

Osmania University

Faculty of Informatics

Master of Computer Applications (MCA)

CBCS (Choice-Based Credit System) Syllabi

Semester I and II

With effect from the academic year 2026-2027

**Under R26 Academic Regulations
(Applicable to Students Admitted
from the Academic Year 2026–2027)**



**Faculty of Informatics
Osmania University
2026-2027**

Contents

MCA Program Outcomes	3
Sustainable Development Goals(SDG)- Courses Mapping Matrix.....	5
Semester I.....	5
Semester II	6
General Course Structure.....	7
SCHEME OF INSTRUCTION	1
Semester – I	1
Semester – II	1
Detailed Syllabus for Semester II	23
Model Question Paper Format for CIE.....	49
Model Question Paper Format for SEE	50


Dean,
Faculty of Informatics
Osmania University

MCA Program Outcomes

PO1	Computational Knowledge: Apply knowledge of Computing fundamentals, Computing specialization, Mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and Conceptualization of computing models from defined problems and requirements.
PO2	Problem analysis: Identify, formulate, research literature, and solve complex Computing problems reaching substantiated conclusions using fundamental principles of Mathematics, Computing sciences, and relevant domain disciplines.
PO3	Design /Development of Solutions: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex Computing problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO5	Modern Tool Usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
PO6	Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.
PO7	Life-long Learning: Recognize the need, and have the ability, to engage in independent learning for continual development as a Computing professional.
PO8	Project management and finance: Demonstrate knowledge and understanding of computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO9	Communication Efficacy: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
PO10	Societal and Environmental Concern: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the

P.V. Sudhakar

CBOS & Dean | Faculty of Informatics

3
Dean,
Faculty of Informatics
Osmania University

	consequential responsibilities relevant to professional computing practice.
PO11	Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.
PO12	Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.


*Dean,
Faculty of Informatics
Osmania University*

Semester II

Course	SDG 4	SDG 8	SDG 9	SDG 11	SDG 12	SDG 13	Remarks
Applied Mathematics	✓	✓	✓				Statistical analysis and decision making
Database Systems	✓	✓	✓	✓			Smart databases and digital governance
Design and Analysis of Algorithms	✓	✓	✓				Efficient algorithmic solutions
Machine Learning	✓	✓	✓	✓	✓	✓	AI for sustainability and intelligent systems
Data Engineering with Python	✓	✓	✓	✓	✓		Data-driven decision support systems
Organization Behaviour	✓	✓			✓		Ethical leadership and sustainable organizations
Database Systems Lab	✓	✓	✓	✓			Practical database application development
Design and Analysis of Algorithms Lab	✓	✓	✓				Algorithm implementation and optimization
Machine Learning Lab	✓	✓	✓	✓	✓	✓	Practical AI for sustainable applications
Summer Internship / Mini Project	✓	✓	✓	✓	✓	✓	Industry collaboration and real-world innovation

- SDG 4 (Quality Education) is mapped to every course because all courses contribute directly to knowledge, skills, and lifelong learning.
- SDG 8 (Decent Work and Economic Growth) is mapped to almost all MCA courses due to their focus on employability and professional competencies.
- SDG 9 (Industry, Innovation and Infrastructure) is central to the MCA curriculum and is therefore associated with nearly all technical courses.
- SDG 11 (Sustainable Cities and Communities) is mapped where technologies such as AI, databases, machine learning, and data engineering contribute to smart city and digital governance applications.
- SDG 12 (Responsible Consumption and Production) is included for courses involving efficient data management, sustainable business practices, and responsible resource utilization.
- SDG 13 (Climate Action) is mapped to AI, Machine Learning, and Internship/Projects where solutions can address environmental sustainability and climate-related challenges

P. V. Gollu

Dean,
Faculty of Informatics
Osmania University

General Course Structure

Definition of Credit

Type	Credit
1 Hr. Lecture (L) per week	1 Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. Practical (P) per week	0.5 Credit
2 Hours Practical(P) per week	1 Credit

Abbreviation and Definition

Abbreviations	Definitions
PCC	Professional Core Course
PEC	Professional Elective Course
MGC	Management Course
HSC	Humanities Course
LCC	Laboratory Core Course
CIE	Continuous Internal Evaluation
SEE	Semester End Evaluation
HPW	Hours Per Week
L	Lecture Hours/Week
T	Tutorial Hours/Week
P	Practical Hours/Week
Cr	Credits

Scheme of Examination

	CIE-Marks	SEE-Marks	CIE-Duration	SEE-Duration
Theory	30	70	1 hr	3hrs
Practical	25	50	2 hr	3hrs
Internship	50	-	3hrs	-
Project	50	100	3hrs	3hrs

Credit Table R26(2026-2028)

Semester	Credits
I	24
II	24
III	28
IV	22
Total	98

P.V. Sridhar

Dean,
Faculty of Informatics
Osmania University

SCHEME OF INSTRUCTION

Master of Computer Applications

Semester – I

Course Code	Course Title	Credits	HPW (L/P)	Max Marks (CIE + SEE)
Theory				
PCC101	Mathematical Foundations of Computer Science	3	3L	30 + 70
PCC102	Data Structures using C	3	3L	30 + 70
PCC103	Object Oriented Programming Using Python	3	3L	30 + 70
PCC104	Operating Systems	3	3L	30 + 70
PCC105	Artificial Intelligence	3	3L	30 + 70
MGC106	Managerial Economics and Accountancy	3	3L	30 + 70
Practical				
LCC102	Data Structures using C Lab	2	4P	25 + 50
LCC103	Python Programming Lab	2	4P	25 + 50
HSC110	Soft Skills Lab	2	4P	25 + 50
Total		24	30	255+570

Semester – II

Course Code	Course Title	Credits	HPW (L/P)	Max Marks (CIE + SEE)
Theory				
PCC201	Applied Mathematics	3	3L	30 + 70
PCC202	Database Systems	3	3L	30 + 70
PCC203	Design and Analysis of Algorithms	3	3L	30 + 70
PCC204	Machine Learning	3	3L	30 + 70
PCC205	Fundamentals of Data Engineering	3	3L	30 + 70
PCC206	Computer Networks	3	3L	30 + 70
Practical				
LCC202	Database Systems Lab	2	4P	25 + 50
LCC203	Design and Analysis of Algorithms Lab	2	4P	25 + 50
LCC204	Machine Learning Lab	2	4P	25 + 50
SIP310	Summer Internship/ Mini Project *			
Total		24	30	255+570

*To be completed after the second semester, with credits to be awarded in the third semester.

