SHRI GURU RAM RAI UNIVERSITY

[Estd. by Govt. of Uttarakhand, vide Shri Guru Ram Rai University Act no. 03 of 2017 & recognized by UGC u/s (2f) of UGC Act 1956]

CURRICULUM & SYLLABI
FOR

Master of Computer Application (MCA)

School of Engineering & Technology

(w.e.f. 2025-2026)

Master of Computer Application (MCA)

Eligibility for admission:

Passed B.C.A/ B.Sc. (Computer Science)/ B.Sc. (IT) / B.E. (CSE)/ B.Tech. (CSE) / B.E. (IT) / B.Tech. (IT) or equivalent Degree.

OR

Passed any graduation degree (e.g.: B.E. / B.Tech. / B.Sc / B.Com. / B.A./ B. Voc./ etc.,) preferably with Mathematics at 10+2 level or at Graduation level Obtained at least 50% marks (45% marks in case of candidates belonging to reserved category) in the qualifying examination.

(For students having no Mathematics background compulsory **bridge course** will be framed by the respective University/ Institution and additional bridge courses related to computer subjects as per the norms of the concerned University).

Duration of the Programme: 2 years

Examination Scheme:

Components	Internal	Assignment & Teacher Assessment	External (ESE)
Weightage (%)	20	20	60

Programme outcome (POs)

Students will be able to

PO1	Technical Expertise	Acquire knowledge of Computing Fundamentals, Basic Mathematics, Computing Specialization and Domain Knowledge of proper computing models from defined problems.
PO2	Problem Analysis	Identify, formulate and analyze complex engineering problems reading substantiated conclusions using first principles mathematics, computer science and relevant domains.
PO3	Software Development	Ability to design efficient solution for complex, real-life problem, system software or process as per needs and specifications.
PO4	Research-Based Problem Solving	Use research-based knowledge and research methods including design of experiments, analysis & interpretation of data & synthesis of information to provide valid conclusions.
PO5	Modern Technology and Tool	Ability to demonstrate skills to use modern technologies and tools to analyse problems.
PO6	Ethical Practices	Ability to perform professional practices in an ethical way, keeping in the mind cyber regulations & laws, responsibilities and norms of professional computing practices.
PO7	Lifelong Learning	Ability to develop confidence for self-education and life-long learning in the broadest context of technological change
PO8	Team Projects	Ability to demonstrate knowledge & understanding of the engineering and management principles and apply them as a member & as a leader in a team to manage multidisciplinary projects.
PO9	Communication Skills	Ability to effectively communicate with the technical community and with the society about complex computing activities in both verbal and written form, design documentation, make effective presentations.
PO10	Societal and Global Impact of IT Solutions	Ability to understand the impact of IT solutions in a global and societal context.

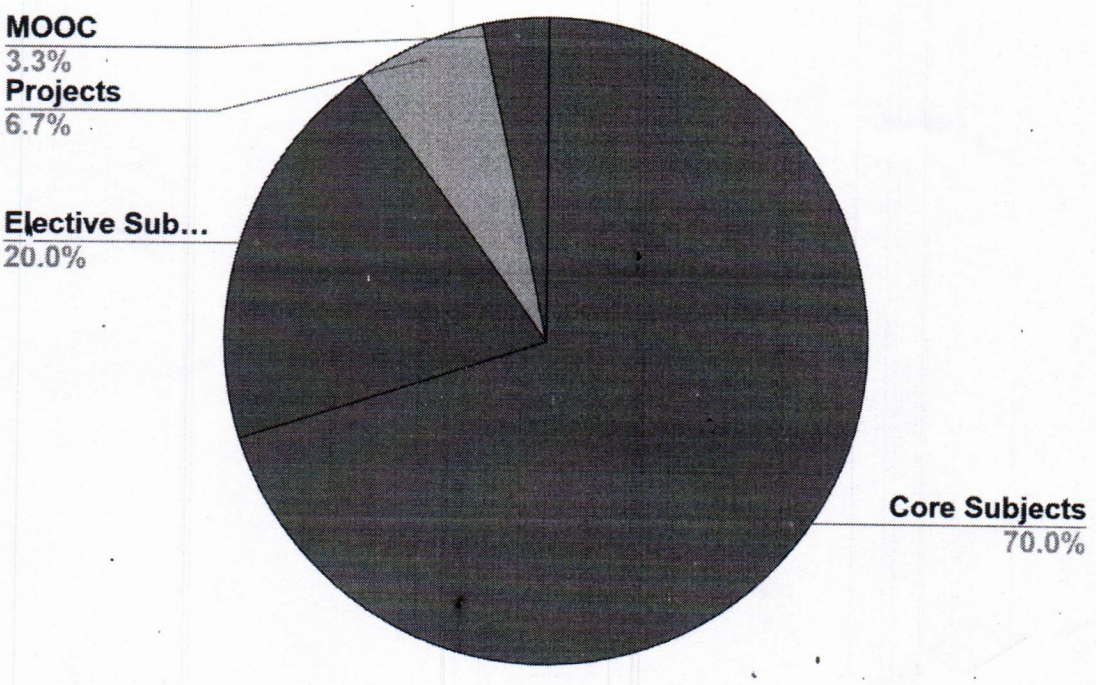
PO11	Team Collaboration	Ability to work multi-disciplinary team both as a member and leader, as per need.
PO12	Societal Betterment	Ability to apply innovation to a suitable opportunity to create value and wealth for the betterment of the individual and society at large.

Program Specific Outcome (PSOs)

PSO1	Technical Proficiency	Design, develop, and implement efficient software solutions using modern computing tools and technologies to address real-world problems.
PSO2	Problem-Solving Skills	Apply advanced computational techniques and algorithms to analyze complex problems and provide innovative, scalable solutions.
PSO3	Professional Competence	Demonstrate expertise in software development lifecycle, project management, and teamwork to deliver high-quality IT solutions in diverse domains.
PSO4	Research and Innovation	Utilize emerging technologies and research methodologies to create sustainable, secure, and ethical IT solutions for industry and societal needs.

Course Structure and Credit Distribution

Core Subjects	Elective Subjects	Projects	MOOC
21	6	2	1



Semester wise Course Structure and Credit Distribution

	Core Subjects	Elective Subjects	Projects	MOOC	Total
Semester I	22				22
Semester II	18	4			22
Semester III	10	10	2		22
Semester IV			14	4	18
	50	**14**	**16**	**4**	**84**

General Course Structure and Theme:

Definition of Credit:

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

Course Code and Definition:

Course Code	Definitions
L	Lecture
T	Tutorial
P	Practical
C	Credits
Core	Core Subject
Elective	Elective Subject
SM	Seminar
PR	Project
RP	Research project

STUDY & EVALUATION SCHEME
Choice Based Credit System
Master of Computer Application (MCA)

First Semester

S. No.	Course Category	Couse Code	Course Name	Periods				Evaluation scheme		Subject Total
				L	T	P	C	Sessional (Internal)	External (ESE)	
Theory										
1	Core	MCA101	Computer Organization & Architecture	3	0	0	3	40	60	100
2	Core	MCA102	Python	3	0	0	3	40	60	100
3	Core	MCA103	Mathematics for Data Science	3	0	0	3	40	60	100
4	Core	MCA104	Operating System	3	0	0	3	40	60	100
5	Core	MCA105	Cyber Security	3	0	0	3	40	60	100
6	Core	MCA106	Professional Communications	2	1	0	3	40	60	100
7	Core	MCA107	Human Values and Ethics	2	0	0	2	40	60	100
Practical										
8	Core	MCAP12	Python Lab	-	-	2	1	40	60	100
9	Core	MCAP14	Linux Lab	-	-	2	1	40	60	100
			Total	**19**	**1**	**4**	**22**	**360**	**540**	**900**

Second Semester

S. No.	Course Category	Couse Code	Course Name	Periods				Evaluation scheme		Subject Total
				L	T	P	C	Sessional (Internal)	External (ESE)	
Theory										
1	Core	MCA201	Data Structure	3	0	0	3	40	60	100
2	Elective	MCA202.1	Data Science using Python	3	0	0	3	40	60	100
		MCA202.2	Java Programming							
3	Core	MCA203	DBMS	3	0	0	3	40	60	100
4	Core	MCA204	Software Engineering	3	0	0	3	40	60	100
5	Core	MCA205	Algorithm Analysis & Design	3	0	0	3	40	60	100
6	Core	MCA206	Computer Network	3	0	0	3	40	60	100
Practical										
7	Core	MCAP21	Data Structure Lab	-	-	2	1	40	60	100
8	Elective	MCAP22.1	Data Science Lab	-	-	2	1	40	60	100
		MCAP22.2	Java Lab							
9	Core	MCAP23	SQL Lab	-	-	2	1	40	60	100
10	Core	MCASM24	Seminar and Presentation	-	-	2	1	100	-	100
			Total	**18**	**0**	**8**	**22**	**460**	**540**	**1000**

Third Semester

S. No.	Course Category	Couse Code	Course Name	Periods				Evaluation scheme		Subject Total
				L	T	P	C	Sessional (Internal)	External (ESE)	
Theory										
1	Core	MCA301	C# Dot Net	3	0	0	3	40	60	100
2	Core	MCA302	Network Security & Cryptography	3	0	0	3	40	60	100
3	Core	MCA303	Artificial Intelligence	3	0	0	3	40	60	100
4	Elective	MCA304.1	Machine Learning	3	0	0	3	40	60	100
		MCA304.2	Full Stack Web Development							
5	Elective	MCA305.1	Introduction to Deep Learning	3	0	0	3	40	60	100
		MCA305.2	Blockchain Technology							
		MCA305.3	Cloud Computing							
6	Elective	MCA306.1	Research Methodology and Publication Ethics	3	0	0	3	40	60	100
		MCA306.2	Software Project Management							
		MCA 306.3	Theory of Automata							
Practical										
7	Core	MCAP31	C# Dot Net Lab	-	-	2	1	40	60	100
8	Elective	MCAP34.1	Machine Learning Lab	-	-	2	1	40	60	100
		MCAP34.2	Full Stack Web Development Lab							
9	Project	MCAPR3	Project/Research Project	-	-	4	2	40	60	100
			Total	**18**	**0**	**8**	**22**	**360**	**560**	**900**

Fourth Semester

S. No.	Course Category	Couse Code	Course Name	Periods				Evaluation scheme		Subject Total
				L	T	P	C	Sessional (Internal)	External (ESE)	
Practical										
1	MOOC	MCA401	MOOC	-	-	-	04	25	75	100
2	Project	MCAPR4	Project/Research Project	-	-	28	14	200	400	600
			Total	-	-	**28**	**18**	**225**	**475**	**700**
Total (1st to 4th Semester)							**84**			**3500**

- Students are required to complete a 12-week MOOC course (equivalent to 4 credits) on a latest technology topic **not included in their current curriculum**, through the SWAYAM portal or any other Government Portal.
- Prior approval must be obtained from the department before enrolling.
- Upon completion, students must submit the course certificate and marksheet to the department for the credit to be transferred to their MCA program.
- Marksheet will reflect the Course Title of MOOC course done by student.

Course code	**: MCA101**				
Course Name	**: COMPUTER ORGANIZATION & ARCHITECTURE**				
Semester /Year	**: Ist Semester**				
	L	**T**	**P**	**C**	
	3	**0**	**0**	**3**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Understand the fundamental components and operations of computer systems, including CPU, memory, I/O, and basic digital electronics concepts like logic gates and Karnaugh Maps.
2. Explain the structure and function of processors, including instruction sets, addressing modes, and the fetch-decode-execute cycle.
3. Describe the memory hierarchy, including cache, main memory, and secondary storage, and their roles in computer performance.
4. Identify input/output mechanisms, such as programmed I/O, interrupt-driven I/O, and DMA, and their theoretical applications in computer systems.
5. Recognize basic computer architectures and emerging trends like multicore processors and cloud computing hardware.

COURSE CONTENTS

Unit 1: Introduction to Computer Systems and Digital Electronics (8 Hours)

Components of a computer, evolution of computers, binary, octal, hexadecimal number systems, number system conversions, logic gates, truth tables, Karnaugh Maps (K-Map) for logic simplification.

Unit 2: Processor Fundamentals (10 Hours)

Structure of CPU, registers, ALU, control unit, types of instructions, addressing modes, fetch-decode-execute cycle, introduction to pipelining.

Unit 3: Memory Organization (8 Hours)

Cache memory, main memory, secondary storage, memory types, cache mapping techniques, cache coherence, basics of memory organization, SRAM vs. DRAM.

Unit 4: Input/Output Systems (8 Hours)

Types of I/O devices, I/O ports, buses, programmed I/O, interrupt-driven I/O, DMA, role of I/O in computer systems.

Unit 5: Computer Architecture and Emerging Trends (6 Hours)

Von Neumann architecture, introduction to parallel processing, multicore processors, cloud computing hardware basics.

Text Books:

TB1. Computer System Architecture, PHI/Pearson Education, M. Morris Mano
TB2. Digital Computer Fundamentals, Tata McGraw Hill, Thomas C. Bartee

Reference Books:

RB1. Computer Organization, Vravice, Zaky & Hamacher (TMH Publication)
RB2. Structured Computer Organization, Tannenbaum(PHI)

RB3. Computer Organization, Stallings(PHI)
RB4. William Stalling, "Computer Organization & Architecture" ,Pearson Educaiton Asia

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Recall fundamental concepts of digital electronics, number systems, and basic computer components.
CO2	Explain the working of CPU components, instruction execution, and memory hierarchy.
CO3	Use logic gates, K-Maps, and addressing modes to solve computational problems.
CO4	Compare different memory technologies, cache mapping techniques, and I/O mechanisms.
CO5	Assess the efficiency of pipelining, cache coherence, and interrupt-driven I/O systems.
CO6	Design optimized logic circuits and memory architectures using learned principles.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1
CO2	3	2	1	1	2	1	1	1	1	1	1	1	2	2	1	1
CO3	3	3	2	1	2	1	1	1	1	1	1	1	3	3	2	1
CO4	3	3	2	2	2	1	1	1	1	1	1	1	3	3	2	1
CO5	3	3	2	2	3	1	1	1	1	1	1	1	3	3	2	2
CO6	3	3	3	3	3	1	2	1	1	1	1	1	3	3	3	3
AVG	3	2.5	1.8	1.6	2.3	1	1.1	1	1	1	1	1	2.6	2.5	1.8	1.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA102**				
Course Name	**: PYTHON**				
Semester /Year	**: Ist Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Master the core of writing Python scripts
2. Understand decision-making and functions in python
3. Interpret Object-oriented programming features in python
4. Gain knowledge of data structures in python
5. Explore file handling and database operations in python

COURSE CONTENTS

Unit 1: Introduction to Python and Basic Concepts (8 hours)
Python programming fundamentals, Python's features, and applications, installation of Python 3.x and setup of IDLE or VS Code, variables, data types, and type conversion, input/output with print() and input(), operators, string operations, creation and execution of Python scripts

Unit 2: Control Structures and Functions (8 hours)
Control flow and modularity, conditional statements, loops with control statements (break, continue), nested loops and conditionals, function definition with parameters and return values, variable scope (local, global), default, keyword, and variable-length arguments, lambda functions, basic error handling with try-except.

Unit 3: Data Structures in Python (8 hours)
Python's built-in data structures, including list creation, indexing, slicing, and methods, tuple creation and immutability, dictionary operations for key-value pairs, set operations nested data structures, list comprehensions, iteration over data structures, applications of each data structure.

Unit 4: File Handling and Modules (8 hours)
File operations and code organization, covering file handling, file modes, CSV file processing with the csv module, Python modules and packages, importing standard modules, creation of custom modules, overview of the Python Standard Library, file-related error handling.

Unit 5: Object-Oriented Programming and Advanced Concept (8 hours)
Object-oriented programming and features, including class and object creation, class attributes and methods, constructors (init) and instance variables, inheritance, polymorphism, encapsulation with public and private attributes, multi-class relationships, regular expressions with the re module.

Text Books:

TB1. Budd T A, "Exploring Python", McGraw-Hill Education.
TB2. Mark Lutz, "Learning Python", O'Reilly.

