. RecoveryandCheck pointing:Failurerecoverymechanisms,check pointingalgorithms,rollback recovery.	**10**
5	Cloud and Virtualization Cloud Computing Concepts: Virtualization, cloud architecture, cloud services (IaaS, PaaS, SaaS).Virtual Machines and Hypervisors: Types of virtualization (full, para, hardware-assisted), hypervisor types (Type-1, Type-2).Containerization: Docker, Kubernetes, micro services architecture. Resource Management in Cloud Systems: Scheduling, load balancing, dynamic resource allocation. Case Studies: VMware, Amazon Web Services (AWS), Google Cloud Platform (GCP).	**10**

Course Outcomes

CO1Understand the architecture and synchronization mechanisms of distributed systems.
CO2 Analyzeandhandledeadlockconditionsinadvancedoperatingsystems.
CO3 Managememoryandfilesystemsinadistributedenvironment.
CO4 Implementfaulttolerance,security,andrecoveryprotocolsinoperatingsystems.
CO5 Explorecloudcomputingtechnologiesandvirtualizationtechniques.

Textbook:**1.**DistributedSystems:PrinciplesandParadigmsbyAndrewS.Tanenbaumand Maarten VanSteen 2.Modern Operating Systems byAndrewS. Tanenbaum 3.Distributed Operating **Systems** by MukeshSinghal and Niranjan G. Shivaratri.

ReferenceBook:Real-TimeSystems:DesignPrinciplesforDistributedEmbedded Applications by Hermann Kopetz.

Assessment method: (Continuous Internal Assessment = 50%, Final Examination = 50%)

School Name- School of Computer Applications			
Program-MCA			**Semester-Ist**
Course Name-Advance Operating System Lab			
A.Y 2025-26	**Course Code-M020224153**	**Batch-2023-25**	**CIE Marks- 60 (MM)**
Total Teaching Hours 15	**Total Credits- 01**		**ESE Marks- 40 (MM)**
Type of Course- Practical			**Total Marks- 100 (MM)**

Course Objectives/Course Description

- Understand how process management efficiently managing processes while ensuring system stability, resource utilization, and responsiveness.
- Manage and optimize the execution of processes (or jobs) in a system in such a way that system resources (like CPU, memory, etc.) are utilized efficiently.
- Understand how a computer system allocates memory to processes in an efficient and effective manner.• **Implement deadlock detection and avoidance mechanisms**to ensure a computer system can either**prevent deadlocks from occurring**or**detect and recover from deadlocks**when they do occur**.**
- ImplementanRPCmechanismwhereaclientcancallfunctionsonaserverlocatedonadifferent machine.

	Topics	**No. of Teaching hours/ (Lecture)**
	List of Experiments (Indicative & not limited to) 1. Manage and monitor process using `fork()`, `exec()`, and `wait()`. 2. Implement the job scheduling algorithm First Come First Served. 3. Implement the job scheduling algorithm Round Robin. 4. Implement the job scheduling algorithm Shortest Job First. 5. Implement the job scheduling algorithm Priority Scheduling. 6. Implement the memory allocation technique First Fit. 7. Implement the memory allocation technique Best Fit. 8. Implement the memory allocation technique Worst Fit. 9. Simulate deadlock detection and avoidance: a) Banker's algorithm. b) Resource allocation Graph (RAG) 10. Write a program to simulate producer-consumer problem using semaphore.	

Course Outcomes

CO1Understandthe process management.

CO2 Implement job scheduling algorithms. **CO3 Learnthe memory allocation techniques.** **CO4 Gain the knowledge of deadlock detection and avoidance mechanism.** **CO5 Understandtheworkingof producer-consumer problem using semaphore.**

Assessment method: (Continuous Internal Assessment = 60%, Final Examination = 40%)

,