P. V. Sudhakar

CBoS & Dean | Faculty of Informatics

Dean,
Faculty of Informatics
Osmania University

Detailed Syllabus for Semester II**PCC201****Applied Mathematics****Credits: 3****Course Objectives:** After completing this course, students will be able to:

1. Understand the concepts of partial differentiation and functions of several variables.
2. Apply total differentiation and optimization techniques for solving problems involving maxima and minima.
3. Analyze systems of linear equations and determine linear dependence and independence of vectors and matrices.
4. Understand vector spaces, subspaces, bases, eigen values, and eigenvectors in linear algebra.
5. Apply diagonalization techniques and use eigen value methods in solving differential equations and related applications.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Compute partial derivatives and analyze continuity and limits of functions of two variables.
2. Solve problems involving total differentials, implicit functions, and constrained optimization using Lagrange multipliers.
3. Solve systems of linear equations and determine the linear independence or dependence of vectors and matrices.
4. Demonstrate knowledge of vector spaces, bases, eigen values, eigenvectors, and characteristic equations.
5. Diagonalize matrices and apply eigen value techniques to solve differential equations and engineering applications.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	–	1	–	1	–	–	–	–	–
CO2	3	3	1	1	1	–	1	–	–	–	–	–
CO3	3	3	2	1	2	–	1	–	–	–	–	–
CO4	3	2	2	2	2	–	1	–	–	–	–	–
CO5	3	3	3	2	2	–	2	1	1	–	–	1

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

Course Contents

UNIT – I: Partial Differentiation

Introduction, Functions of Two Variables, Neighborhood of a Point $((a, b))$, Continuity of a Function of Two Variables, Continuity at a Point, Limit of a Function of Two Variables, Partial Derivatives, Homogeneous Functions

UNIT – II: Total Differentiation and Optimization

Theorem on Total Differentials, Composite Functions, Differentiation of Composite Functions, Implicit Functions, Maxima and Minima of Functions of Two Variables, Lagrange's Method of Undetermined Multipliers.

UNIT – III: Linear Equations in Linear Algebra

Systems of Linear Equations, Consistent and Inconsistent Systems, Solution Sets of Linear Systems, Trivial and Non-Trivial Solutions, Linear Independence, Linear Independence of Matrix Columns, Characterization of Linearly Dependent Sets

UNIT – IV: Vector Spaces and Eigen values

Vector Spaces and Subspaces, Linearly Independent Sets, Bases, Eigenvalues and Eigenvectors, The Characteristic Equation, Diagonalizing Matrices with Non-Distinct Eigenvalues, Applications to Differential Equations

UNIT- V: Probability & Sampling

Basic terminology, three types of probability, probability rules, statistical independence, statistical dependency, Bayes' theorem. Probability Distributions - Random variables, expected values, binomial distribution, Poisson distribution, normal distribution, choosing the correct distribution. Sampling and Sampling Distributions - Random sampling, Non-Random Sampling distributions, operational considerations in sampling. Estimation - Point estimates, interval estimates, confidence intervals, calculating interval estimates of the mean and proportion, t-distribution, determination of sample size in estimation.

Suggested Readings

1. Lay, David C. Linear Algebra and Its Applications. 6th ed., Pearson Education Limited, 2022.
2. Lang, Serge. Introduction to Linear Algebra. 3rd ed., Springer, 2000.
3. Strang, Gilbert. Linear Algebra and Its Applications. 4th ed., Cengage Learning, 2005.
4. T. Veerarajan, Probability, Statistics and Random Processes, 3rd Edition, Tata McGraw-Hill, 2008

P. V. Sridhar

CBoS & Dean | Faculty of Informatics

PCC202**Database Systems****Credits: 3****Course Objectives**

1. Introduce database concepts along with ER modeling.
2. Learn about relational databases and SQL query language.
3. Define advanced SQL.
4. Study database transactions and explore concurrency concepts.
5. Introduce NoSQL.

Course Outcomes

1. Explain database concepts and model requirements using the ER model.
2. Formulate relational algebra queries from textual specifications.
3. Write SQL queries for given problems.
4. Elaborate on indexing and hashing, and describe concurrency control concepts.
5. Understand NoSQL technology.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	1	2	–	2	–	1	–	–	–
CO2	3	3	3	2	2	–	2	–	1	–	–	–
CO3	3	3	3	2	3	–	2	–	1	–	–	–
CO4	3	3	2	3	3	1	2	–	1	–	–	–
CO5	3	2	2	2	3	–	2	–	1	–	–	1

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents****Unit-I**

Introduction: Database system applications, purpose of database systems, views of values, nested sub-queries, complex queries, views, modification of the database, joined relations data, database languages, relational databases, database design, object-based and semi-structured databases, data storage and querying, transaction management, data mining and analysis, database architecture, database users and administrators. **Database Design and the E-R Model:** Overview of the design process, the entity-relationship model, constraints, entity-relationship diagrams, entity-relationship design issues, weak entity sets, extended E-R features, database design for a banking enterprise, reduction to relational schemas, and other aspects of database design.