TB3. Y. Daniel Liang, "Introduction to Programming Using Python", Pearson

Reference Books:

RB1. Kenneth A. Lambert, "The Fundamentals of Python: First Programs", Cengage Learning

RB2. Allen Downey, "Think Python: How to Think Like a Computer Scientist", O'Reilly

RB3. Reema Thareja, "Python Programming using Problem Solving Approach", Oxford University Press

RB4. Joel Grus, "Data Science from Scratch", O'Reilly

RB5. Tony Gaddis, "Starting out with Python", Pearson

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Knowledge and remember the programming constructs used in python.
CO2	Understanding the facts behind the sequences, functions, modules, files, database and object oriented etc. used in Python
CO3	Apply data structure primitives like strings, lists, tuples, sets and dictionaries on various types of data with or without using functions, object-oriented concepts to the programs in Python etc.
CO4	Distinguish and analyze basic constructs of Python and how constructs can be used all together.
CO5	Evaluate the programming constructs of Python to provide verdict on findings.
CO6	Create python programs using various programming constructs of Python.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	1	2	1	1	1	1	1	2	1	1	1
CO2	3	2	2	1	2	1	2	1	1	1	1	1	3	2	1	1
CO3	3	3	3	2	3	1	2	2	1	2	2	1	3	3	2	2
CO4	2	3	2	2	2	2	2	1	2	2	1	1	2	3	1	2
CO5	2	3	2	3	2	3	2	1	2	2	1	2	2	3	1	3
CO6	3	3	3	3	3	2	3	3	3	3	3	2	3	3	3	3
AVG	2.7	2.5	2.2	2	2.3	1.7	2.2	1.5	1.7	1.8	1.5	1.3	2.5	2.5	1.5	2

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA103**				
Course Name	**: MATHEMATICS FOR DATA SCIENCE**				
Semester /Year	**: Ist Semester**				
	L	**T**	**P**	**C**	
	3	**0**	**0**	**3**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives:

1. To study methods of collection, classification, tabulation of data.
2. To apply various statistical methods and analyses the data
3. To find correlation between the data.
4. To study probability and probability distribution to solve various real life problems
5. To test hypothesis using various statistical test.

COURSE CONTENTS

Unit 1: Introduction to Statistics and Data Collection (8 hours)
Definition and scope of statistics, types of data (qualitative vs. quantitative, discrete vs. continuous), data collection methods (surveys, experiments, observational studies), sampling techniques (random, stratified, cluster, systematic), data organization (frequency distributions, histograms, bar charts), measures of central tendency (mean, median, mode), practical exercises using real-world datasets.

Unit 2: Descriptive Statistics and Data Visualization (8 hours)
This unit focuses on summarizing and visualizing data through statistical measures and graphical tools. Topics include measures of dispersion (range, variance, standard deviation, interquartile range), percentiles and quartiles, data visualization (box plots, scatter plots, pie charts, stem-and-leaf plots), skewness and kurtosis, practical exercises using software (e.g., Excel, R, or Python) for descriptive statistics and visualization.

Unit 3: Probability and Probability Distributions (8 hours)
This unit explores the principles of probability and key probability distributions. Topics include basic probability concepts (events, sample spaces, probability rules), conditional probability and independence, Bayes' theorem, random variables (discrete and continuous), common distributions (binomial, Poisson, normal, exponential), practical exercises solving probability problems and simulating distributions.

Unit 4: Statistical Inference (8 hours)
This unit covers techniques for making inferences about populations based on sample data. Topics include point estimation and interval estimation, confidence intervals for means and proportions, hypothesis testing (null and alternative hypotheses, p-values, significance levels), types of errors (Type I and Type II), tests for means, proportions, and variances (t-tests, z-tests, chi-square tests), practical exercises conducting hypothesis tests and constructing confidence intervals using software.

Unit 5: Correlation, Regression, and Advanced Topics (8 hours)
Correlation (Pearson and Spearman correlation coefficients), simple linear regression (model fitting, interpretation, diagnostics), multiple regression (concepts and applications).

introduction to non-parametric tests, time series analysis (basic concepts and trends), practical exercises analyzing datasets with regression models and non-parametric tests using software.

Text Books:

TB1. Gerald Keller : Managerial Statistics

Reference Books:

RB1. Richard Levin & David Rubin : Statistics for management, Prentice Hall.

RB2. Anderson, Sweeny & Williams: Statistics for Business and Economics, South W

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall fundamental statistical concepts, data types, and measures of central tendency/dispersion
CO2	Explain probability distributions, sampling techniques, and their applications in real-world scenarios
CO3	Calculate descriptive statistics (mean, variance, etc.) and probabilities for given datasets.
CO4	Interpret frequency distributions, correlation coefficients, and regression results to identify patterns.
CO5	Compare the appropriateness of different statistical methods (e.g., t-tests vs. z-tests) for specific
CO6	Construct frequency tables, histograms, or box plots to summarize and visualize data effectively.

CO –PO Mapping:

	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	-	-	-	-	-	-	-	-	-	-	1	-	-	-
CO2	3	2	-	1	-	-	1	-	-	1	-	-	2	1	-	1
CO3	3	3	-	2	-	-	1	-	-	-	-	-	2	2	-	1
CO4	3	3	1	3	-	-	2	-	1	1	-	-	2	3	1	2
CO5	2	3	-	3	-	1	2	-	1	2	-	1	1	3	-	3
CO6	2	2	2	2	-	-	1	1	2	1	1	-	2	2	2	2
AVG	2.5	2.3	0.5	1.8	0	0.2	1.2	0.2	0.7	0.8	0.2	0.2	1.7	1.8	0.5	1.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA104**				
Course Name	**: OPERATING SYSTEM**				
Semester /Year	**: Ist Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives:

1. To study types of Operating System and functions performed by operating system
2. To know Process & Process Management.
3. To learn CPU Scheduling and Process Synchronization.
4. To study Deadlock and its prevention and avoidance.
5. To study Memory management, Virtual Memory.
6. To learn File Management & Disk Management.

COURSE CONTENTS

UNIT 1: Introduction to Operating System (8 hours)
Definition, Types of OS- Simple batch system, Time sharing systems, Real time systems, Multiprocessor systems, Distributed systems, Network based operating system.
System components - OS Services, System Calls, Utility programs

UNIT 2: Process concepts (8 hours)
Process, Process States, Process Transition, PCB, CPU Scheduling Algorithms(preemptive and Non preemptive(FCFS, SJF, RR ,Priority Scheduling), Co-operating process, IPC(Pipes, Signal, Message passing) , Threads- Overview, Benefits, User & Kernel Threads.

UNIT 3: Process Synchronization Deadlock (8 hours)
Critical Section problem, Critical Regions, Synchronization hardware, Classic Problems of Synchronization.
Semaphores, Monitor, swap Instruction, Test and Set Instruction.
Deadlocks: Deadlock and its necessary conditions, Methods for Handling Deadlocks, Deadlock prevention, RA Graph, Deadlock Avoidance, Banker Algorithm, Deadlock Detection and Recovery from Deadlock.

UNIT 4: Memory Management (8 hours)
Logical vs. Physical address space, Contiguous memory allocation, Non-Contiguous memory allocation- Paging, Segmentation, Swapping, Segmentation with paging.
Virtual Memory: Demand paging - Performance, Page replacement, Page replacement algorithms (FCFS, LRU, Optimum), Allocation of frames, Thrashing.

UNIT 5: File Systems and Directory Management (8 hours)
File concept, access methods, Allocation methods-contiguous, linked and index allocation, File protection, File permission, Directory **System** – single level, tree structured, Absolute path and Relative path.
Disk Management: Disk structures, Disk Scheduling, Disk Performance Parameter.

Text Books:

TB1. Abraham Silberschatz, Peter Baer Galvin & Greg Gagne, "Operating System Concepts", Sixth Edition, John Wiley & Sons, Inc.

Reference Books:

RB1. Milankovic M "Operating System concepts and Design", Tata Mcgraw hill.

RB2. Deitel H.M. "An Introduction to Operating Sysems" Pearson Education.

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall the definitions, types, and components of operating systems, including system calls and utility programs,
CO2	Explain the concepts of process states, CPU scheduling algorithms, and inter-process communication mechanisms
CO3	Implement process synchronization techniques and deadlock handling methods, such as semaphores and Banker's Algorithm
CO4	Compare and analyze memory management techniques, including paging, segmentation, and page replacement algorithms
CO5	Assess the effectiveness of disk scheduling algorithms and file allocation methods for optimizing system performance
CO6	Design a solution for a given operating system problem, integrating concepts like virtual memory or file system design

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1					1						1	1		
CO2	3	2		1	1		2		1				1	2		1
CO3	2	3	3	2	2	1	2	1			1		3	3	2	2
CO4	2	3	2	2	2		2			1			2	3	1	1
CO5	2	3	2	2	2	1	2		1	2		1	2	3	2	2
CO6	2	3	3	3	2	2	2	2	2	2	2	2	3	3	3	3
AVG	2.3	2.5	2.5	2	1.8	1.3	1.8	1.5	1.3	1.7	1.5	1.5	2	2.5	2	1.8

Course code	**: MCA105**				
Course Name	**: CYBER SECURITY**				
Semester /Year	**: Ist Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives:

1. Introduce foundational cybersecurity concepts, including threats, vulnerabilities, and the CIA triad (Confidentiality, Integrity, Availability).
2. Teach practical security measures, such as configuring firewalls, encryption, and secure web application practices.
3. Develop skills in intrusion detection and prevention, including log analysis and using tools like Snort or WAFs.
4. Enable evaluation of security policies through risk assessment and best practices for system hardening.
5. Prepare students to design security solutions, integrating principles like defense-in-depth and incident response for real-world scenarios.

COURSE CONTENTS

Unit 1: Introduction to Cybersecurity (8 Hours)
Definition of cybersecurity, its importance, and real-world applications, the CIA triad (Confidentiality, Integrity, Availability), types of cyber threats including malware (viruses, worms, ransomware), phishing, and social engineering, cybersecurity terminology such as assets, threats, risks, and exploits, principles like defense-in-depth and least privilege

Unit 2: Cybersecurity Vulnerabilities and Cybersecurity Safeguards (8 Hours)
Types of vulnerabilities including software bugs, weak passwords, and misconfigurations, common cyber threats like ransomware, spyware, and DDoS attacks, cybersecurity safeguards including firewalls, antivirus software, and encryption, introduction to access control and user authentication.

Unit 3: Securing Web Applications, Services, and Servers (8 Hours)
Basics of web applications and services (HTTP, HTTPS, APIs), common web vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), securing web servers (e.g., Apache, Nginx) through configuration best practices, introduction to secure coding principles like input validation and sanitization, use of web application firewalls (WAF), SSL/TLS for secure communication.

Unit 4: Intrusion Detection (8 Hours)
Introduction to intrusion detection systems (IDS) and their types (network-based, host-based), common signs of intrusion like unusual network traffic or system behavior, tools for intrusion detection such as Snort or Suricata, log analysis and monitoring basics, understanding false positives and negatives in detection, incident identification and initial response steps.

Unit 5: Prevention Systems (8 Hours)
Introduction to intrusion prevention systems (IPS) and their role in blocking threats, implementing network security controls like VPNs and network segmentation, endpoint

protection strategies using antivirus and anti-malware tools, multi-factor authentication (MFA) and strong password policies, basics of incident prevention planning and security policies, emerging prevention techniques like zero-trust architecture.

Text Books:

TB1. Cyber security Essentials, Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, Sybex

Reference Books:

RB1. Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, B. B. Gupta, D. P. Agrawal, Haoxiang Wang, CRC Press

RB2. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall fundamental cybersecurity concepts, including the CIA triad, common threats (e.g., malware, phishing), and key terminology (e.g., vulnerabilities, exploits)
CO2	Explain how cybersecurity safeguards (e.g., firewalls, encryption) mitigate specific vulnerabilities and threats.
CO3	Implement basic security measures such as configuring a firewall rule, enabling HTTPS on a web server, or applying input validation to prevent SQL injection.
CO4	Differentiate between types of intrusions (e.g., network-based vs. host-based) by analyzing logs or IDS alerts
CO5	Assess the effectiveness of a given security policy or system (e.g., password strength, network segmentation) using cybersecurity best practices.
CO6	Design a comprehensive security plan for a small organization, integrating principles like defense-in-depth, least privilege, and incident response.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1
CO2	3	2	1	1	2	3	1	1	1	2	1	1	2	1	1	2
CO3	3	2	3	1	3	2	2	2	1	1	2	1	3	2	2	2
CO4	3	3	2	2	3	1	1	1	2	1	1	1	2	3	1	2
CO5	2	3	2	2	3	3	2	1	2	3	1	2	2	3	2	3
CO6	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
AVG	2.7	2.3	2	1.7	2.5	2.5	1.7	1.5	1.7	1.8	1.5	1.5	2.2	2.2	1.7	2.2

Course code	**: MCA106**				
Course Name	**: PROFESSIONAL COMMUNICATION**				
Semester /Year	**: Ist Semester**				
		L	**T**	**P**	**C**
		2	**1**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objective:

1. Enhance the Employability and Career Skills of students
2. Orient the students towards grooming as a professional.
3. Make them Employable.
4. Develop their confidence and help them attend interviews successfully and achieve growth by acquiring professionalism as a habit.

COURSE CONTENTS

UNIT I (8 hours)

Introduction to Soft Skills– Hard skills & soft skills – employ-ability and career Skills—Grooming as a professional with values—Time Management—Conflict management, Anger Management-Stress Management

UNIT II (8 hours)

Self-Introduction-organizing the material – Written communication -Introducing oneself to the audience – introducing the topic – answering questions – individual presentation practice—presenting the visuals effectively – 5-minute presentations

UNIT III (8 hours)

Introduction to Group Discussion— Participating in group discussions – understanding group dynamics – brainstorming the topic — questioning and clarifying –Group discussion strategies-activities to improve Group discussion skills

UNIT IV (8 hours)

Social etiquette, Interview etiquette – dress code – body language

UNIT V (8 hours)

Interview, types attending job interviews– telephone/online interview -one to one interview &panel interview –CV writing, Job application, FAQs related to job interviews

Text Books:

TB1. Effective Communication and soft skills Author Nitin Bhatnagar and Mamta Bhatnagar, publisher Pearson

TB2. Basic Communication skills for technology Author-Rutherford, Publisher -Pearson Publication

TB3. Business Communication Author N Gupta, Publisher -Sathya Bhawna Publication

TB4. Comprehension and communication skills Author-Varinder Kumar. Publisher Kalyani

Reference Books:

RB1. English communication Author Amit Ganguly, Publisher -SBPD publication

RB2. The art and science of business communication fourth edition, Author -PD Chaturvedi Mukesh Chaturvedi, Publisher Pearson

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO 1	Recall key concepts of soft skills, professional etiquette,, and interview techniques to enhance employability.
CO 2	Explain the principles of effective self-introduction, group discussion dynamics, and stress management in professional settings.
CO 3	Apply appropriate verbal and non-verbal communication techniques, including body language and dress code, in mock interviews and presentations.
CO 4	Analyze group discussion scenarios to identify effective strategies and clarify responses for improved participation.
CO 5	Evaluate the effectiveness of personal communication styles and professional behaviors in simulated job interviews and group discussions.
CO6	Create a professional CV and deliver a structured 5-minute presentation incorporating visuals and ethical communication practices.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1						2	2	1	3	1	1	1			2	
CO2						2	2	1	3	1	1	1			2	
CO3						2	1	2	3	2	2	1			3	
CO4		1		1		2	1	2	3	2	2	1		1	2	1
CO5		1		1		2	1	2	3	2	2	1		1	2	1
CO6					1	2	1	2	3	2	2	2	1		3	1
AVG	0	0.3	0	0.3	0.2	2	1.3	1.7	3	1.7	1.7	1.2	0.2	0.3	2.3	0.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA107**				
Course Name	**: HUMAN VALUES AND ETHICS**				
Semester /Year	**: Ist Semester**				
	L	**T**	**P**	**C**	
	2	**0**	**0**	**2**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objective:

1. Develop an understanding of core human values and their role in personal and societal growth.
2. Introduce fundamental ethical theories and frameworks for analyzing moral dilemmas.
3. Apply ethical principles to real-world professional, social, and global challenges.
4. Enhance critical thinking to evaluate ethical issues in technology and society.
5. Foster responsible decision-making by integrating values and ethics into daily life.

COURSE CONTENTS

Unit 1: Introduction to Human Values (6 Hours)

Definition and importance of human values, core human values such as truth, respect, compassion, justice, and responsibility, the role of values in personal and societal development.

Unit 2: Understanding Ethics (7 Hours)

Introduction to ethics, its meaning and types such as personal, professional, and societal, **Ethical theories** like utilitarianism, deontology, and virtue ethics,
ethical dilemmas and decision-making frameworks, and ethics in everyday life such as honesty, fairness, and accountability.