CBoS & Dean | Faculty of Informatics

Unit-II

Relational Model: Structure of relational databases, fundamental relational algebra operations, additional relational algebra operations, extended relational algebra operations, null values, modification of databases. **Structured Query Language (SQL):** Data definition, basic structure of SQL queries, set operations, aggregate functions, null.

Unit-III

Advanced SQL: SQL data types and schemas, integrity constraints, authorization, embedded SQL, dynamic SQL, functions and procedural constructs, recursive queries, and advanced SQL features.

Relational Database Design: Features of good relational design, atomic domains and first normal form, functional dependency theory, decomposition using functional dependencies.

Unit-IV

Indexing and Hashing: Basic concepts, ordered indices, B+tree index files, Btree index files, multiple-key access, static hashing, dynamic hashing, comparison of ordered indexing and hashing, bitmap indices. **Index Definition in SQL Transactions:** transaction concepts, transaction states, implementation of atomicity and durability, concurrent executions, serializability, recoverability, implementation of isolation, testing for serializability.

Unit-V

Concurrency Control: Lock-based protocols, timestamp-based protocols, validation-based protocols, multiple granularity, multi-version schemes, deadlock handling, insert and delete operations, weak levels of consistency, concurrency of index structures. **Recovery System:** Failure classification, storage structure, recovery and atomicity, log-based recovery, recovery with concurrent transactions, buffer management, failure with loss of nonvolatile storage, advanced recovery techniques, remote backup systems. **NoSQL:** Need for NoSQL, aggregate data models, more details on data models, distribution models, consistency, version stamps, map-reduce, key-value databases, document databases, column-family stores, graph databases, schema migrations.



CBOS & Dean | Faculty of Informatics

Suggested readings

1. Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. Database System Concepts. 7th ed., McGraw Hill, 2020.
2. Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems. 3rd ed., McGraw-Hill Education, 2003.
3. Elmasri, Ramez, Shamkant B. Navathe, and Shyam K. Gupta. Fundamentals of Database Systems. 7th ed., Pearson, 2017.
4. Tiwari, Shashank. Professional NoSQL. 1st ed., Wiley, 2011.

P.V. Subha

CBoS & Dean | Faculty of Informatics

PCC203**Design and Analysis of Algorithms****Credits: 3****Course Objectives**

1. Learn algorithms time complexity.
2. Learn divide and conquer approach.
3. Learn greedy method.
4. Learn dynamic programming.
5. Learn backtracking.

Course Outcomes

1. Carry out algorithms time complexity.
2. Explain divide and conquer approach.
3. Illustrate greedy method.
4. Elaborate dynamic programming.
5. Explore backtracking.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	3	2	–	2	–	1	–	–	–
CO2	3	3	3	2	2	–	2	–	1	–	–	–
CO3	3	3	3	2	2	–	2	–	1	–	–	1
CO4	3	3	3	3	2	–	2	–	1	–	–	1
CO5	3	3	3	3	2	–	2	–	1	–	–	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents****Unit- I**

Introduction to Algorithms: Algorithm Specifications, Performance Analysis, Randomized Algorithms. **Elementary Data Structures:** Stacks and Queues, Trees, Dictionaries, Priority Queues, Sets and Disjoint Set Union, Graphs.

Unit-II

Divide and Conquer: Binary Search, Finding the Maximum and Minimum, Merge Sort, Quick Sort, Selection sort, Strassen's Matrix Multiplication, Convex Hull. **The Greedy Method:** Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees, Single Source Shortest Paths.

P. V. Sridhar
CBoS & Dean | Faculty of Informatics

Unit-III

Dynamic Programming: General Method, Multistage Graphs, All-Pairs Shortest Paths, Single-Source Shortest Paths, Optimal Binary Search Trees, 0/1 Knapsack, The Traveling Salesperson Problem. **Basic Traversal and Search Techniques:** Techniques for Binary Trees, Techniques for Graphs, Connected Components and Spanning Trees, Biconnected Components and DFS.

Unit-IV

Back Tracking: General Method, 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack Problem. **Branch-Bound:** The Method, 0/1 Knapsack Problem, Traveling Sales Person.

Unit-V

NP-Hard and NP-Completed Problems: Basic Concepts, Cook's Theorem, NP-Hard Graph Problems, NP-Hard Scheduling Problems, NP-Hard Code Generation, Some Simplified NP-Hard Problems.

Suggested Readings

1. Horowitz, Ellis, SartajSahni, and SanguthevarRajasekaran. Fundamentals of Computer Algorithms. 2nd ed., Universities Press, 2008.
2. Pannerselvam, R. Design and Analysis of Algorithms. 3rd ed., PHI Learning Pvt. Ltd., 2019.
3. Pandey, Hari Mohan. Design Analysis and Algorithms. 1st ed., University Science Press, 2009.
4. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms. 4th ed., MIT Press, 2022.



PCC204**Machine Learning****Credits: 3****Course Objectives:** After completing this course, students will be able to:

1. Understand the machine learning lifecycle, problem formulation, data preparation, model development, and evaluation processes.
2. Apply supervised learning techniques for regression and classification tasks using appropriate performance measures.
3. Analyze decision tree and ensemble learning methods for building robust predictive models.
4. Explore dimensionality reduction and unsupervised learning techniques for data analysis and pattern discovery.
5. Develop and evaluate artificial neural network models using Scikit-learn and PyTorch for real-world machine learning applications.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Explain machine learning concepts, types of learning systems, challenges in machine learning, and the end-to-end machine learning workflow.
2. Implement and evaluate classification and regression models including Linear Regression, Logistic Regression, and multiclass classification techniques using appropriate performance metrics.
3. Develop predictive models using Decision Trees, Random Forests, Boosting, Bagging, Voting Classifiers, and other ensemble learning methods.
4. Apply dimensionality reduction techniques such as PCA and implement clustering and Gaussian Mixture Models for unsupervised learning tasks.
5. Design, train, and evaluate Artificial Neural Networks and Multi-Layer Perceptrons using Scikit-learn and PyTorch, including image classification and deep learning applications.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	3	—	2	—	1	—	—	—
CO2	3	3	3	3	3	—	2	—	1	—	—	1
CO3	3	3	3	3	3	—	2	—	1	—	—	2
CO4	3	3	3	3	3	—	2	—	1	—	—	2
CO5	3	3	3	3	3	—	2	—	1	1	1	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Course Contents**


CBoS & Dean | Faculty of Informatics

Unit-I

Machine Learning Landscape: machine learning introduction, use of machine learning, types of machine learning systems, main challenges of machine learning, testing and validating, **End-to End Machine Learning:** Working with real data, frame the problem, performance measure, get the data, explore and visualize the data to gain insights, prepare the data for machine learning algorithms, select and train a model, fine-tune your model.

Unit-II

Training a binary classifier: performance measures, multiclass classification, error analysis, multi-label classification, multi-output classification, MNIST, **Training Models:** Linear regression, Gradient descent, regularized linear models, logistic regression.

Unit-III

Decision Trees: Training and visualizing a decision tree, making predictions, estimating class probabilities, the CART training algorithm, Gini impurity, Entropy, regularization hyperparameters, **Ensemble Learning and Random Forest:** voting classifiers, bagging and pasting, random forests, boosting, stacking.

Unit-IV

Dimensionality Reduction: the curse of dimensionality, main approaches for dimensionality reduction, PCA, **Unsupervised Learning Techniques:** clustering algorithms, Gaussian mixtures-using gaussian mixtures for anomaly detection, selecting the number of clustering, Bayesian gaussian mixture models.

Unit-V

Introduction to Artificial Neural Networks: from biological to artificial neurons, building and training MLPs with scikit-learn, hyperparameter tuning guidelines, **Building neural networks with PyTorch:** PyTorch fundamentals, implementing linear regression, implementing a regression MLP, implementing mini-batch gradient descent using dataloaders, model evaluation, building an image classifier with PyTorch.

P.V. Sridhar

Suggested Readings

1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition, O'reilly, 2025.
2. Stephen Marsland, "Machine Learning: An Algorithmic Perspective", Chapman & Hall/CRC, 2nd Edition, 2014.
3. Kevin Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012
4. EthemAlpaydin, "Introduction to Machine Learning", Third Edition, Adaptive Computation and Machine Learning Series, MIT Press, 2014
5. Tom M Mitchell, "Machine Learning", McGraw Hill Education, 2013.

P.V. Subbarao

PCC205**Fundamentals of Data Engineering****Credits: 3****Course Objectives:** After completing this course, students will be able to:

1. Understand the Role of Data Engineering in Modern Organizations
2. Design Scalable and Reliable Data Architectures.
3. Select and Implement Appropriate Technologies Across the Data Engineering Lifecycle
4. Optimize Data Storage and Ingestion Techniques.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Demonstrate a comprehensive understanding of the data engineering lifecycle, encompassing essential skills, activities, and the roles of a data engineer in modern business settings.
2. Design and assess scalable and reliable data architectures that integrate modern data platforms like data lakes and warehouses.
3. Utilize decision-making skills to select optimal technologies and frameworks based on team capabilities, performance, cost, and interoperability across the data engineering lifecycle.
4. Implement effective data storage solutions by applying knowledge of distributed storage, partitioning, data retention, and modern storage architectures such as data lakehouses and stream-to-batch systems.
5. Build and manage data ingestion pipelines that handle both bounded and unbounded data while ensuring scalability, data integrity, and reliability through techniques like ETL, change data capture, and message queues.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	2	—	2	—	1	—	—	—
CO2	3	3	2	3	3	—	2	—	1	—	—	—
CO3	3	3	3	2	3	—	2	—	1	—	—	1
CO4	3	3	3	3	3	—	2	—	2	—	—	1
CO5	3	3	3	2	3	1	2	—	1	—	—	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

CBoS & Dean | Faculty of Informatics

Course Contents

Unit-I

Data Engineering Described : Definition, Lifecycle, Evolution of the Data Engineer, Data Engineering and Data Science , Data Engineering Skills and Activities, Data Maturity and the Data Engineer, The Background and Skills of a Data Engineer, Business Responsibilities , Technical Responsibilities, The Continuum of Data Engineering Roles, from A to B , Data Engineers Inside an Organization, Internal-Facing Versus External-Facing Data Engineers, Data Engineers and Other Technical Roles, Data Engineers and Business Leadership.

The Data Engineering Lifecycle: The Data Lifecycle Versus the Data Engineering Lifecycle, Generation: Source Systems, Storage, Ingestion, Transformation, Serving Data, Major Undercurrents Across the Data Engineering Lifecycle, Security, Data Management, DataOps, Data Architecture, Orchestration, Software Engineering.

Unit-II

Designing Good Data Architecture: What Is Data Architecture, Enterprise Architecture Defined, Data Architecture Defined, “Good” Data Architecture, Principles of Good Data Architecture, Major Architecture Concepts.

Domains and Services, Distributed Systems, Scalability, and Designing for Failure, Tight Versus Loose Coupling: Tiers, Monoliths, and Microservices, **User Access:** Single Versus Multitenant, Event-Driven Architecture,

Brownfield Versus Greenfield Projects, Examples and Types of Data Architecture, Data Warehouse, Data Lake, Convergence, Next-Generation Data Lakes, and the Data Platform; Modern Data Stack, Lambda Architecture, Kappa Architecture, The Dataflow Model and Unified Batch and Streaming, Architecture for IoT, Data Mesh, Other Data Architecture Examples, Who’s Involved with Designing a Data Architecture.