Unit 3: Application of Values and Ethics (7 Hours)

Professional ethics such as integrity, transparency, and workplace responsibility,
Social ethics such as respect for diversity, empathy, and community welfare,
Balancing personal values with professional obligations, and global ethical challenges like sustainability and social justice.

Text Books:

TB1. A Foundation Course in Human Values and Professional Ethics by R.R. Gaur, R. Asthana, and G.P. Bagaria

TB2. Professional Ethics and Human Values by M. Govindarajan, S. Natarajan, and V.S. Senthilkumar

Reference Books:

RB1. Human Values and Ethics: Achieving Holistic Excellence by and Debangshu Chakraborty, S.K. Chakraborty, Icfai University Press

RB2. Human Values and Ethics in the Workplace by Glenn Martin

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO 1	Recall and define key human values (truth, respect, justice) and ethical theories (utilitarianism, deontology)
CO 2	Explain the role of values in personal growth and societal development, and differentiate between personal, professional, and societal ethics.
CO 3	Use ethical decision-making frameworks to resolve moral dilemmas in personal and professional contexts.
CO 4	Compare and contrast ethical theories, evaluating their strengths and weaknesses in real-world scenarios.
CO 5	Assess the ethical implications of workplace actions and social issues (e.g., sustainability, diversity) using moral reasoning.
CO6	Design a personal code of ethics integrating core values and ethical principles for responsible citizenship.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO2	-	-	-	-	-	3	1	-	1	2	-	1	-	-	-	1
CO3	-	-	-	-	-	3	1	1	1	2	1	2	-	-	1	2
CO4	-	-	-	1	-	3	1	-	1	2	-	2	-	-	-	2
CO5	-	-	-	1	-	3	1	-	1	3	-	3	-	-	-	3
CO6	-	-	-	2	-	3	2	-	2	3	-	3	-	-	1	3
AVG	-	-	-	1.3	-	2.8	1.2	0.2	1.2	2.3	0.2	2.2	-	-	0.3	2.2

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAP12**				
Course Name	**: PYTHON LAB**				
Semester /Year	**: Ist Semester**				
		L	**T**	**P**	**C**
		0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Basic programming constructs and functions in python.
2. Understand the applicability of data structures like lists, tuples, sets and dictionaries inpython applications.
3. Use object-oriented programming features of python to develop applications.
4. Learn how to use exception handling in applications for error handling.
5. Database and file based programming

COURSE CONTENTS

- Working with Jupyter notebook.
- Programs based on loops and conditional statements.
- Programs based on string manipulations.
- Programs based on List, tuples, sets and dictionary.
- Working with user defined functions.
- Working with lambda, map, filter and reduce functions.
- Programs based on recursion.
- Programs for file handling in Python.
- Programs for Sorting and searching.
- Database handling in Python using SQLite3.
- Working with in built and user defined modules,
- Working with Object Oriented Programming in Python.

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Describe the program creation in Python through usage of appropriate constructs
CO2	Demonstrate the working of basic programming constructors in Python.
CO3	Apply data structure primitives like strings, lists, tuples, sets and dictionaries on various types of data with or without using functions, object-oriented concepts to the programs in Python etc.
CO4	Analyze basic constructs of Python and how constructs can be used all together.
CO5	Evaluate the programs and its logic.
CO6	Develop programs using methods of constructs define in Python.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	1	2								3		1	1
CO2	3	3	3	2	2	1	2				1		3	1		1
CO3	1	3	3	1	1								3			1
CO4	3	3	3	3	3	3	1				1		3	2	2	2
CO5	3	2	2	2	1								3		2	2
CO6	3	3	3	3	1	1	1	1	1	1	1	1	3	1	1	2
AVG	**2.7**	**2.7**	**2.8**	**2.0**	**1.7**	**1.7**	**1.3**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**3.0**	**1.3**	**1.5**	**1.5**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAP14**				
Course Name	**: LINUX LAB**				
Semester /Year	**: Ist Semester**				
		L	**T**	**P**	**C**
		0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives:

The objectives of this course are

1. To learn basic knowledge about architecture of Linux and different basic Commands of Linux
2. To learn how to use process management.
3. To learn basic structure and various commands of LINUX system.
4. To learn the importance of system administration tasks.
5. To learn the shell programming.
6. To learn basics of filter commands and Tools of LINUX system

COURSE CONTENTS:

- How to apply basic commands of LINUX
- How to apply filter commands.
- How to apply LINUX system administration commands
- Unix shell program based on conditional statements
- Unix shell program based on loops
- Unix shell program based on filters
- Unix shell program based on arrays

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Able to acquire knowledge and remember basic commands of LINUX and shell programming constructs
CO2	Able to understand basic commands of LINUX and shell programming constructs.
CO3	To apply basic commands of LINUX and shell programming constructs.
CO4	To analyse difference between basic commands of LINUX and shell programming constructs
CO5	Able to evaluate expressions using basic commands of LINUX and shell programming constructs.
CO6	Able to create applications/software using shell programming constructs

CO-PO Mapping:

	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1		1			1					3			1
CO2	3	1	2	1		2	1						3			1
CO3	1	1	3		1		2		2				2		1	1
CO4	2	1	1	1		2				1	2		3			1
CO5	2	1	2		1		1					2	3			1
CO6		1				2						2				
AVG	2.2	1	1.8	0.4	0.6	1	0.8	0.2	0.4	0.2	0.4	0.66	2.8	0	0.2	1

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

2nd Semester

Course code	**:**	**MCA201**			
Course Name	**:**	**DATA STRUCTURE**			
Semester /Year	**:**	**IInd Semester**			
	L	**T**	**P**	**C**	
	3	**0**	**0**	**3**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To learn basic knowledge about data structure and arrays.
2. To learn how to create and use linked list and its applications.
3. To learn the importance of static, and dynamic use of stack and queues.
4. To learn the basic terminology of trees.
5. To learn basics of sorting and searching techniques

COURSE CONTENTS

Unit –I (8 hours)
[10 Hours]
Introduction: Basic Terminology, Elementary Data Organization, Structure operations, algorithm Complexity and time-Space trade-off, **Arrays& Linked list**: Array Definition, Representation and Analysis, Single and Multidimensional, Sparse Matrices, Recursive definition and processes, recursion in C, example of recursion, Tower of Hanoi, Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array

Unit – II (8 hours)
Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop,
Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack. **Queues:** Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues.

Unit – III (8 hours)
Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, path length algorithm. Huffman Algorithm. Binary Search Tree (BST), Insertion and Deletion in BST.

Unit – IV (8 hours)
Sorting: Selection sort, Bubble sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort.
Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies.

Unit – V (8 hours)

Introduction to Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal.

Text Books:

TB1. Lipschutz, "Data Structure", TMH

Reference Books:

RB1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia

RB2. R. Kruseetal, "Data Structures and Program Design in C" Pearson Education

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall fundamental concepts, terminologies, and representations of data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
CO2	Explain the operations and applications of data structures, including recursion, sorting, searching, and hashing techniques.
CO3	Implement algorithms for data structure operations like insertion, deletion, traversal, and sorting using appropriate programming techniques.
CO4	Compare and analyze the time and space complexities of various data structures and algorithms to determine their efficiency.
CO5	Assess the suitability of specific data structures and algorithms for solving real-world computational problems.
CO6	Design and develop efficient software solutions for complex problems using appropriate data structures and algorithms.

CO-PO Mapping:

	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0
CO2	3	2	0	0	0	0	0	0	0	0	0	0	3	1	0	0
CO3	2	0	3	0	3	0	0	0	0	0	0	0	3	0	2	0
CO4	0	3	0	2	0	0	0	0	0	0	0	0	0	3	0	0
CO5	0	3	0	1	0	0	0	0	0	2	0	0	0	3	0	1
CO6	0	1	3	2	0	0	0	0	0	0	0	1	2	2	3	2
AVG	1.3	1.7	1	0.8	0.5	0	0	0	0	0.3	0	0.2	1.8	1.5	0.8	0.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlate

Course code	**: MCA202.1**				
Course Name	**: DATA SCIENCE USING PYTHON**				
Semester /Year	**: IInd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Develop foundational skills in Python programming and data science tools (NumPy, Pandas, Matplotlib, Seaborn) for effective data handling and visualization.
2. Equip learners with techniques to acquire, clean, and transform structured and unstructured datasets for analysis.
3. Enable students to perform exploratory data analysis (EDA) to identify patterns, trends, and insights from real-world datasets.
4. Train learners in creating meaningful visualizations and interactive dashboards to communicate data-driven insights effectively.
5. Foster ethical awareness and best practices in data science, including bias detection, privacy considerations, and societal impact.

COURSE CONTENTS

Unit 1: Introduction to Data Science and Python (8 hours)

Fundamentals of data science and Python programming, definition and scope of data science, role of a data scientist in various industries, data science workflow including data collection, cleaning, exploration, and visualization.

Setting up Python environment with Anaconda and Jupyter Notebook, introduction to Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn, installing and configuring Anaconda and Jupyter Notebook, writing basic Python scripts for data handling, introduction to virtual environments. Hands-on exercises include setting up a Python environment and performing basic data operations using Pandas.

Unit 2: Data Acquisition and Cleaning (8 hours)

Acquiring and preparing data for analysis, types of data including structured, semi-structured, and unstructured, common data formats such as CSV, JSON, Excel, and SQL databases, handling missing values, dealing with duplicates and inconsistent data, data type conversions and formatting, reading and writing data with Pandas using functions like read_csv, read_json, and read_excel, handling missing data with dropna and fillna. Practical implementation includes cleaning a CSV dataset with missing values and inconsistent formats.

Unit 3: Data Manipulation and Transformation (8 hours)

Transforming and aggregating data, data wrangling techniques such as filtering, sorting, and grouping data, joining and merging datasets, reshaping data using pivot tables, melting, and stacking, summarizing data with group-by operations, computing aggregates like sum, mean, and count, using Pandas for filtering with loc, iloc, and query, merging datasets with merge, concat, and join, creating pivot tables and reshaping data with pivot and melt. Hands-on

exercises involve aggregating sales data by region and product category to derive meaningful insights.

Unit 4: Data Visualization (8 hours)

Visualizing data to communicate insights, principles of data visualization and its importance in data science, types of visualizations including bar charts, line plots, scatter plots, and histograms, best practices for effective visualizations, using Matplotlib for basic plotting and customization, using Seaborn for advanced visualizations and themes, using Plotly for interactive visualizations, creating basic plots with Matplotlib functions like plot, scatter, and bar, enhancing visualizations with Seaborn functions like heatmap, boxplot, and pairplot, building interactive dashboards with Plotly. Practical implementation includes visualizing trends in a time-series dataset.

Unit 5: Data Exploration and Analysis (8 hours)

Exploring and analyzing data to uncover patterns, exploratory data analysis (EDA) to understand data distributions and relationships, identifying patterns and trends, summarizing key insights from data, efficient data handling with Pandas, introduction to chunking and memory optimization for large datasets, performing EDA with Pandas functions like describe, value_counts, and groupby, visualizing relationships using scatter plots and correlation matrices, handling large datasets with read_csv and chunksize. Hands-on exercises include exploring a customer dataset to identify purchase patterns and creating visualizations to summarize findings.

Text Books

TB1. Pandas for everyone, Python Data Analysis by Daniel Y. Chen, Pearson publication

TB2. Laura Igual, Santi Seguí, "Introduction to Data Science - A Python Approach to Concepts,

Reference book:

RB1. Wes McKinney, "Python for Data Analysis", O'Reilly

RB2. Luca Massaron, John Paul Mueller, "Python for Data Science for Dummies"

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Recall the fundamental concepts of data science, including its scope, workflow, and key Python libraries (NumPy, Pandas, Matplotlib, Seaborn) used for data handling and visualization.
CO2	Explain the processes of data acquisition, cleaning, and transformation, including handling various data formats (CSV, JSON, Excel) and techniques for managing missing or inconsistent data using Pandas.
CO3	Apply data wrangling techniques such as filtering, grouping, and merging datasets using Pandas functions to prepare and transform data for analysis.

CO4	Analyze datasets through exploratory data analysis (EDA) to identify patterns, trends, and relationships using Pandas and visualization tools like Seaborn and Matplotlib.
CO5	Evaluate the effectiveness of different visualization techniques (e.g., bar charts, heatmaps, interactive plots) to communicate insights from data analysis, adhering to best practices.
CO6	Create an interactive data dashboard using Plotly to visualize trends and insights from a dataset, demonstrating proficiency in data handling and visualization.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	3	1	2	1	1	1	1	1	3	2	2	1
CO2	2	3	2	2	3	2	1	1	1	1	1	1	3	3	1	2
CO3	2	2	3	2	3	1	1	1	1	2	1	1	3	3	2	2
CO4	1	3	1	3	2	2	1	1	2	2	1	1	2	3	1	3
CO5	1	1	1	2	2	1	1	1	3	3	1	2	1	2	3	3
CO6	1	1	2	1	3	1	1	3	3	3	3	2	2	2	3	3
AVG	**1.67**	**1.83**	**1.67**	**1.83**	**2.67**	**1.33**	**1.17**	**1.33**	**1.83**	**2**	**1.33**	**1.17**	**2.33**	**2.5**	**2**	**2.33**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA202.2**				
Course Name	**: JAVA PROGRAMMING**				
Semester /Year	**: IInd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Learn how to create a GUI in Java
2. Server-side programming with the help of Servlet and JSP
3. Design and Develop Web applications
4. Database connection with the help of Java JDBC

COURSE CONTENTS

Unit I: Core Java Fundamentals (8 hours)
Features of Java, bytecode and Java Virtual Machine (JVM), Java Development Kit (JDK), data types, operators, control statements including if, else, switch, while, do-while, for, and for-each, arrays including single and multidimensional arrays, string handling using String and StringBuffer classes, command-line arguments, and wrapper classes. It also includes exception handling concepts such as try, catch, throw, throws, finally, built-in exceptions, and custom exceptions.

Unit II: Object-Oriented Programming in Java (8 hours)
Object-oriented principles including classes and objects, constructors and constructor overloading, method overloading, use of this and static keywords, final and super keywords, inheritance and method overriding, abstract classes, interfaces and interface inheritance, dynamic method dispatch, nested and inner classes including anonymous inner classes, and an understanding of the Object class in Java.

Unit III: GUI Programming with AWT and Swing (8 hours)
Layout managers such as FlowLayout, BorderLayout, GridLayout, and GridBagLayout, AWT components like labels, buttons, checkboxes, textfields, textareas, lists, scrollbars, panels, and menus, and Swing components such as JFrame, JPanel, JButton, JTextField, JComboBox, JTable, JMenu, and JScrollPane. The unit also includes event handling using the event delegation model, various event listener interfaces, applet basics including lifecycle, HTML applet tags, and the use of graphics, fonts, and colors in GUI. Basics of JavaBeans are also covered.

Unit IV: File I/O, Collections, and Multithreading (8 hours)
Input/output operations using byte and character streams, File class, FileReader, FileWriter, BufferedReader, InputStreamReader, and OutputStreamWriter. It also covers the collection framework including List, ArrayList, Vector, Enumeration, and Properties, along with an introduction to the java.util package. Additionally, it includes multithreading concepts using the Thread class and Runnable interface, thread lifecycle, thread priority, and synchronization.

Unit V: Web Programming with JDBC, Servlets, and JSP (8 hours)

Java Database Connectivity (JDBC) with topics such as database connection, Statement and PreparedStatement, executing queries, transactions, and stored procedures. It also covers servlets including servlet lifecycle, javax.servlet package, handling GET and POST requests, session tracking using cookies and HTTP session. JavaServer Pages (JSP) are introduced with scripting elements, implicit objects, standard actions, and directives. The unit concludes with the basics of web servers like Apache Tomcat, XAMPP, and WAMP.

Text Books:

TB1. Herbert Schildt (2006), "The Complete Reference Java 2 (Updated to Cover J2SE 1.4)", Tata McGraw-Hill .

TB2. Cay S. Horstmann Gary Cornell, "Core Java 2 Volume-I Fundamentals", PEARSON Education.

Reference Books:

RB1. Michael Morgan, "Java 2 for Professionals Developers", SAMS, Techmedia.

RB2. Bruce Echel, "Thinking in Java, The Definitive Introduction to Object-Oriented Programming in the Language of World-Wide-Web", PEARSON Education.

RB3. Philip Heller and Simon Roberts, "Java 2 Developer's Hand Book", BPB Publication.

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall core Java concepts and OOP principles
CO2	Explain the working of exception handling, multithreading, and GUI components (AWT/Swing) with examples.
CO3	Develop programs using arrays, strings, file I/O, and collections to solve computational problems.
CO4	Compare and contrast Java technologies (JDBC vs. Servlets/JSP) and evaluate their suitability for real-world applications.
CO5	Assess the efficiency, scalability, and ethical implications of Java-based solutions (e.g., thread safety, data security).
CO6	Design and implement a full-stack Java application (GUI + Database + Web) integrating JDBC/Servlets/JSP.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	-	-	-	-	-	-	-	-	-	3	1	-	-
CO2	2	3	2	1	-	-	-	-	-	-	-	-	2	3	1	-
CO3	2	2	3	1	2	-	-	-	-	-	-	-	3	2	1	-
CO4	1	3	2	3	1	1	-	-	-	1	-	-	2	3	1	2
CO5	-	2	1	2	-	3	1	-	-	2	-	1	1	2	3	2
CO6	-	1	3	2	3	2	2	3	2	2	3	2	3	2	3	3
AVG	1.6	2	2	1.8	1.5	1.5	1	1.5	1	1.25	1.5	1	2.3	2.2	1.8	1.75

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA203**				
Course Name	**: DBMS**				
Semester /Year	**: IInd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Develop a broad understanding of database concepts and database managementsystem software, data models, schemas and instances, data constraints, relational algebra and calculus.
2. Acquire Knowledge to model an application's data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.
3. Be able to write SQL commands to create and manipulate database objects.
4. Be able to discuss importance of normalization and improve the database design by applying various normal forms.
5. Get in depth knowledge of concurrency control mechanisms, transactionmanagement techniques and database security.

COURSE CONTENTS

UNIT I – Basic concepts (10 hours)
Database, Characteristics of the Database, Approach & advantages of using DBMS. Data Models, Schemas& Instances. Database abstraction & Data Independence. Overall structure of Database, Data Dictionary, Database Users, Role of DBA. Data Modeling using the Entity-Relationship Model -Entity types, Entity Sets, Attributes and Keys, Relationship & its Types, Enhanced ER Model- Specialization, Generalization, Aggregation.

UNIT II –Relational Model, Languages & Systems (10 hours)
Relational Data Model Concepts and Constraints. Relational Algebra, select, project & join operations. Overview of keys (primary, composite, foreign, alternate, candidate), relational Calculus. **SQL:** DDL statements, DML statements, Views, sequence, synonyms, sub queries, joins, transaction commands, specifying constraints, Indexes in SQL.

UNIT III – Relational Data Base Design (10 hours)
Function Dependencies & Normalization, Normal forms (INF, 2NF, 3NF & BCNF). Lossless join & Dependency preserving, decomposition, multivalued dependencies, join dependencies (4NF & 5NF).

UNIT IV – Transactions, Concurrency Control, Recovery Techniques (5 hours)
Basic concept, Transaction state and execution, ACID properties, Transaction Log, Schedules, Serializable schedule, serializability, conflict serializability, view serializability

UNIT V – Concurrency Control and Recovery Techniques (5 hours)
Concurrency control, Problems of concurrency control, **concurrency control protocols:** Lock-Based Protocols, Two Phase Locking Protocol, Timestamp-Based Protocols, Validation-Based Protocols, **Database Recovery:** Types of database failure, Types of database recovery, Recovery Techniques, log based recovery, checkpoints.

Text Books:

TB1. Elmsari and Navathe, "Fundamentals of Database Systems", Pearson Education

TB2. Korth, Silberschatz, "Fundamentals of Database System Concepts", TMH

TB3. Ivan Bayross, "SQL, PL/SQL the Programming language of Oracle", BPB publications

TB4. Desai B., "An Introduction to Database Concepts", Galgotia Publications, New Delhi.

Reference Books:

RB1. Ullman J. D., "Principals of Database Systems", Galgotia Publications

RB2. C.J.Date, A. Kannan, S. Swamynathan "An Introduction to Database Systems", Pearson Education

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Recall fundamental concepts of databases, data models, ER diagrams, and SQL syntax.
CO2	Explain relational algebra operations, normalization forms, and ACID properties with examples.
CO3	Design normalized relational schemas (up to BCNF) and implement them using SQL queries (DDL/DML).
CO4	Evaluate transaction schedules for serializability and concurrency issues (e.g., deadlocks).
CO5	Assess recovery techniques (log-based, checkpoints) and concurrency protocols (2PL, timestamp) for given scenarios.
CO6	Develop a database application with ER modeling, SQL queries, and transaction handling for real-world problems.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	-	1	-	-	-	-	-	-	-	3	1	-	-
CO2	3	2	1	1	-	-	-	-	-	-	-	-	2	3	-	1
CO3	2	2	3	1	2	-	-	1	-	-	1	-	3	2	3	1
CO4	1	3	2	3	2	-	-	-	-	-	-	-	1	3	1	2
CO5	1	2	1	2	3	1	-	-	-	-	-	-	2	2	-	3
CO6	2	3	3	2	3	2	1	3	2	1	3	2	3	3	3	3
AVG	2	2.2	1.8	1.5	1.8	0.5	0.2	0.7	0.3	0.2	0.7	0.3	2.3	2.3	1.3	1.7

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**:**	**MCA204**			
Course Name	**:**	**SOFTWARE ENGINEERING**			
Semester /Year	**:**	**IInd Semester**			
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To understand the nature of software development and software life cycle.
2. Explain methods for capturing specifying, visualizing analyzing software requirement.
3. To know basic of testing and understanding concept of software quality assurance.
4. To understand the concept of software cost model.
5. To develop good quality software and able to maintain quality of software
6. To develop knowledge of tools available for software development.

COURSE CONTENTS

Unit I Introduction to Software Engineering (6 hours)

Introduction Evolution of Software Engineering, Software Engineering: A layered technology, process frame work and software engineering paradigms, Software process technology, Software Requirements Analysis, Analysis Principles, Modeling the system architecture, Software prototyping and specification.

Unit II Software Design (8 hours)

Software Design Process, Concepts, Principles, Architectural Design, Data Design, Mapping requirements into Software Architecture, Effective modular design, Procedural Design, Interface design.

Unit III Software Cost Model (8 hours)

Software Quality Planning and Project Management Evaluation of individual projects: Technical assessment, cost-benefit analysis (Evaluation Techniques), and Risk evaluation, Concept of Software Project Management and its importance, software cost estimation techniques, different types of project metrics, Models for cost estimation (COCOMO, Putnam's, function point), Introduction to project scheduling, project schedules, project and activities, scheduling activities, Schedule development methods (Critical Path Method, Critical Chain Scheduling, PERT).

Unit IV Software Quality Assurance (10 hours)

Software Quality Assurance Introduction, Quality Planning, Quality Assurance, Quality Control, Tools and Techniques of Quality Control, Pareto analysis, Six Sigma, Cost of Quality, software quality metrics (McCal's Quality Model, Boehm's Quality Model, Dromey's Quality Model), Capability maturity models.

Unit V Project Management (8 hours)

Project Management – Definitions; Factors Influencing Project Management – Project Manager, Project Management Activities, Stakeholders; Project Communication; Project Development Phases; Project Charter; Statement of Work (SoW); Project Management Associations.

Text books:

TB1. Software Engineering, Rogers S. Pressman, MH
TB2. Fundamentals of Software Engineering, Ghezzi, PHI
TB3. Software Engineering, Pankaj Jalote, PHI

Reference Books:

RB1. Jalote, Pankaj, "Software Engineering", New Delhi: Narosa 2002
RB2. Schaum's Series, "Software Engineering" TMH

Course Outcomes (COs):

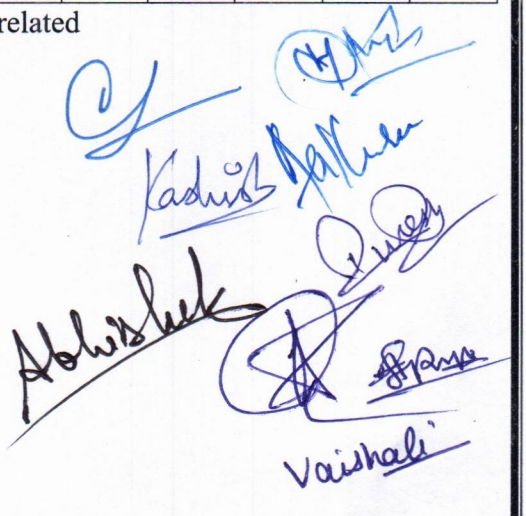
After completion of this course, the learners will be able to: -

CO #	Detailed Statement of the CO
CO1	Recall key concepts of software engineering, including SDLC models, requirements analysis, and quality assurance frameworks.
CO2	Explain software design principles, cost estimation models (e.g., COCOMO), and quality metrics (e.g., McCall's model).
CO3	Demonstrate the use of tools (e.g., UML, CPM/PERT) to design software architectures and project schedules.
CO4	Evaluate software requirements, design trade-offs, and risk factors to propose optimized solutions.
CO5	Assess software quality using maturity models (CMMI) and ethical practices in project management.
CO6	Develop a software project plan with specifications, prototypes, and a QA strategy for real-world problems.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1									2	1		
CO2	2	3	1	1									3	2	1	
CO3	1	2	3	1	3								3	2	3	1
CO4		3	3	2			1			1		1	2	3	2	1
CO5				1		3	1			2		1			2	3
CO6			2				1	3	2	1	3	2	1		3	2
AVG	**1.5**	**2.25**	**2**	**1**	**1.5**	**1.5**	**0.5**	**1.5**	**1**	**0.75**	**1.5**	**1**	**2.2**	**1.75**	**2**	**1.5**

3 – Highest Correlated, 2 – Medium Correlated, 1 – Low Correlated



Course code	**: MCA205**				
Course Name	**: ALGORITHMS ANALYSIS & DESIGN**				
Semester /Year	**: IInd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Understand the important concepts of algorithms design and their analysis.
2. Analyze the efficiency of alternative algorithmic solutions to the problem.
3. Understand different algorithm paradigms like Divide and Conquer, Greedy, Dynamic, Backtracking and Branch and Bound.
4. Identify the appropriate data structures, algorithm design techniques and assess their impact on the performance of programs

COUSE CONTENTS

UNIT-I (9 hours)

Algorithms, Analysis of Algorithms, Design of Algorithms, and Complexity of Algorithms Growth of Functions: Asymptotic notations, Mathematical Analysis of Non-Recursive and Recursive Algorithms. Recurrences and Solution of Recurrence Equations- The Substitution method, The Recurrence–Tree Method, The Master Method.

UNIT-II (9 hours)

Sorting in polynomial Time: Insertion sort, Merge sort, Heap sort, and Quick sort, Sorting in Linear Time: Counting sort, Radix Sort, Bucket Sort. Red-Black Trees, Augmenting Data Structure.

UNIT-III (9 hours)

Greedy Technique: Fractional Knapsack Problem, Activity Selection Problem, Dynamic Programming: 0/1 Knapsack Problem, Matrix-Chain Multiplication. Backtracking: Hamiltonian Circuit Problem; Branch-and-Bound: Assignment Problem, Traveling Salesperson Problem;

UNIT-IV (9 hours)

Graph: Introduction, Representation of Graph, BFS, DFS, Minimum Spanning Tree: Prims Algorithm, Kruskal Algorithm, Single Source.
Shortest Paths: Bellman-Ford Algorithm, Dijkstra Algorithm, All Pair Shortest Paths: Floyd-Warshall Algorithm,
Flow Network: Maximum flow -min cut theorem.

UNIT-V (6 hours)

NP-Completeness: P, NP, NP-Hard & NP-Complete Class, Reducibility & NP-Complete Problems. Approximation Algorithms: The Vertex Cover Problem, The Set Covering Problem

Text Books:

TB1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms", PHI

TB2. S. Dasgupta, C. Papadimitriou and U.Vazirani, "Algorithms", McGraw Hill Higher Education

Kashish Arora
Vaishali
Abhishek

TB3. J. Kleinberg and E. Tardos, "Algorithm Design", Pearson Education

Reference Books:

RB1. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, Prentice-Hall of India,

RB2. Ellis Horowitz, Sartaj Sahni, S. Rajasekharan: Fundamentals of Computer Algorithms, Universities press,

RB3. Anany Levitin: Introduction to The Design & Analysis of Algorithms, 2nd Edition, Pearson Education.

RB4. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning.

RB5. Baase: Computer Algorithms: Introduction to Design and Analysis, Addis

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Recall fundamental algorithmic concepts, asymptotic notations, and recurrence relations.
CO2	Explain the working principles of sorting, graph algorithms, and NP-completeness classes.
CO3	Implement algorithms such as divide-and-conquer, greedy methods, and dynamic programming to solve computational problems.
CO4	Compare the efficiency of algorithms using mathematical analysis and empirical evaluation.
CO5	Assess the correctness and optimality of algorithmic solutions for real-world problems.
CO6	Design optimized algorithms for complex problems (e.g., shortest path, NP-hard approximations) and justify their efficiency.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	1	-	-	-	-	-	-	-	3	1	-	1
CO2	3	3	2	2	1	-	-	-	1	-	-	-	3	3	1	2
CO3	3	3	3	2	2	-	1	1	1	-	1	-	3	3	2	2
CO4	2	3	2	3	3	-	1	-	2	-	-	-	2	3	2	2
CO5	2	3	3	3	3	1	1	-	2	1	-	1	2	3	3	2
CO6	2	3	3	3	3	1	2	1	2	2	2	2	2	3	3	3
AVG	2.5	2.7	2.3	2.3	2.2	0.3	0.8	0.3	1.3	0.5	0.5	0.5	2.5	2.7	1.8	2

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA206**				
Course Name	**: COMPUTER NETWORK**				
Semester	**: IInd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
4. Independently understand basic computer network technology.
5. Identify the different types of network topologies and protocols.
6. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.

COURSE CONTENTS

Unit 1 (8 hours)

Introductory Concepts: Goals and Applications of Networks, Need of Network, Elements of Network, Network structure and architecture, OSI &TCP/IP reference model, Transmission modes, networks topology, Guided and un Guided Media, High Speed networks: FDDI, Integrated services digital networks.

Unit 2 (12 hours)

Data Link Layer: Framing, error control (Single parity bit checking, 2D parity bit checking, checksum, CRC, Hamming Error Correction code), Elementary data link protocols: - sliding windows protocols (Stop & Wait ARQ, GO Back to N & Select repeat ARQ), High Level Data Link Control

Medium access sub layer: Channel allocations (static and dynamic channel allocation), ALOHA Protocols- Pure ALOHA, slotted ALOHA, Carrier Sense Multiple Access Protocols, CSMA/CD, CSMA/CA.

Unit 3 (10 hours)

Network Layer: Point-to Point networks, routing algorithms, Adaptive and Non adaptive Routing algorithm, (Distance vector and Link state routing algorithm), congestion control strategies & algorithms (Token bucket and leaky bucket algorithm), internetworking, IPV4 & IP Datagram, IP addresses, Class of IP, Subnetting, masking, IPv6.

Unit 4 (8 hours)

Transport Layer: Design issues, connection management, Three-way handshaking, TCP window Management, User Datagram Protocol, Transmission Control Protocol, Port number and socket address.

Unit 5 (6 hours)

Application Layer: Network Security, symmetric and Asymmetric encryption algorithm, DES, RSA algorithms, Domain Name System, Simple Network Management Protocol, Electronic mail, SMTP, POP, File Transfer Protocol, Hyper Text Transfer Protocol.