Unit-III

Choosing Technologies Across the Data Engineering Lifecycle: Team Size and Capabilities, Speed to Market, Interoperability, Cost Optimization and Business Value, Total Cost of Ownership, Total Opportunity Cost of Ownership, FinOps, Optimization, Performance, and the Benchmark Wars.

Data Generation in Source Systems: Sources of Data: How Is Data Created? Source Systems: Main Ideas, Files and Unstructured Data, APIs, Application Databases (OLTP Systems), Online Analytical Processing System, Change Data Capture, Logs, Database Logs, CRUD, Insert-Only.

Messages and Streams, Types of Time, Source System Practical Details, Databases, APIs, Data Sharing, Third-Party Data Sources, Message Queues and Event-Streaming Platforms, Undercurrents and Their Impact on Source Systems, Security, Data Management, DataOps, Data Architecture, Orchestration, Software Engineering.

P. V. Sudha

CBOS & Dean | Faculty of Informatics

Unit-IV

Storage: Data Storage Systems, Single Machine Versus Distributed Storage, Eventual Versus Strong Consistency, Cache and Memory-Based Storage Systems, The Hadoop Distributed File System, Streaming Storage, Indexes, Partitioning, and Clustering, Data Engineering Storage Abstractions, The Data Warehouse, The Data Lake, The Data Lakehouse, Data Platforms.

Stream-to-Batch Storage Architecture, Big Ideas and Trends in Storage, Data Catalog, Data Sharing, Schema, Separation of Compute from Storage, Data Storage Lifecycle and Data Retention, Single-Tenant Versus Multitenant Storage

Unit-V

Ingestion: What Is Data Ingestion, Key Engineering Considerations for the Ingestion Phase, Bounded Versus Unbounded Data, Frequency, Synchronous Versus Asynchronous Ingestion, Serialization and Deserialization, Throughput and Scalability, Reliability and Durability, Payload, Push Versus Pull Versus Poll Patterns, Batch Ingestion Considerations, Snapshot or Differential Extraction, File-Based Export and Ingestion, ETL Versus ELT

Inserts, Updates, and Batch Size, Data Migration, Message and Stream Ingestion Considerations, Schema Evolution, Late-Arriving Data, Ordering and Multiple Delivery, Replay, Time to Live, Message Size, Error Handling and Dead-Letter Queues, Consumer Pull and Push, Location, Ways to Ingest Data, Direct Database Connection, Change Data Capture, APIs, Message Queues and Event-Streaming Platforms, Managed Data Connectors, Moving Data with Object Storage, EDI, Databases and File Export.

Suggested Readings

1. Joe Reis and Matt Housley, *Fundamentals of Data Engineering Plan and Build Robust Data Systems*, Published by O'Reilly Media, Inc., July 2022: First Edition.
2. Paul Crickard, *Data Engineering with Python: Work with massive datasets to design data models and automate data pipelines using Python*, Kindle Edition Packt Publishers.



PCC206**COMPUTER NETWORKS****Credits 03****Course Objectives:**

1. To study the design issues in network layer and various routing algorithms.
2. To introduce internet routing architecture and protocols.
3. To learn the flow control and congestion control algorithms in Transport Layer.
4. To introduce the TCP/IP suite of protocols and the networked applications supported by it.
5. To learn basic and advanced socket system calls.

Course Outcomes :

At the end of the course the student will be able to:

1. Apply the function of each layer of OSI and trace the flow of information from one node to another node in the network.
2. Understand the principles of IP addressing and internet routing.
3. Analyze the working of networked applications such as DNS, mail file transfer and www.
4. Implement client-server socket-based networked applications.
5. Implement the use of Computer Networks for Large scale Networked systems

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	3	—	2	—	1	—	—	—
CO2	3	3	3	2	3	—	2	—	1	—	—	—
CO3	3	3	3	2	3	—	2	—	1	—	1	1
CO4	3	3	3	3	3	1	2	—	1	—	—	—
CO5	3	3	3	2	3	—	2	—	1	—	—	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

UNIT – I

Data Communications: Components, analog and digital signals and Encoders, Modems, S232 Interfacing. Switching: Circuit Switching, Message Switching and Packet Switching. Topologies – Concept of layering.-Protocols and Standards – ISO / OSI model, TCP/IP.

UNIT – II

Data Link Layer: Error Control: Error detection and correction (CRC and Hamming code for single bit correction). Flow Control: stop and wait – - sliding window protocols-go Back-N ARQ – selective repeat ARQ. MAC LAYER: Ethernet IEEE 802.3LAN, Manchester encoding, Binary exponential algorithm, Efficiency calculation, ARP and RARP.

UNIT – III

Network Layer : Internetworks – virtual circuit and Datagram approach Routing – Distance Vector Routing ,Link State Routing , OSPF and BGP IPv4 , addressing, Subnetting, IPv6, CIDR, ICMP and IGMP protocols.


CBoS & Dean | Faculty of Informatics

UNIT – IV

Transport Layer: Services of transport layer, Multiplexing and crash recovery. Transmission Control Protocol (TCP) – TCP window management Congestion Control, timer management and User Datagram Protocol (UDP). BE: COMPUTER SCIENCE AND ENGINEERING With effect from Academic year 2024-25

UNIT – V

Socket Programming: Primitive and advanced system calls, client/server iterative and concurrent programs IO multiplexing, Asynchronous IO and select system call. Application Layer: Domain Name Space (DNS) – SMTP – FTP – HTTP.