Text Books:

TB2. W. Stallings, "Data and Computer Communication", Macmillan Press

TB3. Forouzan, "Data Communication and Networking", TMH

Reference Books:

RB1. Tanenbaum, "Data Communication & Computer Network"

RB2. Jefferey, Piyasat , "Networking Essentials" , PHI, 2007

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall the fundamental concepts, components, and reference models (OSI, TCP/IP) of computer networks, including transmission modes and network topologies.
CO2	Explain the functionality of data link layer protocols, error control mechanisms, and medium access control techniques like ALOHA and CSMA.
CO3	Apply routing algorithms and congestion control techniques to design efficient network configurations for given scenarios.
CO4	Analyze network performance by comparing IPv4 and IPv6 addressing schemes, subnetting, and error detection methods.
CO5	Evaluate the effectiveness of transport layer protocols (TCP, UDP) and application layer protocols (HTTP, SMTP) in meeting specific network requirements.
CO6	Design a secure network system incorporating encryption algorithms (DES, RSA) and appropriate protocols to address real-world communication needs.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3						1						2			
CO2	3	2					1		1				2	1		
CO3	3	2	3		2		1						3	2	2	
CO4	3	3		2	2		1		1				2	3		1
CO5	3	3	2	2	2	1	1		1	1			3	3	2	2
CO6	3	3	3	2	2	3	1	1	1	2	1	2	3	3	3	3
AVG	3	2.17	2	1.33	1.67	0.67	1	0.17	0.67	0.5	0.17	0.33	2.5	2	1.17	1

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**:**	**MCAP21**			
Course Name	**:**	**DATA STRUCTURE LAB**			
Semester /Year	**:**	**IInd Semester**			
		L	**T**	**P**	**C**
		0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To learn how to design the algorithms to solve the programming problems.
2. To learn how to discriminate the usage of various structures in approaching the problem solution.
3. To learn how to use effective and efficient data structures in solving various Computer Engineering domain problems
4. To learn how to analyse the problems to apply suitable algorithm and data structure.
5. To learn how to use appropriate algorithmic strategy for better efficiency

COURSE CONTENTS

- Program based on arrays
- Program based on strings
- Program based on Link List & types of Link List
- Program based on stack
- Program based on Queues
- Program based on Trees
- Program based on different types of sortings
- Program based on Graphs

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Analyze & understand the difference between linear and non linear DS & implement array & link list and its types
CO2	Understand and implement stack and queues using array and link list
CO3	Understand and implement BST, addition and deletion of nodes, Huffman algorithm etc.
CO4	To implement different sorting techniques like selection Bubble, insertion, merge quick sort etc.
CO5	To understand and implement linear and binary search
CO6	To create and implement of graphs like directed and undirected graphs etc

CO-PO Mapping:

	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2				1						2	1		1
CO2	3	1	2			1	1						2	1		1
CO3	3	2	3	1	1	2	2						2	1	1	1
CO4	1	1	3	1		2	2						2	1		1
CO5	1	1	3	1		1	2						2	2	1	1
CO6	1	1	3			1	2						2	1		1
AVG	1.8	1.3	2.6	0.5	0.1	1.1	1.5						2	1	0.3	1

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlate

Course code	**: MCAP22.1**				
Course Name	**: DATA SCIENCE LAB**				
Semester /Year	**: IInd Semester**				
		L	**T**	**P**	**C**
		0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Using the data science using Python
2. Understanding the use of NumPy in data science
3. To understand the dataframe, reading csv, tsv file and working on it.
4. Handling the missing value and data cleansing
5. Understanding the concept of visualize the numbers in graphs using the Python libraries Metplotlib and Seaborn

COURSE CONTENTS

- Programs based using NumPy.
- Programs based on using series and dataframe using Pandas.
- Reading CSV files in Panadas
- Handling missing values in Panadas
- Using concat(), join(), append() and groupby in Panadas
- Plotting line plot, histogram, bar chart, pie chart, scatter plot, subplot etc. using Matplotlib and saving the figure.
- Plotting line plot, histogram, bar plot, scatter plot, heatmap, etc. using Seaborn

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Design algorithms involving more complex data structures, and can implement it.
CO2	Understanding the performance of multiple methods and models, recognize the connections between how the data were collected and the scope of conclusions from the resulting analysis
CO3	Apply models and use different measures of model to assess outputs.
CO4	Analyse the data model applied using the data science tools.
CO5	Evaluate the data from disparate sources, cleaning the data and transform data from one format to another
CO6	Developing the program using various tools of data science.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3				2		1	1					2			1
CO2	1	3	2	1	2		1	1			1		3	2	1	2
CO3	1	3	2	1	2		1	1			1		3	2		2
CO4	1	3	2	1	2		1	1			1		3	2		2
CO5	1	3	3	2	1	1	1	1	1	1	2	1	3		2	2
CO6	3	3	3	2	2	1	1	1	1	1	2	1	3	2	2	2
AVG	**1.7**	**3.0**	**2.4**	**1.4**	**1.8**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.4**	**1.0**	**2.8**	**2.0**	**1.7**	**1.8**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code : **MCAP22.2**				
Course Name : **JAVA LAB**				
Semester /Year : **IInd Semester**				
	L	**T**	**P**	**C**
	0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Develop proficiency in writing, compiling, and debugging Java programs using core concepts like data types, control structures, and arrays.
2. Design and implement object-oriented solutions using classes, inheritance, polymorphism, and interfaces to solve real-world problems.
3. Create interactive GUI applications using AWT/Swing components, event handling, and layout managers.
4. Apply file I/O operations, collections, and multithreading concepts to build efficient and scalable Java applications.
5. Develop and integrate database-driven applications using JDBC, Servlets, or JSP to demonstrate end-to-end software solutions.

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall syntax, data types, control structures, and core Java concepts to write basic programs.
CO2	Explain object-oriented principles (inheritance, polymorphism) and apply them to design Java classes and methods.
CO3	Develop GUI applications using AWT/Swing components and event handling to solve user-interface problems.
CO4	Debug and optimize Java programs by identifying errors in multithreading, file I/O, or collections.
CO5	Test and validate JDBC/Servlet-based applications for functionality, efficiency, and security.
CO6	Design and implement a full-stack Java project (e.g., inventory system) integrating OOP, GUI, and database connectivity.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	1	1	1	1	1	1	1	3	1	1	1
CO2	3	2	2	1	1	1	1	1	1	1	1	1	3	2	1	1
CO3	2	2	3	1	3	1	1	1	1	1	1	1	3	2	1	1
CO4	2	3	2	3	2	1	1	1	1	1	1	1	2	3	1	2
CO5	1	2	2	2	2	3	1	1	1	1	1	1	2	2	2	3
CO6	2	3	3	2	3	2	2	3	2	2	3	2	3	3	3	2
AVG	**2.2**	**2.2**	**2.2**	**1.7**	**2.2**	**1.5**	**1.2**	**1.3**	**1.2**	**1.2**	**1.3**	**1.2**	**2.7**	**2.2**	**1.5**	**1.7**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAP23**				
Course Name	**: SQL LAB**				
Semester /Year	**: IInd Semester**				
		L	**T**	**P**	**C**
		0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Working expertise of DDL and DML commands with their application on solving real time problems.
2. Ability to apply filters using where clause, and nested queries, integrity constraints at table level and column level and to use built-in functions including numeric, character and date functions.
3. Adequate knowledge to fetch data from multiple tables using different types of JOIN operations.

COURSE CONTENTS

- Queries used for creating and managing tables i.e. Data Definition Language (DDL) e.g., create table, alter table, drop, rename table etc.
- Queries used for manipulating data i.e. Data Manipulation Language (DML) e.g., inserting rows into a table, update rows in a table, delete rows from a table,
- Writing and executing basic SQL queries
- Including constraints while creating tables.
- Queries based on restricting and sorting data
- Queries based on single row functions used in character, number and date.
- Queries based on displaying data from multiple tables.
- Aggregating data using group functions
- Queries based on subqueries.
- Describing and creating view, retrieving data through a view, alter definitions of a view, insert, delete and update data through a view, drop a view

Course outcomes (COs):

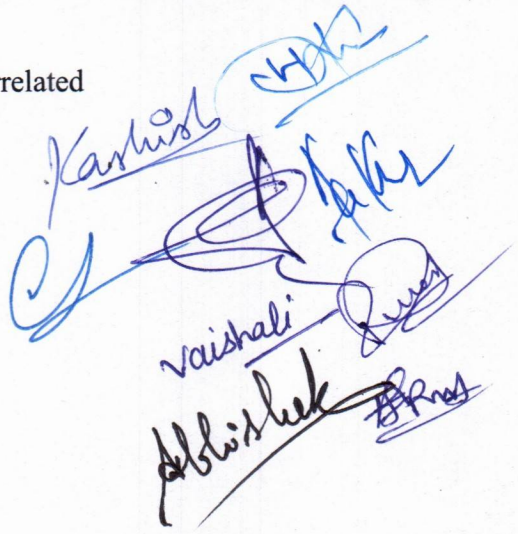
Upon successful completion of the course a student will be able to

CO #	Detailed Statement of the CO
CO1	Defining the databases, tables and query a database using SQL DML/DDL commands.
CO2	Understanding the sub languages used in SQL to work with database
CO3	Demonstrate the use of constraints, relational algebra operations and Grouping.
CO4	Analyse the knowledge of SQL queries in while developing database applications.
CO5	Evaluate the concept of Views, Rollback, Commit, Grant and Revoke Permission.
CO6	Design solutions for real world problems/case studies by creating efficient database schema.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	1	3	2	3	2	1						3			1
CO2	2	1	2	1	2	1		1					3	1		1
CO3	3	1		1	1	1	2						3	1	2	2
CO4	1	1	3	2	1		3	1			2	1	3	2	2	1
CO5	1	1	2			1	2	2			3		1	1	3	1
CO6		3	3	1	1	1	1						3	2	2	2
AVG	**1.6**	**1.3**	**2.6**	**1.4**	**1.6**	**1.2**	**1.8**	**1.3**			**2.5**	**1.0**	**2.7**	**1.4**	**2.3**	**1.3**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated



Course code	**: MCASM24**			
Course Name	**: SEMINAR AND PRESENTATION**			
Semester /Year	**: IInd Semester**			
	L	**T**	**P**	**C**
	0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Identify and compare technical and practical issues related to the area of course specialization.
2. Outline annotated bibliography of research demonstrating scholarly skills.
3. Prepare a well-organized report employing elements of technical writing and critical thinking.
4. Demonstrate the ability to describe, interpret and analyze technical issues and develop competence in presenting.

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO	Detailed Statement of the CO
CO1	Defining the aim of the seminar topic.
CO2	Understanding the seminar topic and requirements of technical resources.
CO3	Apply the critical thinking on the topic of the seminar
CO4	Illustrate the work done in the topic with presentation.
CO5	Work is evaluated by a panel to boost the confidence to the student.
CO6	Create technical documents.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3			2			2					2	1	1	1	2
CO2			3	2							2		1		1	2
CO3					2				3				1		2	2
CO4									3				1		3	
CO5	2						2		2					1	2	2
CO6	2	2	1	2	2		2		2			1	1	1	2	2
AVG	**2.3**	**2.0**	**2.0**	**2.0**	**2.0**		**2.0**		**2.5**		**2.0**	**1.5**	**1.0**	**1.0**	**1.8**	**2.0**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

3rd Semester

Course code	**: MCA301**				
Course Name	**: C# DOT NET**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To study the features of .NET Technologies and to understand the framework and environment.
2. To learn C# programming fundamentals for console application development.
3. To learn use of C# libraries and exception handling techniques.
4. To learn ADO. NET and advance features of C#.
5. To learn .NET assemblies and attributes

COURSE CONTENTS

UNIT I (9 hours)

Introduction to .NET Framework And **C#**.NET framework, MSIL, CLR and its components, CLS, CTS, Just in time (JIT) Compiler,Base class library, Namespaces, Assemblies, DLL - Hell Problem, Garbage Collection.

Introduction to C#, Visual studio console app, Boxing and Unboxing, loops, Array, Enumerations, structures.

UNIT II C# Object oriented programming (9 hours)

OOPs, Classes and objects, ,Encapsulation, Inheritance, Polymorphism, Abstract class, Sealed class, Interface ,Constructor and Destructors, Method Overloading ,Method overriding, Operator Overloading, Modifiers, Indexers , Collections Namespaces, Delegates, Event handling , Exception Handling.

UNIT III IO and Threads (6 hours)

Multithreading, Thread pooling, App domains, Concurrency and synchronization- Locks, Monitors, Mutexes, System.IO, Streams, TextWriter, TextReader, BinaryWirter, BinaryReader.

UNIT IV ADO.Net, C# windows forms for data control (8 hours)

Grid, Datasource and databinding controls, Connected and disconnected scenarios, Dataset, connections, Adapters, commands, datareaders. Windows Forms and Controls in details: Windows form, Windows Forms Properties and Events, Windows Form Controls etc.

UNIT V ASP.NET (8 hours)

Introduction to ASP.NET, Architecture, Working with Web and HTML Controls, Server Controls, Overview of ASP.NET Validation Controls, Data-base connectivity using ASP.net. Introduction of XML, Using XML with ASP.net

Master Pages, Displaying Data with the Grid View Control, State management.

Text Books:

TB1. C#.Net Developers Guide- Greg Hack, Jason Werry, SaurabhNandu. (SyngRess)
TB2. Wrox Press Professional C# – Simon Robinson, Jay Glynn

Reference Books:

RB1. Addison Wesley –C# Developers Guide to ASP.Net
RB2. Wiley," Beginning Visual C# 2008", Wrox
RB3. Claudia M. Baca, Patti, PMP: Project Management Professional Workbook, Sybex, Workbook

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Knowledge and remember.NET Framework, its runtime environment and application development IDE of Visual Studio.
CO2	Understand the concept of object oriented for making programs.
CO3	Implement C# language constructs in the form of stand-alone console and window form applications.
CO4	Analyze and Understand database concepts in ADO.NET and apply the knowledge to implement distributed data-driven applications.
CO5	Design, document, debug ASP.NET web forms with server and validation controls and implement ASP.NET web services.
CO6	Create the programs based on console, windows and web application.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	1	1	1	1	1	1	1	3	1	1	1
CO2	2	3	2	1	1	1	1	1	1	1	1	1	2	3	1	1
CO3	3	2	3	1	2	1	1	1	1	1	1	1	3	2	2	1
CO4	3	3	3	3	2	2	1	2	1	2	2	1	3	3	2	2
CO5	3	2	3	2	3	2	1	2	2	3	2	2	3	2	3	3
CO6	3	2	3	2	3	2	1	3	2	2	3	2	3	2	3	2
AVG	2.8	2.2	2.5	1.7	2.2	1.5	1	1.7	1.3	1.7	1.7	1.3	2.8	2.2	2	1.7

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA302**				
Course Name	**: NETWORK SECURITY & CRYPTOGRAPHY**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Understand the basic Knowledge of Network Security Concepts &the Challenges and Scope of Information Security
2. Understand the basic Concept of Block Cipher &the Importance of Cryptographic Algorithms and their Uses.
3. Learn and Understand Encryption Techniques & Access Control Mechanism Used for User Authentication and Authorization.
4. Understand the concept of authentication of message in network by using different techniques.
5. Aware and Learn the Usages of Sockets Layer (SSL), Secure Internet Protocol (IP) and HTTPS

COURSE CONTENTS

UNIT I Introduction To security (8 hours)
Basic aspects of Network security, Attacks active & passive attacks, Types of attack, VIRUS and its categories, Conventional Encryption, Conventional Encryption Model, Substitution & transposition techniques, Bit level encryption and operations.

UNIT II Introduction Block Cipher Differential & Linear Cryptanalysis (5 hours)
Block Cipher Design Principles, Block Cipher Modes of Operations, steganography, Differential & Linear Cryptanalysis

UNIT III Conventional Encryption Algorithms (9 hours)
DES, DES Algorithm, DES strength ,2DES, 3 DES, Men in Middle attack, DES standard, AES, Blowfish, International Data Encryption Algorithm IDEA, RC-5, CAST-128, RSA, Key Distribution, Deffie Hellmen Key Exchange Algorithm, Random Number Generation.

UNIT IV Message Authentication & Hash Functions (10 hours)
Authentication Requirements, Authentication Functions, Message Authentication Codes MAC, HMAC, Hash Function, Birthday Attacks, Message Digest Algorithm: MD5 & SHA, Digital Signature, Digital Signature Standard (DDS), Proof of Digital Signature Algorithm, Digital certificate. X.509, X.25, Secure Electronic Transaction, , Authentication Protocol, Authentication Applications: Kerberos.

UNIT V Email, Internet Security (7 hours)
Secure Socket Layer, Directory Authentication Service, Electronic Mail Security, PEM, Pretty Good Privacy (PGP), IP security, S/MIME Security: Architecture, Authentication Header, Encapsulating Security Payloads, cyberlaws

Text Books:

TB1. Atul Kahate, "Cryptography and Network Security" TMH

Reference Books:

RB1. William Stallings, "Cryptography and Network Security: Principles and Practice",Prentice hall, New Jersey

RB2. Johannes A. Buchmann, "Introduction to Cryptography" Springer-Verlag

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall fundamental concepts of encryption, attacks (active/passive), and cryptographic algorithms (DES, AES, RSA).
CO2	Explain block cipher modes, hash functions, digital signatures, and their role in secure communication.
CO3	Implement symmetric (DES/AES) and asymmetric (RSA) encryption techniques using modern tools.
CO4	Evaluate vulnerabilities in cryptographic systems (e.g., MITM, Birthday attacks) and compare algorithm strengths.
CO5	Assess authentication protocols (Kerberos, SSL/TLS) and ethical implications of cyber laws.
CO6	Design a secure communication model using hybrid cryptography (e.g., PGP, IPsec) for real-world scenarios.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1
CO2	3	2	1	1	1	1	1	1	1	2	1	1	2	2	1	1
CO3	2	2	3	1	3	1	1	1	1	1	1	1	3	2	2	1
CO4	2	3	2	3	2	1	1	1	1	2	1	1	2	3	1	2
CO5	1	2	1	2	1	3	1	1	2	3	1	2	1	2	1	3
CO6	1	2	3	2	2	2	3	3	2	3	2	3	2	2	3	2
AVG	2	2	1.8	1.7	1.7	1.5	1.3	1.3	1.3	2	1.2	1.5	2.2	2	1.5	1.7

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated



Course code	**: MCA303**				
Course Name	**: ARTIFICIAL INTELLIGENCE**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To create application and understanding of both the achievements of AI and theory underlying those achievements.
2. To introduce concept search in AI with help of Heuristic search technique.
3. To introduce the concept of Natural Language Processing.
4. To introduce the concept of Knowledge Representation.

COURSE CONTENT

UNIT I General Issues and overview of AI (8 hours)

The AI problems: what is an AI technique; Characteristics of AI applications Problem Solving, Search and Control Strategies General Problem solving; Production systems; Control strategies; forward and backward chaining Exhaustive searches: Depth first Breadth first search.

UNIT II Heuristic Search Techniques (8 hours)

Hill climbing; Branch and Bound technique; Best first search and A* algorithm; AND/OR Graphs; Problem reduction and AO* algorithm; Constraint Satisfaction problems Game Playing Min Max Search procedure; Alpha-Beta cutoff; Additional Refinements.

UNIT III Knowledge Representation (8 hours)

First Order Predicate Calculus; Skolemnisation; Resolution Principle and Unification; Inference Mechanisms Horn's Clauses; Semantic Networks; Frame Systems and Value Inheritance; Scripts; Conceptual Dependency AI Programming Languages Introduction to LISP, Syntax and Numeric Function; List manipulation functions; Iteration and Recursion.

UNIT IV Natural Language Processing and Parsing Techniques (8 hours)

Context – Free Grammar; Augmented Transition Nets (ATN); Semantic Analysis, Case and Logic Grammars; Planning Overview – An Example Domain: The Blocks Word; Component of Planning Systems; Goal Stack Planning (linear planning).

UNIT V Expert Systems (8 hours)

Introduction to Expert Systems, Architecture of Expert Systems; Expert System Shells; Knowledge Acquisition; Case Studies; MYCIN.

Text Books:

TB1. Elaine Rich and Kevin Knight: Artificial Intelligence – Tata McGraw Hill.

TB2. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems – Prentice Hall of India.

Reference Books:

RB1. Elaine Rich and Kevin Knight: Artificial Intelligence – Tata McGraw Hill.

RB2. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems – Prentice Hall of India.
RB3. Nils J. Nilsson: Principles of Artificial Intelligence – Narosa Publication house.
RB4. Artificial Intelligence : A Modern Approach, Stuart Rusell, Peter Norving, Pearson Education 2nd Edition.
RB5. Artificial Intelligence, Winston, Patrick, Henry, Pearson Education.
RB6. Artificial Intelligence by Gopal Krishna, Janakiraman, Wesley, California, 2002

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall key AI concepts, including problem-solving strategies, search techniques, and knowledge representation methods, as
CO2	Explain the principles of heuristic search algorithms, natural language processing techniques, and expert system architectures'
CO3	Apply AI techniques such as A* algorithm, resolution principle, or goal stack planning to solve specific computational problems
CO4	Analyze the effectiveness of AI methods like Min-Max search, semantic networks, or MYCIN's rule-based system in addressing complex problems
CO5	Evaluate the suitability of AI approaches, such as Alpha-Beta pruning or ATNs, for specific applications like game playing or NLP
CO6	Design and implement a simple AI system, such as a LISP-based knowledge representation or a planning system

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	1	0	2	0	0	0	0	0	2	1	0	1
CO2	3	2	1	2	1	0	2	0	1	1	0	1	2	2	1	2
CO3	3	3	3	2	2	0	2	1	1	1	1	1	3	3	2	2
CO4	3	3	2	3	2	1	2	1	1	2	1	2	3	3	2	3
CO5	3	3	2	3	2	1	2	1	1	2	1	2	3	3	2	3
CO6	3	3	3	3	3	1	3	2	2	2	2	3	3	3	3	3
AVG	**3**	**2.5**	**2**	**2.3**	**1.8**	**0.5**	**2.2**	**0.8**	**1**	**1.3**	**0.8**	**1.5**	**2.7**	**2.5**	**1.7**	**2.3**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA304.1**				
Course Name	**: MACHINE LEARNING**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Recall core ML concepts like supervised/unsupervised learning and key evaluation metrics.
2. Implement ML models using Python and analyze their performance on real datasets.
3. Compare different algorithms by evaluating bias-variance tradeoffs and model complexity.
4. Design an end-to-end ML solution for a real-world problem with ethical considerations.
5. Optimize models through hyperparameter tuning and interpret results effectively.

COURSE CONTENTS

Unit 1: Introduction to Machine Learning (8 Hours)

Fundamentals of machine learning, definition, scope, and types including supervised, unsupervised, and reinforcement learning. Basic concepts and terminology such as features, labels, training and testing data, overfitting, underfitting, bias, variance, and evaluation metrics like accuracy, precision, recall, and F1-score. **Data preprocessing techniques:** data cleaning, handling missing values, feature scaling, normalization, train-test split, and cross-validation.

Unit 2: Supervised Learning (8 Hours)

Supervised learning techniques, linear regression, loss functions (MSE, MAE), and optimization using gradient descent, implemented with Python (scikit-learn), **Classification algorithms:** logistic regression for binary and multiclass classification, decision trees, k-nearest neighbors (k-NN), and classification evaluation metrics, building and evaluating simple supervised models.

Unit 3: Unsupervised Learning (8 Hours)

Unsupervised learning, clustering techniques such as k-means clustering, hierarchical clustering with dendrograms, and clustering evaluation using silhouette score, dimensionality reduction, including Principal Component Analysis (PCA), its concepts and use cases, and the distinction between feature selection and feature extraction, visualizing high-dimensional data. Hands-on practice involves implementing clustering and PCA on sample datasets.

Unit 4: Advanced Topics in Machine Learning (8 Hours)

Advanced machine learning concepts, including ensemble methods like bagging, boosting, random forests, and gradient boosting machines, with practical implementation on real datasets, Support Vector Machines (SVM), including linear and non-linear SVM and the kernel trick.

Unit 5: Practical Applications and Model Deployment (8 Hours)

Practical aspects of machine learning, model evaluation and tuning through hyperparameter tuning using grid search and random search, cross-validation techniques, and regularization (L1, L2) to avoid overfitting,

Text Book:

TB1. "Understanding Machine Learning: From Theory to Algorithms" by Shai Shalev-Shwartz and Shai Ben-David
TB2. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili
TB3. "Machine Learning for Dummies" by John Paul Mueller and Luca Massaron

Reference Books:

RB1. "Pattern Recognition and Machine Learning" by Christopher M. Bishop
RB2. "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy
RB3. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron
RB4. "Introduction to Machine Learning" by Ethem Alpaydin
RB5. "Machine Learning: The Art and Science of Algorithms that Make Sense of Data" by Peter Flach
RB6. "Applied Predictive Modeling" by Max Kuhn and Kjell Johnson

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall fundamental concepts of machine learning, including supervised/unsupervised learning, evaluation metrics, and preprocessing techniques.
CO2	Explain the working principles of ML algorithms (e.g., linear regression, k-means, SVM) and their mathematical foundations (e.g., loss functions, gradient descent).
CO3	Implement ML models (classification, regression, clustering) using Python libraries (scikit-learn) and evaluate their performance on real datasets.
CO4	Compare and contrast different ML algorithms (e.g., decision trees vs. SVM) by analyzing bias-variance tradeoffs, model complexity, and performance metrics.
CO5	Assess the suitability of ML models for real-world problems by interpreting results, diagnosing overfitting/underfitting, and tuning hyperparameters.
CO6	Design an end-to-end ML pipeline (data preprocessing, model selection, deployment) to solve a societal or domain-specific problem.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1
CO2	3	3	1	1	2	1	1	1	1	1	1	1	2	3	1	1
CO3	2	2	3	2	3	1	1	2	2	1	2	1	3	3	2	1
CO4	2	3	2	3	2	1	1	1	2	1	1	1	2	3	1	2
CO5	1	2	2	3	3	2	1	2	2	2	2	1	2	2	3	3
CO6	1	2	3	3	2	2	2	3	3	3	3	3	2	2	3	3
AVG	2	2.2	2	2.2	2.2	1.3	1.2	1.7	1.8	1.5	1.5	1.3	2.3	2.3	2	2

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA304.2**				
Course Name	**: FULL STACK WEB DEVELOPMENT**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Develop proficiency in building responsive web applications using modern frameworks like React for frontend and Node.js/Express for backend.
2. Apply problem-solving skills to design and implement full-stack solutions that integrate databases (MongoDB) and RESTful APIs.
3. Evaluate and optimize web applications for performance, security, and scalability while adhering to industry best practices.
4. Demonstrate collaborative development practices through version control (Git) and team-based project work.
5. Create and deploy functional full-stack applications that address real-world problems with consideration for ethical and societal impacts.

COURSE CONTENTS

Unit 1: Introduction to Web Development (8 Hours)

Basics of web development, understanding client-server architecture, frontend vs. backend development, tools such as VS Code, Git, and Chrome DevTools, HTML structure, tags, and semantics, CSS styling, flexbox, and grid, design principles.

Unit 2: JavaScript Fundamentals (8 Hours)

Core JavaScript concepts, variables, data types, and operators, functions, loops, and conditionals, DOM manipulation and events, JavaScript features like arrow functions, destructuring, and modules, working with arrays and objects.

Unit 3: Frontend Development with React (10 Hours)

Frontend development using React, covering components, props, and state, JSX and rendering, handling events and forms, React Router for navigation, state management with useState and useEffect, fetching data from APIs.

Unit 4: Backend Development with Node.js & Express (8 Hours)

Backend development, setting up a Node.js server, creating RESTful APIs with Express, handling HTTP requests and responses, introduction to MongoDB, CRUD operations with MongoDB, connecting frontend to backend.

Unit 5: Deployment and Best Practices (6 Hours)

Deployment and best practices, including deploying frontend, deploying backend, environment variables and configuration, writing clean and modular code, basic error handling and debugging,

Text Books:

TB1. S. Vidhya, R.Vijaya Lakshmi, Lakshmi Kumari CH, SK. Arshiya Julma, P. Naveena, "FULL STACK DEVELOPMENT" SPH

TB2. Thomas Powell, "HTML & CSS: The Complete Reference",

Reference Books:

RB1. Philip Ackermann, "Full Stack Web Development: The Comprehensive Guide", SPD

Course Outcomes (CO):

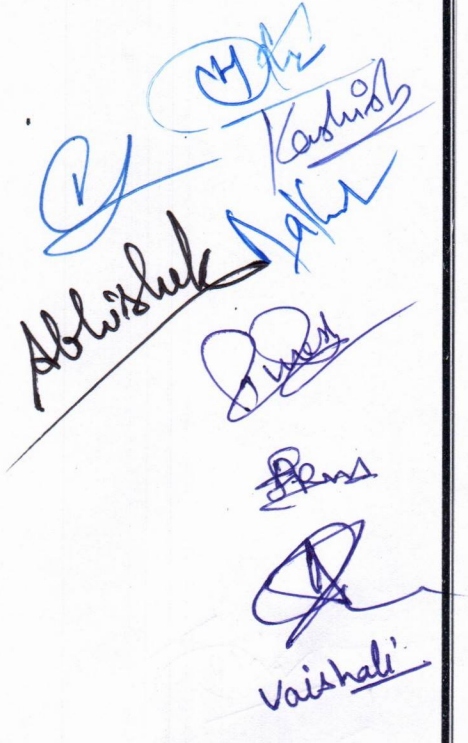
After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall core concepts of HTML, CSS, JavaScript, and client-server architecture.
CO2	Explain the role of React, Node.js, and Express in building full-stack applications.
CO3	Develop interactive frontend UIs using React and connect them to backend APIs (Node.js/Express).
CO4	Evaluate and debug full-stack applications by identifying performance bottlenecks and security risks.
CO5	Assess the effectiveness of deployed solutions based on scalability, usability, and industry best practices.
CO6	Design and deploy a secure, full-stack web application (frontend + backend + database) to solve a real-world problem.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	-	2	-	-	-	-	-	-	-	3	1	-	-
CO2	3	2	1	-	1	-	-	-	-	-	-	-	2	2	1	-
CO3	2	2	3	1	2	1	-	1	1	1	1	-	3	3	2	1
CO4	1	3	2	2	3	2	-	-	-	1	-	-	2	3	1	2
CO5	-	2	1	1	2	3	1	-	2	3	-	-	1	1	2	1	3
CO6	1	3	3	2	3	2	2	3	2	3	3	2	3	3	3	3
AVG	1.7	2.2	1.8	1.2	2.2	1.3	0.5	0.7	0.8	1.3	0.7	0.5	2.3	2.3	1.3	1.8

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated



Course code	**: MCA305.1**				
Course Name	**: INTRODUCTION TO DEEP LEARNING**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Develop a strong understanding of core deep learning concepts, including neural network architectures, activation functions, and optimization techniques.
2. Gain practical skills in implementing and training deep learning models using modern frameworks like TensorFlow and PyTorch.
3. Learn to analyze and troubleshoot model performance, including handling overfitting, tuning hyperparameters, and applying regularization methods.
4. Explore ethical considerations in AI, such as bias mitigation and societal impact, while designing deep learning solutions.
5. Enhance teamwork and communication skills through collaborative projects, presentations, and discussions on real-world AI applications.

COURSE CONTENTS

Unit 1: Fundamentals of Deep Learning (8 hours)

Core concepts of deep learning, definition, and applications, the structure and components of neural networks including neurons, layers, and activation functions, differences between supervised and unsupervised learning in deep learning, an introduction to tools and frameworks such as Python, NumPy, and TensorFlow/PyTorch,

Unit 2: Neural Network Architectures (8 hours)

Neural network architectures, including perceptrons and multi-layer perceptrons (MLPs), the mechanics of backpropagation and gradient descent optimization, an overview of loss functions and regularization techniques, an introduction to convolutional neural networks (CNNs) for image data.

Unit 3: Convolutional and Recurrent Neural Networks (8 hours)

Specialized neural network architectures such as convolutional neural networks (CNNs) with their convolution, pooling, and fully connected layers, applications of CNNs in image classification and object detection, recurrent neural networks (RNNs) for sequential data, an introduction to LSTMs and GRUs for time-series data.

Unit 4: Training and Optimization of Deep Learning Models (8 hours)

Training and enhancing deep learning models, including data preprocessing and augmentation for deep learning, hyperparameter tuning such as learning rate, batch size, and epochs, methods to prevent overfitting like dropout and weight decay, an introduction to transfer learning and pre-trained models.

Unit 5: Applications and Emerging Trends (8 hours)

Practical applications and recent advancements in deep learning, exploring its use in computer vision for tasks like image segmentation and face recognition, its applications in natural language processing for sentiment analysis and chatbots, an introduction to generative models such as autoencoders and GANs, ethical considerations and challenges in deep learning.

Text Book:

TB1. Deep Learning: A Practical Approach by Dr. Rajiv Chopra, Khanna Publishing House

TB2. Deep Learning: An Essential Guide to Deep Learning for Beginners by Herbert Jones

TB3. Deep Learning & Applications by Dr. A. Dennis Ananth, Dr.S. Markkandeyan, Dr. M. Rajakumaran, Dr. B. Lakshmi, Quill Tech Publications

TB4. Deep Learning By Example by Ahmed Menshawy by Packt

Reference Books:

RB1. Neural Networks and Deep Learning by Charu C. Aggarwal, Springer

RB2. Deep Learning By Example by Ahmed Menshawy, Packt

RB3. Deep learning Fundamentals and Applications for Beginners by Dr. A. GOPI KANNAN, SIP

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall core concepts of deep learning, neural network architectures (e.g., MLPs, CNNs, RNNs), and activation functions.
CO2	Explain the working of backpropagation, gradient descent, and regularization techniques (e.g., dropout, L2).
CO3	Implement neural networks (MLPs, CNNs, RNNs) using frameworks like TensorFlow/PyTorch for real-world datasets (e.g., MNIST, CIFAR-10).
CO4	Diagnose model performance issues (e.g., overfitting, vanishing gradients) and select appropriate optimization techniques.
CO5	Compare and justify the use of different architectures (e.g., CNN vs. Transformer) for tasks like image classification or NLP.
CO6	Design an end-to-end deep learning solution (e.g., chatbot, object detector) integrating ethical considerations (e.g., bias mitigation).