References:

1. Computer Networks, 5th Edition, Andrew S. Tanenbaum , David J. Wetherall , Pearson Education, 2021.
2. Computer Networks: A Systems Approach, Larry Peterson and Bruce Davie, Elsevier , 5th Edition, 2021.
3. Computer Networking: A Top-Down Approach, 6th Edition, James F. Kurose , Keith W. Ross, Pearson , 2022.

P. V. Sridhar

CBoS & Dean | Faculty of Informatics

LCC202**Database Systems Lab****Credits: 2****Course Objectives:** After completing this laboratory course, students will be able to:

1. Understand database design principles and develop Entity-Relationship (ER) models and relational schemas for real-world applications.
2. Apply SQL commands and integrity constraints for database creation, manipulation, and querying.
3. Design and implement advanced database objects such as procedures, functions, triggers, and indexes.
4. Analyse normalization, transaction management, concurrency control, and recovery mechanisms in database systems.
5. Explore NoSQL databases and compare relational and non-relational database models for modern data management applications.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Design ER diagrams, relational schemas, and database structures by applying database modeling concepts and integrity constraints.
2. Create and manage databases using SQL DDL and DML commands, joins, set operations, aggregate functions, and nested queries.
3. Develop stored procedures, functions, triggers, and indexes to enhance database functionality and performance.
4. Apply normalization techniques, transaction control operations, concurrency control methods, and recovery procedures to ensure database consistency and reliability.
5. Implement NoSQL database operations using MongoDB or Cassandra and compare their features, query mechanisms, and applications with relational databases.

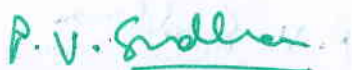
CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	3	—	2	—	1	—	—	—
CO2	3	3	3	2	3	—	2	—	1	—	—	—
CO3	3	3	3	2	3	—	2	—	1	—	1	1
CO4	3	3	3	3	3	1	2	—	1	—	—	—
CO5	3	3	3	2	3	—	2	—	1	—	—	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping*P. V. Sridhar***CBoS & Dean | Faculty of Informatics**

Lab Experiments

1. Design an Entity-Relationship (E-R) diagram for a Banking Management System and convert it into relational schemas.
2. Develop a database schema for a Library Management System by identifying entities, attributes, relationships, and constraints.
3. Create a database for a Student Information System and demonstrate the use of primary keys, foreign keys, and integrity constraints.
4. Create tables for a College Database using SQL Data Definition Language (DDL) commands and enforce key constraints.
5. Perform SQL operations using:
 - a. SELECT
 - b. INSERT
 - c. UPDATE
 - d. DELETE
 - e. WHERE
 - f. ORDER BY
 - g. GROUP BYon a sample Employee database.
6. Write SQL queries using joins, set operations (UNION, INTERSECT, EXCEPT), aggregate functions, and nested subqueries.
7. Implement integrity constraints (PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, NOT NULL) and test their enforcement.
8. Create SQL procedures, functions, and triggers for a Banking or Student Database application.
9. Perform normalization of a given relation up to 3NF/BCNF by identifying functional dependencies and decomposition steps.
10. Create and analyse different types of indexes (Single-column, Composite, and B+ Tree indexes) and compare query performance.
11. Implement transaction management using COMMIT, ROLLBACK, and SAVEPOINT commands on a Banking Database.



12. Simulate concurrent transactions and demonstrate serializability, recoverability, and isolation levels.
13. Implement lock-based concurrency control and demonstrate deadlock detection and resolution techniques.
14. Simulate log-based recovery techniques for transaction failures and analyse database recovery procedures.
15. Install and use a NoSQL database (MongoDB/Cassandra) to:
 - a) Create collections/tables
 - b) Insert and retrieve documents
 - c) Perform aggregation queries
 - d) Compare NoSQL operations with SQL databases.

Mini Project (Comprehensive DBMS Application)

Database-Driven Management System

Develop a complete application such as:

- Student Management System
- Library Management System
- Hospital Management System
- Banking Management System

The project should include:

- E-R Diagram Design
- Relational Schema Design
- SQL Queries and Views
- Stored Procedures/Triggers
- Transactions and Concurrency Handling
- Indexing for Performance Optimization
- NoSQL Integration (Optional)
- Front-end Interface using Python/Java/Web Technologies

Suggested Tools: MySQL, PostgreSQL, Oracle, MongoDB, Python (Tkinter/Streamlit), PHP, Java, or Web Technologies.

P. V. Sudha

LCC203**Design and Analysis of Algorithms Lab****Credits: 2****Course Objectives:** After completing this laboratory course, students will be able to:

1. Understand and implement fundamental data structures such as stacks, queues, trees, and graphs for efficient data organization and processing.
2. Apply divide and conquer techniques to design and analyse efficient sorting and searching algorithms.
3. Implement greedy and dynamic programming approaches to solve optimization problems.
4. Develop solutions for graph traversal and shortest path problems using appropriate algorithmic techniques.
5. Analyse and implement backtracking and branch-and-bound methods for solving complex combinatorial and optimization problems.

Course Outcomes: Upon successful completion of the course, students will be able to:

1. Implement fundamental data structures including stacks, queues, binary search trees, and graph representations using suitable programming techniques.
2. Develop and analyse divide and conquer algorithms such as Merge Sort, Quick Sort, and Maximum-Minimum search for performance evaluation.
3. Apply greedy and dynamic programming techniques to solve optimization problems including Fractional Knapsack, Job Sequencing, and 0/1 Knapsack.
4. Implement graph algorithms such as BFS, DFS, Floyd-Warshall, and connected component identification for graph-based problem solving.
5. Design and evaluate solutions using backtracking and branch-and-bound techniques for problems such as N-Queens, Graph Coloring, Hamiltonian Cycle, Knapsack, and Travelling Salesperson Problem, while analysing computational complexity and performance.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	3	—	2	—	1	—	—	—
CO2	3	3	3	3	3	—	2	—	1	—	—	1
CO3	3	3	3	3	3	—	2	—	1	—	—	2
CO4	3	3	3	3	3	—	2	—	1	—	1	1
CO5	3	3	3	3	3	—	2	—	1	—	1	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping**Lab Experiments**

1. Implement Stack and Queue data structures using arrays/linked lists and perform basic operations (push, pop, enqueue, dequeue).
2. Develop a program to implement Binary Search Tree (BST) operations such as insertion, deletion, and traversal.
3. Implement Graph representation using Adjacency Matrix and Adjacency List and perform basic graph operations.


CBoS & Dean | Faculty of Informatics

4. Implement Merge Sort and Quick Sort algorithms and compare their execution times for different input sizes.
5. Develop a program to find the Maximum and Minimum elements in an array using the Divide and Conquer approach.
6. Implement Greedy algorithms for: Fractional Knapsack Problem, Job Sequencing with Deadlines and compare their results.
7. Implement the 0/1 Knapsack Problem using Dynamic Programming and analyse its time complexity.
8. Develop a program to solve the All-Pairs Shortest Path problem using the Floyd-Warshall algorithm.
9. Implement Breadth First Search (BFS) and Depth First Search (DFS) traversals on a graph and identify connected components.
10. Implement the N-Queens (8-Queens) problem using the Backtracking technique.
11. Develop a program to solve the Graph Coloring problem using Backtracking.
12. Implement the Traveling Salesperson Problem (TSP) using the Branch and Bound technique.
13. Implement the 0/1 Knapsack Problem using the Branch and Bound approach and compare the results with Dynamic Programming.
14. Develop a program to solve the Hamiltonian Cycle problem using Backtracking and analyse the solution space.
15. Perform a comparative study of Polynomial-Time and NP-Hard problems by implementing algorithms for:
 - a) Minimum Spanning Tree (Polynomial-Time)
 - b) Traveling Salesperson Problem (NP-Hard)
 - c) Analyze execution time and computational complexity.

P.V. Gollu

CBoS & Dean | Faculty of Informatics

Mini Project

Algorithm Visualization and Performance Analyzer

Develop an application that:

- Implements multiple sorting, searching, graph, and optimization algorithms.
- Visualizes algorithm execution step-by-step.
- Compares execution times and complexities.
- Generates performance charts for different input sizes.
- Provides a user-friendly interface using Python (Tkinte / Streamlit).

Suggested Tools: Python, NetworkX, Matplotlib, Streamlit/Tkinter.

P. V. Sudha

CBoS & Dean | Faculty of Informatics

LCC204**Machine Learning Lab****Credits: 2****Course Objectives:** After completing this laboratory course, students will be able to:

1. Apply data pre-processing, exploratory data analysis, and end-to-end machine learning workflows for solving real-world problems.
2. Develop and evaluate supervised learning models for regression and classification tasks using appropriate performance metrics.
3. Implement decision tree and ensemble learning techniques to improve predictive accuracy and model robustness.
4. Apply dimensionality reduction and unsupervised learning techniques for pattern discovery, clustering, and anomaly detection.
5. Design, train, evaluate, and deploy artificial neural network models using Scikit-Learn, PyTorch, and Streamlit.

Course Outcomes: Upon successful completion of the laboratory course, students will be able to:

1. Perform data pre-processing, feature engineering, exploratory data analysis, and build complete machine learning pipelines for real-world datasets.
2. Implement and evaluate regression and classification models including Linear Regression, Logistic Regression, binary classification, multiclass classification, and multilabel classification techniques.
3. Develop Decision Tree, Random Forest, Bagging, Boosting, and Voting Classifier models and compare their performance using suitable evaluation metrics.
4. Apply Principal Component Analysis (PCA), K-Means Clustering, and Gaussian Mixture Models (GMMs) for dimensionality reduction, clustering, and anomaly detection tasks.
5. Build, train, evaluate, and deploy Artificial Neural Networks and image classification models using Scikit-Learn, PyTorch, and Streamlit for end-to-end machine learning applications.

CO-PO Mapping Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	—	2	—	1	—	—	1
CO2	3	3	3	3	3	—	2	—	1	—	—	1
CO3	3	3	3	3	3	—	2	—	1	—	—	2
CO4	3	3	3	3	3	—	2	—	1	—	—	2
CO5	3	3	3	3	3	—	2	—	2	1	1	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

CBoS & Dean | Faculty of Informatics

Lab Experiments

1. Perform Exploratory Data Analysis (EDA) on a real-world dataset using Pandas and Matplotlib to identify patterns, correlations, and missing values.
2. Implement data pre-processing techniques including handling missing values, feature scaling, encoding categorical variables, and train-test splitting.
3. Build an end-to-end machine learning pipeline that includes data collection, pre-processing, model training, prediction, and performance evaluation.
4. Train a Binary Classifier using the MNIST dataset to distinguish a specific digit and evaluate its performance using confusion matrix, precision, recall, and F1-score.
5. Implement Linear Regression using Scikit-Learn and compare the performance of Ordinary Least Squares and Regularized Regression (Ridge/Lasso).
6. Develop a Logistic Regression model for a classification problem and perform error analysis for multiclass and multilabel classification tasks.
7. Implement a Decision Tree classifier, visualize the generated tree, and analyze the effects of Gini Impurity and Entropy criteria.
8. Develop a Random Forest classifier and compare its performance with a single Decision Tree on a standard dataset.
9. Implement Ensemble Learning techniques including Bagging, Voting Classifiers, and Boosting, and compare their prediction accuracy.
10. Apply Principal Component Analysis (PCA) for dimensionality reduction and compare model performance before and after PCA.
11. Implement K-Means Clustering and determine the optimal number of clusters using the Elbow Method and Silhouette Score.
12. Develop a Gaussian Mixture Model (GMM) for clustering and anomaly detection and compare its performance with K-Means.
13. Build and train a Multi-Layer Perceptron (MLP) using Scikit-Learn for a classification problem and evaluate its performance.
14. Implement Linear Regression using PyTorch and compare the results with the Scikit-Learn implementation.
15. Develop an Image Classification model using PyTorch on the MNIST dataset using mini-batch gradient descent and evaluate the trained model.