CO-PO Mapping:

	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	-	-	-	-	-	-	-	3	1	-	1
CO2	2	3	1	2	1	-	-	-	-	-	-	-	2	3	-	1
CO3	3	2	3	2	3	1	-	2	1	1	2	-	3	2	2	2
CO4	2	3	2	3	2	1	1	1	-	1	1	-	2	3	1	3
CO5	1	2	1	2	2	2	-	1	2	3	1	2	1	2	2	2
CO6	1	1	3	2	3	3	2	3	3	3	3	3	2	1	3	3
AVG	**2**	**2**	**1.8**	**2**	**2.2**	**1.2**	**0.5**	**1.2**	**1**	**1.3**	**1.2**	**0.8**	**2.2**	**2**	**1.3**	**2**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA305.2**				
Course Name	**: BLOCKCHAIN TECHNOLOGY**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives: The objectives of this course are

In this course, the learners will be able to develop expertise related to the following: -

1. To provide Knowledge and concept of cryptography and blockchain.
2. To make familiar with Bitcoin and cryptocurrency with its various impact on blockchain.
3. To know about the Ethereum technology with its multiple aspects.
4. To provide an understanding about and Smart Contracts.
5. To develop an understanding about the various applications of Blockchain.

Course Content

Unit-I (10 hours)

Distributed systems, Blockchain, Generic elements of a blockchain, Benefits and limitations of blockchain, Tiers of blockchain technology, Features of a blockchain, Types of blockchain, Consensus,

Unit-II (10 hours)

Decentralization: Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization, Smart contracts, Decentralized Organizations,

Unit-III (10 hours)

Cryptography: Symmetric Cryptography, Confidentiality, Integrity, Authentication, Non-repudiation, Public Key Cryptography: Asymmetric cryptography, Public and private keys, Hash functions

Unit-IV (10 hours)

Introduction to Bitcoin and Ethereum. Discussing few case studies of blockchain i.e., Block chain in Financial services and Government services.

Text Books:

TB1. Imran Bashir, Mastering Blockchain, Packt

Reference Books

RB1. Daniel Drescher, Block chain basics A non-technical introduction in 25 steps, Après
RB2. Paul Vigna and Michael J.Casey. The Age of Cryptocurrency

Vaishali Kashish Abhishek

COURSE OUTCOMES (CO):

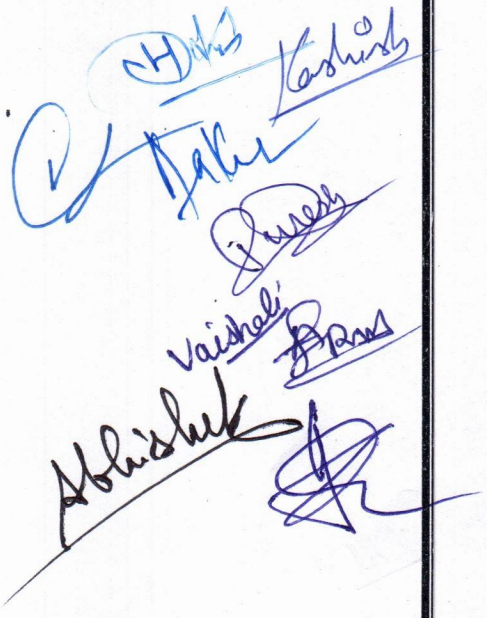
After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Identify and define the basic concepts of blockchain technology.
CO2	Describe the basic concepts, technology used for blockchain and primitives of the distributed computing and cryptography related to blockchain.
CO3	Illustrate the security features in blockchain technologies and ways of achieving it.
CO4	Analyze the use of decentralization, consensus mechanism, smart contract etc. used in block chain technology.
CO5	Evaluate some technologies based upon block chain.
CO6	Discuss case studies based on the blockchain technology.

CO-PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3						2			1			1	1		2
CO2	3	2	1		1		1			1			2	2		2
CO3	2	1	1	1	2	1	1			1		1	2	2		2
CO4	2	2	1	1	2	2	1			1			2	2		2
CO5	1				2		1						2	1		2
CO6	1		1	2	1		1						2	2	2	2
AVG	2.0	1.7	1.0	1.3	1.6	1.5	1.2			1.0		1.0	1.8	1.7	2.0	2.0

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated



Course code	**: MCA305.3**				
Course Name	**: CLOUD COMPUTING**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To learn basic concepts, types and characteristics of cloud computing
2. To learn Cloud Computing Architecture and service models.
3. To learn Virtualization and its types in cloud computing.
4. To learn fundamental concepts and architecture of cloud computing security.
5. To learn basics of SO and cloud-based storages.

COURSE CONTENTS

UNIT-I: Cloud Computing Fundamentals (5 hours)
Cloud Computing definition, benefits of cloud computing, characteristics of cloud, History of Cloud Computing, Cloud Architecture,

UNIT II: Cloud Computing Models and Services (5 hours)
Cloud deployment Model: private, public, community and hybrid cloud public vs private clouds, Cloud service model: IaaS, PaaS, SaaS. Challenges of cloud computing.

UNIT III: Cloud Virtualization (12 hours)
Virtualization, Characteristics, Virtualization in Cloud Computing, Pros and cons of Virtualization – Types of Virtualization –System Vm, Process VM, Virtual Machine monitor – Virtual machine properties. Hypervisors, Multitenancy, Types of Tenancy, Virtualization - Architecture Clustering, Grid Computing and Virtualization, Virtual Infrastructure, CPU Virtualization, Network and Storage Virtualization, Cloud Tools - VMware, Eucalyptus, Cloud Sim, Open nebula.

UNIT IV: Security In Cloud Computing (9 hours)
Security Issues in Cloud Computing: Introduction, Security Challenges in Cloud Computing, Information Security, Privacy and Trusting Cloud Computing, Cloud Identity and Access Management (IAM), Authentication and Authorization with Cloud, Software as a security service

UNIT V: Storage On Cloud (9 hours)
Service-Oriented Architecture, **Components of SOA,** Introduction to Storage Systems, Cloud Storage Concepts, Distributed File Systems: HDFS, Ceph FS, Cloud Databases: HBase, MongoDB, Cassandra, DynamoDB

Text Books:

TB1. Gautam Shroff, "Enterprise Cloud Computing Technology Architecture Applications", Cambridge University Press
TB2. Dimitris N. Chorafas, "Cloud Computing Strategies" CRC Press

Reference Books:

RB1. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach" McGraw-Hill Osborne Media

Course Outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Defining the basic concepts, principles and paradigm of Cloud Computing
CO2	Understanding of various Cloud computing models and services.
CO3	Analyzing the significance of implementing virtualization techniques.
CO4	Evaluate the need of security in Cloud computing.
CO5	Interpret the concept SOA and cloud-based storage in Cloud computing.
CO6	Create different cloud databases in Cloud Computing.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	2	1	1	1	2						2			1
CO2	2		1	2	2		1						2			2
CO3	3	1	2		2	1	2	1					2	1		2
CO4	2	1	1	1	1		1						2	1	1	2
CO5	2	2		2	1	1		1					2		1	2
CO6	2	1	1		2		2	2	1		1	1	2	1	1	
AVG	2.3	1	1.1	1.2	1.3	0.6	1.3	0.6	0.16	0	0.16	0.16	2	0.5	0.5	1.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA306.1**				
Course Name	**: RESEARCH METHODOLOGY AND PUBLICATION ETHICS**				
Semester / Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Understand the research process and its significance in computer applications.
2. Familiarize students with various research methods and designs.
3. Develop skills to design research studies and formulate research questions.
4. Gain proficiency in data collection, analysis, and interpretation.
5. Enhance critical evaluation and interpretation of research literature.
6. Develop effective research reporting and presentation skills.
7. Foster ethical conduct and responsible research practices.
8. Stimulate creativity in proposing innovative research directions.

COURSE CONTENTS

Unit 1: Foundations of Research and Ethics (8 hours)

Definition and Scope: Importance of research in computer science, types of research (basic, applied, experimental).

Research Ethics: Ethical considerations in research, responsible conduct, research misconduct, and ethical violations.

Ethical Guidelines: Overview of codes of conduct (e.g., ACM Code of Ethics), role of professional organizations in computer science.

Social Implications: Impact of research on society, ethical issues in emerging areas (e.g., AI, data science, cybersecurity).

Components of a Research Study: Key elements and structure of a research project.

Unit 2: Research Design and Data Collection (10 hours)

Research Problem and Design: Identifying and formulating research problems, objectives, questions/hypotheses, variables, and operationalization.

Research Designs: Exploratory, descriptive, experimental designs, and sampling techniques (including sample size determination).

Data Collection Methods: Primary and secondary data, quantitative methods (surveys, experiments, observation), qualitative methods (interviews, focus groups, case studies).

Data Quality: Validity, reliability, questionnaire design, and survey construction.

Ethical Data Handling: Informed consent, ethical review boards, data privacy, and handling human participants.

Unit 3: Data Analysis and Interpretation (8 hours)

Quantitative Analysis: Descriptive and inferential statistics, use of statistical tools (SPSS, R, Excel).

Qualitative Analysis: Content analysis, thematic analysis.

Interpreting Findings: Drawing conclusions, synthesizing results, and identifying research gaps.

Critical Evaluation: Reviewing research literature, assessing research quality, and proposing future research directions.

Unit 4: Academic Writing and Publication (8 hours)

Research Paper Structure: Components (title, abstract, introduction, methodology, results, conclusion), literature review, and referencing.

Writing and Formatting: Clear and concise writing, citation styles (APA, IEEE, MLA), formatting research papers and proposals.

Publication Process: Types of publications (conference papers, journal articles), steps in the publication process.

Responsible Authorship: Authorship guidelines, collaboration, authorship order, handling disputes, conflicts of interest, and funding disclosure.

Unit 5: Research Integrity and Professional Development (6 hours)

Plagiarism and Intellectual Property: Understanding plagiarism, avoidance techniques (paraphrasing, citing), intellectual property, copyrights, and patents.

Research Metrics and Databases: Citation databases (Web of Science, Scopus), metrics (impact factor, h-index, g-index, i10-index, altmetrics), journal metrics (SNIP, SJR, IPP, CiteScore).

Career Development: Strategies for successful publication, evaluating publication impact, building a research profile, and networking in the research community.

Publishing Controversial Research: Ethical considerations and responsible dissemination.

Text Books:

TB1. Bagchi, Kanak Kanti (2007) Research Methodology in Social Sciences: A Practical Guide, Delhi, Abijeet Publications.

TB2. Kothari, C.R, Research Methodology: An Introduction, Delhi, New Age.

Reference Books:

RB1. Cooper, R. Donald and Pamela S. Schindler, Business Research Methods, Delhi, Tata McGraw-Hill.

RB2. Flyvbjerg, Bent, Making Social Science Matter: Why Social Inquiry Fails and How it can Succeed Again, United Kingdom, Cambridge University Press.

RB3. Goodde and Hatte, Methods in Social Research, New York, McGraw – Hill.

Course Outcome (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall key ethical principles (e.g., plagiarism, informed consent) and research types in computer science.
CO2	Explain research design components (hypotheses, variables, sampling) and ethical guidelines.
CO3	Design a research methodology (quantitative/qualitative) and apply ethical data collection techniques (surveys, interviews).
CO4	Critique research papers for validity, reliability, and ethical compliance using statistical/content analysis tools (SPSS/R).
CO5	Assess authorship disputes, plagiarism cases, and societal impacts of CS research (e.g., AI bias, data privacy).
CO6	Develop a research paper with proper structure (abstract, literature review, citations) and justify methodological choices.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	1	-	2	-	3	1	-	1	1	-	1	-	-	-	2
CO2	2	2	-	3	-	3	1	-	2	1	-	1	-	1	-	3
CO3	2	3	2	3	2	3	2	1	2	2	1	2	2	3	2	3
CO4	2	3	1	3	3	3	2	1	2	2	1	2	1	3	1	3
CO5	1	2	-	2	1	3	2	-	2	3	-	3	-	1	-	3
CO6	2	3	2	3	2	3	2	2	3	2	2	2	2	2	2	3
AVG	**1.67**	**2.33**	**1.25**	**2.67**	**1.6**	**3**	**1.67**	**1**	**2**	**1.83**	**1**	**1.83**	**1.25**	**1.83**	**1.25**	**2.83**

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA306.2**				
Course Name	**: SOFTWARE PROJECT MANAGEMENT**				
Semester / Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Understand the fundamental concepts and principles of software project management.
2. Gain knowledge of project initiation, planning, and estimation techniques.
3. Develop skills in project organization, team management, and effective communication.
4. Apply project execution and monitoring methodologies, such as Waterfall and Agile.
5. Analyze project risks and implement strategies for risk mitigation.
6. Demonstrate the ability to create project charters, scope statements, and schedules.

COURSE CONTENTS

Unit 1: Fundamentals of Software Project Management (8 hours)
Introduction to Software Project Management (SPM), Characteristics of software projects, Role and responsibilities of a software project manager, Project initiation: Feasibility analysis, Project selection, Project charter and scope statement, Project estimation techniques: Expert judgment, Analogy, Parametric, Function Point Analysis, COCOMO Model

Unit 2: Project Planning and Scheduling (9 hours)
Work Breakdown Structure (WBS), Project scheduling: Gantt Charts, PERT, CPM, Resource allocation and leveling, Project risk management: Risk identification, assessment, prioritization, mitigation strategies, and control

Unit 3: Project Execution, Monitoring, and Closure (9 hours)
Project execution methodologies: Waterfall, Agile, and Hybrid models, Project tracking and performance monitoring, Change management, Earned Value Management (EVM), Project closure activities: Deliverable verification, Lessons learned, Knowledge transfer, Post-implementation review, Maintenance

Unit 4: Quality Assurance, Testing, and Configuration Management (7 hours)
Software quality concepts, Software testing techniques: Test planning, execution, Black-box, White-box, Unit and Integration testing, Software configuration management: Baselines, Version control systems (Git, SVN), Change and release management

Unit 5: Team Management, Communication, Documentation, and Tools (6 hours
Project roles, team building, and team dynamics, Effective project communication, Project documentation: Requirements, Design, Test cases, User manuals, Document configuration management, Project management and collaboration tools: Microsoft Project, JIRA, Trello, Slack, GitHub

Text Books:

TB1. "Software Project Management: A Process-Driven Approach" by Ashfaque Ahmed
TB2. "Quality Software Project Management", Robert T. Futrell, Donald F. Shafer, and Linda I.. Shafer, Pearson Education Asia.
TB3. "Managing Global Software Projects", Ramesh Gopalaswamy, Tata McGraw-Hill

Reference Books:

RB1. "Software Project Management: A Unified Framework" by Walker Royce
RB2. "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)" by Project Management Institute (PMI)
RB3. "Project Management: A Systems Approach to Planning, Scheduling, and Controlling" by Harold Kerzner
RB4. "Managing the Software Process" by Watts S. Humphrey

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall key concepts of software project management, including estimation techniques (COCOMO, FPA), scheduling methods (PERT/CPM), and risk management strategies.
CO2	Explain the phases of software project lifecycle (initiation, planning, execution, closure) and compare methodologies (Waterfall, Agile, Hybrid).
CO3	Apply project management tools (WBS, Gantt charts, EVM) and version control systems (Git) to plan and monitor software projects.
CO4	Analyze risks, resource constraints, and performance metrics (SPI, CPI) to troubleshoot project deviations.
CO5	Assess software quality through test cases (black-box/white-box) and recommend improvements for project deliverables.
CO6	Design a comprehensive project plan (charter, WBS, risk register) using industry tools (JIRA, MS Project) for a real-world scenario.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	1	2	1	1	1	1	1	1	1	2	2	1	1
CO2	2	3	2	1	2	1	1	2	2	2	2	1	2	3	2	1
CO3	3	3	3	2	3	1	1	2	2	1	2	1	3	3	3	2
CO4	2	3	2	2	3	2	1	2	2	1	2	1	2	3	2	2
CO5	2	3	2	2	2	3	1	1	2	2	1	2	2	3	2	2
CO6	3	3	3	3	3	2	2	3	3	3	3	3	3	3	3	3
AVG	2.5	2.8	2.2	1.8	2.5	1.7	1.2	1.8	2	1.7	1.8	1.5	2.3	2.8	2.2	1.8

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCA306.3**				
Course Name	**: THEORY OF AUTOMATA**				
Semester /Year	**: IIIrd Semester**				
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Master foundational concepts of automata theory, including finite automata (DFA/NFA), regular expressions, and context-free grammars.
2. Design and convert computational models (e.g., regex to NFA, CFG to PDA) to solve language recognition problems.
3. Analyze language properties using formal tools like the Pumping Lemma and explore computability limits (e.g., halting problem).
4. Apply automata theory to real-world scenarios (e.g., compiler design, pattern matching) and evaluate model trade-offs.
5. Develop problem-solving skills through hands-on exercises, from constructing Turing machines to proving decidability.

COURSE CONTENTS

Unit 1: Introduction to Automata and Finite Automata (8 Hours)

Overview of automata theory and its applications, basic concepts such as alphabets, strings, and languages, finite automata including definitions and types (DFA, NFA), designing Deterministic Finite Automata (DFA), designing Non-deterministic Finite Automata (NFA), equivalence of DFA and NFA. Tutorials include exercises on designing DFAs and NFAs for simple patterns like strings ending with '01'.

Unit 2: Regular Expressions and Regular Languages (8 Hours)

Introduction to regular expressions, constructing regular expressions for simple languages, conversion between regular expressions and finite automata, properties of regular languages focusing on closure properties, pumping lemma for regular languages, applications of regular expressions such as pattern matching. Tutorials involve practice writing regular expressions and converting them to DFAs/NFAs.

Unit 3: Context-Free Grammars and Pushdown Automata (8 Hours)

Introduction to context-free grammars (CFGs), designing CFGs for simple languages, parse trees and derivations, pushdown automata (PDA) with definitions and design, equivalence of CFGs and PDAs, applications of CFGs such as syntax analysis in compilers. Tutorials focus on exercises constructing CFGs and designing PDAs for languages like balanced parentheses.

Unit 4: Turing Machines and Computability (8 Hours)

Introduction to Turing machines with definitions and components, designing Turing machines for simple problems, variants of Turing machines such as multi-tape, concept of decidability and undecidability, introduction to the halting problem. Tutorials include simulating Turing machines for basic computations like string copying.

Unit 5: Applications and Review (8 Hours)

Applications of automata in compiler design, text processing, and verification, limitations of computation models, review of key concepts including finite automata, regular expressions, CFGs, PDAs, and Turing machines.

Text Books:

TB1. Hopcroft, "Introduction to Automata Theory, Languages, and Computation, Pearson

TB2. Arvind "An Introduction to Formal Languages and Automata"

TB3. Rajendra Kumar, "Theory of Automata Languages & Computation", Tata Mcgraw Hill

Reference Books:

RB1. Javier Esparza, Michael Blondin, " Automata Theory: An Algorithmic Approach", MIT Press

RB2. Johannes A. Buchmann, "Introduction to Cryptography" Springer-Verlag

RB3. Pedrycz, Witold, "Automata Theory and Formal Languages", DE GRUYTER

Course Outcomes (CO):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall fundamental concepts of automata theory, including finite automata (DFA/NFA), regular expressions, context-free grammars, and Turing machines.
CO2	Explain the working of computational models (DFA, NFA, PDA, TM) and their equivalence relationships.
CO3	Construct finite automata, regular expressions, and context-free grammars for specified languages, and convert between equivalent models.
CO4	Prove language properties using the Pumping Lemma and analyze decidability/undecidability of problems using Turing machines.
CO5	Compare the computational power of automata models (regular vs. context-free vs. recursive) and assess limitations (e.g., non-regular languages).
CO6	Design Turing machines or PDAs to solve computational problems and propose automata-based solutions for real-world applications (e.g., compilers, pattern matching).

CO-PO Mapping:

	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	1	1	1	1	1	1	1	3	1	1	1
CO2	3	2	1	1	2	1	1	1	1	1	1	1	3	2	1	1
CO3	2	2	3	1	3	1	1	1	1	1	1	1	3	3	2	1
CO4	1	3	1	3	2	1	1	1	1	2	1	1	1	3	1	3
CO5	1	3	1	2	1	1	1	1	1	3	1	2	1	2	2	2
CO6	1	2	3	2	2	2	1	3	2	3	2	2	2	3	3	3
AVG	1.8	2.2	1.7	1.7	2	1.2	1	1.3	1.2	1.8	1.2	1.3	2.2	2.3	1.7	1.8

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAP31**			
Course Name	**: C# DOT NET LAB**			
Semester /Year	**: IIIrd Semester**			
	L	**T**	**P**	**C**
	0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To learn and understand different types of statements in C#.
2. Use of data base for making dynamic websites using C# programming
3. Programming concepts in .Net Framework.
4. Data base connectivity using ADO.net
5. Understand and use of different graphical tools for the development of web page and website using C# programming.

COURSE CONTENTS

1. Write a C sharp program to generate prime numbers between 1 to200 and also print to the console. (ex. 1,2,3,5.....199).
2. Write a program to print ARMSTRONG number.
3. Write a C sharp program using loop that examines all the numbers between 2 and 1000, and displays only Perfect numbers.(A perfect number is the one whose sum of their divisors equals the number itself).For example given the number 6, the sum of its divisors is 6(1+2+3).Hence, 6 is a perfect number.
4. Write a C sharp program to accept an array of integers (10) and sort them in ascending order.
5. Write a program to implement the concept of abstract class.
6. Write a program to implement the concept of sealed class.
7. Write a C sharp program for jagged array and display its item through foreach loop.
8. Write a program in C Sharp using a class that gets the information about employee's such as Emp Id, First Name, Last Name, Basic Salary,Grade,Address, Pin Code and Contact Number. Write a method that calculates the Gross Salary (Basic +DA+HRA) and returns to the calling program and another method for the Net salary (Gross - (P.F + Income Tax)).Finally write a method that prints, a pay slip of an employee, containg all the above components in a proper format to the console.(Grade A = 20,000 , B=15,000 and C=10,000) DA=56% and HRA=20%., Pf=780, ITax.
9. Write a program to demonstrate boxing and unboxing.
10. Write a program to find number of digit, character, and punctuation in entered string.
11. Write a program using C# for exception handling.
12. Write a program to implement multiple inheritances using interface.
13. Write a program in C# using a delegate to perform basic arithmetic operations like addition, subtraction, division, and multiplication.
14. Write a program to get the user's name from the console and print it using different namespace.
15. Demonstrate the concept of Multithreading using locks in C Sharp
16. Write a program to implement Indexer.
17. Write a program to design two interfaces that are having same name methods how we can access these methods in another class.
18. Write a program to implement method overloading.

19. Write a program to implement method overriding
20. Write a program in C sharp to create a calculator in windows form.
21. Create a front end interface in windows that enables a user to accept the details of an employee like EmpId ,First Name, Last Name, Gender, Contact No, Designation, Address and Pin. Create a database that stores all these details in a table. Also, the front end must have a provision to Add, Update and Delete a record of an employee.
22. Create a database named MyDb (SQL or MS Access).Connect the database with your window application to display the data in List boxes using Data Reader.
23. Write a program using ADO.net to insert, update, delete data in back end
24. Display the data from the table in a DataGridView control using dataset.
25. Create a registration form in ASP.NET and use different types of validation controls.
26. Display the data from the table in a Repeater control using dataset in ASP.net.

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Remember the basics of C# programming, different graphics tools and their use.
CO2	Understand of static and dynamic web pages using standard tools and learn various properties of the tools.
CO3	Develop interactive and user friendly websites using front end and back end programming.
CO4	To develop, implement and creating Applications with ADO.NET and SQL server
CO5	Create user interactive web pages using ASP.Net and xml.
CO6	Create console, windows and wed applications

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1			3	1	2					2			1
CO2	3	1	2	3		3	2	3					2			2
CO3	3	2	3	3	1	2	2	3					3	1		1
CO4	2	1	3	3	2	3	2	3					3		1	1
CO5	3	1	3	1	2	3	2	3					3	2	1	1
CO6	2	3	2	2			1	1	2	1			3	1	1	1
AVG	2.7	1.5	2.3	2	0.8	2.3	1.7	2.5	0.3	0.2	0	0	2.7	0.7	0.5	1.2

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAP34.1**			
Course Name	**: MACHINE LEARNING LAB**			
Semester /Year	**: IIIrd Semester**			
L	**T**	**P**	**C**	
0	**0**	**2**	**1**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Implement and evaluate supervised and unsupervised learning algorithms using real-world datasets in Python.
2. Preprocess datasets by handling missing values, scaling features, and splitting data for model training and testing.
3. Analyze model performance using appropriate evaluation metrics (e.g., accuracy, precision, recall, RMSE) and visualization tools.
4. Optimize machine learning models through hyperparameter tuning and cross-validation techniques.
5. Develop and present a complete ML pipeline (from data cleaning to deployment) for a practical problem.

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall and explain key Python libraries (NumPy, Pandas, scikit-learn) and their functions for ML tasks.
CO2	Interpret dataset characteristics, preprocessing steps, and algorithm parameters for given ML problems.
CO3	Implement supervised (e.g., regression, classification) and unsupervised (e.g., clustering) algorithms using real-world datasets.
CO4	Compare model performance by visualizing results (e.g., confusion matrices, PCA plots) and diagnosing errors (e.g., overfitting).
CO5	Assess model robustness using cross-validation, hyperparameter tuning, and metric selection (e.g., F1-score, RMSE).
CO6	Design and deploy an end-to-end ML pipeline (data cleaning → modeling → evaluation) for a novel problem.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	1	2	1	1	1	1	1	1	1	3	1	1	1
CO2	2	3	1	2	2	1	1	1	1	1	1	1	2	3	1	1
CO3	2	2	3	2	3	1	1	2	2	1	2	1	3	3	2	1
CO4	2	3	2	3	2	1	1	1	2	1	1	1	2	3	1	2
CO5	1	2	2	3	3	2	1	2	2	2	2	1	2	2	3	3
CO6	1	2	3	3	2	3	2	3	3	3	3	3	2	2	3	3
AVG	1.8	2.2	2	2.3	2.3	1.5	1.2	1.7	1.8	1.5	1.5	1.3	2.3	2.3	2	2

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAP34.2**			
Course Name	**: FULL STACK WEB DEVELOPMENT LAB**			
Semester /Year	**: IIIrd Semester**			
	L	**T**	**P**	**C**
	0	**0**	**2**	**1**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. Develop hands-on proficiency in building responsive web interfaces using HTML/CSS, JavaScript, and React through practical implementation exercises.
2. Design and implement functional backend APIs using Node.js and Express, integrating with databases (MongoDB) to perform CRUD operations.
3. Demonstrate the ability to connect frontend and backend systems, handling data flow and API communication in full-stack applications.
4. Apply debugging and testing techniques to identify and resolve issues in both client-side and server-side code.
5. Create and deploy a complete full-stack web application that follows industry standards for security, performance, and usability.

Course Outcomes (COs):

After completion of the course, a student will be able to

CO#	Detailed Statement of the CO
CO1	Recall and demonstrate the use of HTML/CSS syntax, JavaScript fundamentals, and React component structure in lab exercises.
CO2	Explain the workflow of client-server communication and RESTful API integration in full-stack applications.
CO3	Develop functional frontend interfaces (React) and backend APIs (Node.js/Express) with database connectivity (MongoDB).
CO4	Debug and troubleshoot full-stack applications by identifying errors in code, API responses, or database queries.
CO5	Test and validate the security, performance, and usability of deployed applications against industry standards.
CO6	Design and deploy a complete full-stack project (frontend + backend + database) to solve a real-world problem, demonstrating end-to-end integration.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1	-	-	2	-	-	-	-	-	-	-	3	1	-	-
CO2	2	2	1	-	2	-	-	-	-	1	-	-	2	2	1	-
CO3	3	2	3	1	3	1	-	1	1	1	1	-	3	3	2	1
CO4	1	3	2	2	3	2	-	-	-	1	-	-	2	3	1	2
CO5	-	2	1	1	3	3	1	-	1	2	-	1	1	2	1	3
CO6	2	3	3	2	3	2	2	3	2	3	3	2	3	3	3	3
AVG	1.8	2.2	1.7	1	2.7	1.3	0.5	0.7	0.7	1.3	0.7	0.5	2.3	2.3	1.3	1.8

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

Course code	**: MCAPR3**			
Course Name	**: PROJECT / RESEARCH PROJECT**			
Semester /Year	**: IIIrd Semester**			
L	**T**	**P**	**C**	
0	**0**	**6**	**3**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

In this course, the learners will be able to develop working expertise of solving complex computing problems through project based learning approach using real world case studies by implementing the concepts studied in the theory courses of this semester.

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Identify the problem and describing it.
CO2	Understand the requirements of the chosen project.
CO3	Apply the collected requirements to define the describe the project in a systematic and comprehensive approach.
CO4	Analyze the technical aspects of the chosen project to find the possible solutions for development of the project.
CO5	Evaluate the effective reports and documentation for all project related activities and solutions.
CO6	Create plan for the project development.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1						1	2		2	1	2	1	1	2
CO2	1	3		2	1	2			1				3	2	2	1
CO3	1	1	3	2	1	1	1			2		1	3	2	2	2
CO4	1	1	1	1	1	1	2	1	1	2	2	3	2		3	
CO5	2			1	3	1	2	1	2	1	3	1	2	1	2	2
CO6	2	3	3	3	1	1	1	3	1	1	1	1	2	1	2	2
AVG	1.7	1.8	2.3	1.8	1.4	1.2	1.5	1.5	1.4	1.5	2.0	1.4	2.4	1.2	2	1.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated

4th Semester

Course code	**: MCA401**				
Course Name	**: MOOC**				
Semester /Year	**: IVth Semester**				
		L	**T**	**P**	**C**
		0	**0**	**0**	**4**

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

1. To familiarize students with cutting-edge technologies through structured online learning modules.
2. To enhance practical skills by providing hands-on experience with tools and platforms relevant to modern computing applications.
3. To bridge the gap between academic knowledge and industry demands by incorporating real-world case studies and expert-led sessions.
4. To encourage self-paced learning and critical thinking by offering flexible access to high-quality MOOC resources.
5. To prepare students for emerging career opportunities by equipping them with certifications and competencies in trending technologies.

- Students are required to complete a 12-week MOOC course (equivalent to 4 credits) on a latest technology topic **not included in their current curriculum**, through the SWAYAM portal.
- Prior approval must be obtained from the department before enrolling.
- Upon completion, students must submit the course certificate and marksheet to the department for the credit to be transferred to their MCA program.
- Marksheet will reflect the Course Title of MOOC course done by student.

Course code	**: MCAPR4**				
Course Name	**: PROJECT / RESEARCH PROJECT**				
Semester /Year	**: IVth Semester**				
	L	**T**	**P**	**C**	
	0	**0**	**28**	**14**	

L - Lecture T – Tutorial P – Practical C – Credit

Course Objectives: The objectives of this course are

In this course, the learners will be able to develop working expertise of solving complex computing problems through project based learning approach using real world case studies by implementing the concepts studied in the theory courses of this semester.

Course outcomes (COs):

Upon successful completion of the course a student will be able to

CO#	Detailed Statement of the CO
CO1	Identify the problem and describing it.
CO2	Understand the requirements of the chosen project.
CO3	Apply the collected requirements to define the describe the project in a systematic and comprehensive approach.
CO4	Analyze the technical aspects of the chosen project to find the possible solutions for development of the project.
CO5	Evaluate the effective reports and documentation for all project related activities and solutions.
CO6	Create plan for the project development.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	1						1	2		2	1	2	1	1	2
CO2	1	3		2	1	2			1				3	2	2	1
CO3	1	1	3	2	1	1	1			2		1	3	2	2	2
CO4	1	1	1	1	1	1	2	1	1	2	2	3	2		3	
CO5	2			1	3	1	2	1	2	1	3	1	2	1	2	2
CO6	2	3	3	3	1	1	1	3	1	1	1	1	2	1	2	2
AVG	1.7	1.8	2.3	1.8	1.4	1.2	1.5	1.5	1.4	1.5	2.0	1.4	2.4	1.2	2	1.5

3: Highest Correlated, 2: Medium Correlated, 1: Lowest Correlated