Mini Project: End-to-End Machine Learning Application with Deployment

Develop a complete machine learning solution that includes:

- Data Collection and Pre-processing
- Exploratory Data Analysis
- Feature Engineering
- Model Selection and Training
- Hyperparameter Tuning
- Model Evaluation
- Deployment using Streamlit



CBoS & Dean | Faculty of Informatics

Suggested Projects

1. House Price Prediction System
2. Student Performance Prediction
3. Loan Approval Prediction
4. Customer Churn Prediction
5. Disease Prediction System
6. Employee Attrition Prediction
7. Sales Forecasting System
8. Handwritten Digit Recognition System

Tools and Libraries

- Python
- NumPy
- Pandas
- Matplotlib
- Scikit-Learn
- PyTorch
- Streamlit
- Jupyter Notebook / Google Colab

P.V. Sudha

SIP310**Summer Internship / Mini Project****Program Description**

The Internship Program/ Mini Project allows MCA students to gain practical experience in the workplace before receiving their Post graduate degree. The internship is a required academic course. The student identifies companies willing to hire him/her on a full-time basis for 6-week period (minimum required) in the summer vacation. The evaluation committee evaluates the students and awards academic credits (1) in third semester. Those students who wish to do a Mini Project can use Problem statements and Data Sources from good quality sources and implement a solution. The student will be evaluated based on the working system that is presented in Semester III of this program.

Intended Learning Outcomes

1. Communicate a practical understanding of how a technology actually operates.
2. Demonstrate the ability to integrate and apply theoretical knowledge and skills developed in various courses to real-world situations in a business organization.
3. Exhibit the ability to effectively work in a professional environment and demonstrate workethic and commitment in a work-based environment.
4. Demonstrate the ability to successfully complete internship assignments.
5. Reflect on personal and professional development needs and set strategic goals for advancing along an intended career path.
6. Communicate effectively in a professional environment in both English and regional language, orally and in writing.

ILO–PO Mapping Matrix

ILOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ILO1	3	2	2	1	3	–	2	–	1	–	–	1
ILO2	3	3	3	2	3	1	2	2	1	1	2	2
ILO3	1	1	2	–	1	3	2	3	2	2	3	1
ILO4	3	3	3	2	3	2	2	2	2	1	2	2
ILO5	1	1	1	–	1	2	3	2	2	–	1	3
ILO6	–	–	–	–	–	1	2	1	3	1	2	2

Mapping Scale: 3 – High, 2 – Moderate, 1 – Low, – No Mapping

P. V. Sudha

CBoS & Dean | Faculty of Informatics

Model Question Paper Format for CIE
MASTER OF COMPUTER APPLICATIONS (MCA)
Osmania University

M.C.A.-- Semester Examination, -----
 Subject: -----

Time: 1 Hours**Max. Marks: 20**

NOTE: i) Answer all questions from Part – A, & any five questions from Part – B, choosing one question from each unit.

ii) BT-Bloom's Taxonomy and CO-Course Outcome

PART – A (5Q x 2M = 10 Marks)

S. No.	Question	Marks	BT	CO
1.	a Theory/Problem			
	b Theory/Problem			
	c Theory/Problem			
	d Theory/Problem			
	e Theory/Problem			

PART – B (2Q x 5M = 10 Marks)

S. No.	Question	Marks	BT	CO
2	a Theory			
	b Theory/Problem			
OR				
3	a Theory			
	b Theory/Problem			
4	a Theory			
	b Theory/Problem			
OR				
5	a Theory			
	b Theory/Problem			



CBoS & Dean | Faculty of Informatics

Model Question Paper Format for SEE MASTER OF COMPUTER APPLICATIONS (MCA)

Code No:

Osmania University

M.C.A.-- Semester Examination, -----

Subject: -----

Time: 3 Hours**Max. Marks: 70**

NOTE: i) Answer all questions from Part – A, & any five questions from Part – B, choosing one question from each unit.

ii) BT-Bloom's Taxonomy and CO-Course Outcome

PART – A (10Q x 2M = 20 Marks)

SNo	Question	Marks	BT	CO
1. a	Unit- I			
b	Unit- I			
c	Unit- II			
d	Unit- II			
e	Unit- III			
f	Unit- III			
g	Unit- IV			
h	Unit- IV			
i	Unit- V			
j	Unit- V			

PART – B (5Q x 10M = 50 Marks)

SNo	Question	Marks	BT	CO
Unit-I				
2a	Theory			
2b	Theory/Problem			
	OR			
3a	Theory			
3b	Theory/Problem			
Unit-II				
4a	Theory			
4b	Theory/Problem			
	OR			
5a	Theory			
5b	Theory/Problem			
Unit-III				
6a	Theory			
6b	Theory/Problem			
	OR			
7a	Theory			
7b	Theory/Problem			



CBoS & Dean | Faculty of Informatics

Unit-IV				
8a	Theory			
8b	Theory/Problem			
	OR			
9a	Theory			
9b	Theory/Problem			
Unit-V				
10a	Theory			
10b	Theory/Problem			
	OR			
11a	Theory			
11b	Theory/Problem			

P.V. Sudha