Scheme & Syllabus

of

Master of Computer Applications (MCA)

(With effect from 2024-2026)

Outcome Based
EducationWith
Choice Based Education

¥ÀoÀåPÀæªÀÄ

(±ÉÊPÀëtÂPÀvÀµÀð 2024-2026)

P.E.S. College of Engineering

Mandya - 571 401, Karnataka
(An Autonomous Institution Affiliated to VTU, Belagavi
Grant -in- Aid Institution (Government of Karnataka), World Bank Funded College
(TEQIP)Accredited by NBA & NAAC and Approved by AICTE, New Delhi.)

¦.E.J¸ï. vÁAwæPÀ ªÀÄºÁ«zÁå®AiÀÄ
ªÀÄAqÀ¼–571 401, PÁðlPÀ
[«.n.AiÀÄÄ., ¨É¼UÁ« CrAiÀÄ°è ¸ÁAiÀÄvÀÛ ¸ÀA¸ÉÜ]

Ph : 08232- 220043, Fax : 08232 – 222075,Web : www.pescemandya.org

Department of Master of Computer Applications (MCA)

ABOUT THE DEPARTMENT

The department of MCA was started in 1992 in PES College of Engineering Mandya, one among the earliest institutes in Karnataka to start MCA Programme and successfully completes 31 years with excellence.The department is run by highly qualified teaching staff. The Three years Programme became autonomous in the year 2008 and is affiliated to Visvesvaraya Technological University and approved by AICTE. Now MCA is Two years course from the academic year 2020-2021. The department has been actively involved in conducting workshops, Technical Seminars from industry persons to provide a platform for sharing and spreading the latest developments in the field of Computer science and applications.

The curriculum for the Master of Computer Applications (MCA) program is designed to meet the need of the industry. Our alumni are placed across the world in MNC companies like HP, iGATE, IBM, EMPHASIS, 6D Technologies, NOKIA-SIEMENS, MINDTREE, TCS, ACCENTURE, INFOSYS, ORACLE, WIPRO, AERIES GLOBAL, TECH-MAHINDRA

VISION AND MISSION

Vision of PESCE

PESCE shall be a leading institution imparting quality engineering and management education developingcreative and socially responsible professionals.

Mission of PESCE

- Provide state of the art infrastructure, motivate the faculty to be proficient in their field of specialization and adopt best teaching-learning practices.
- Impart engineering and managerial skills through competent and committed faculty using outcomebased educational curriculum.
- Inculcate professional ethics, leadership qualities and entrepreneurial skills to meet the societalneeds.
- Promote research, product development and industry-institution interaction.

Vision of the Department

A Department of high repute imparting quality education to develop competent computer applicationsoftware professionals and technocrats to serve the society.

Mission of the Department

Committed to

- To provide state-of-the-art facilities with supportive environment for teaching and learning.
- To prepare the students with curricula of industry expectation.
- Train the students to be competent to solve the real- world problems in the field of computerApplications and nurturing the students with ethical values for well-being in the society.

ACADEMIC PROGRAM

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1. Deliver competence in a global environment as computer software professional with Practice of software engineering principles.

PEO-2. Exhibit Technical and managerial skills to provide solutions for societal acceptable problems andmanage projects.

PEO-3. Excel in profession with effective communication skills, ethical attitude, teamwork and ability torelate computer applications to broader societal context.

PROGRAMME OUTCOMES (POs)

PO-1. (Foundation Knowledge): Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

PO-2. (Problem Analysis): Identify, review, formulate and analyze problems for primarily focusing on customer requirements using critical thinking frameworks.

PO-3. (Development of Solutions): Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

PO-4. (Modern Tool Usage): Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

PO-5. (Individual and Teamwork): Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

PO-6. (Project Management and Finance): Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

PO-7. (Ethics): Commit to professional ethics in managing software projects with financial aspects, learn to use new technologies for cyber security and insulate customers from malware.

PO-8. (Life-long Learning): Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

CREDIT PATTERN

Professional Core Courses

I Semester 10 Credits

II Semester 16 Credits

Total credits for PCC are 26 credits

Integrated Professional Core Courses

I Semester 4 Credits

II Semester 4 Credits

Total credits for IPCC are 08 credits

Professional Core Courses Laboratory

I Semester 1 Credits

II Semester 2 Credits

Total credits for IPCL are 03 credits

Professional Electrical Courses (PEC) (Specialization/ Online) Courses

III Semester 09 Credits

IV Semester 03 Credits

Total credits for PEC courses are 12 credits

Basic Science Course (Mathematics) : 03 (I Semester)

Project Work : 15 Credits (III Semester)

Technical Seminar : 02 Credit **(**IV Semester)

Industry Internship: 11 Credits (IV Semester)

A total of 80 credits for 2 years MCA Programme

MASTER OF COMPUTER APPLICATIONS (MCA)
SCHEME OF TEACHING AND EXAMINATION 2024-2026
I SEMESTER

Sl. No	Course Code	Course Type	Course Title	Hrs/Week	Credits	Examination		
						Marks		
				L: T: P: H		CIE	SEE	Total
1	P24MCA11	IPCC	Programming and Problem Solving in C	3 : 0 : 2 : 5	4	50	50	100
2	P24MCA12	BSC	Discrete Mathematics and Graph Theory	2 : 1 : 0 : 3	3	50	50	100
3	P24MCA13	PCC	Database Management Systems (DBMS)	4 : 0 : 0 : 4	4	50	50	100
4	P24MCA14	PCC	Operating System	2 : 1 : 0 : 3	3	50	50	100
5	P24MCA15	PCC	Web Technologies	3 : 0 : 0 : 3	3	50	50	100
6	P24MCAL16	PCCL	DBMS and Web Technologies Laboratory	0 : 0 : 2 : 2	1	50	50	100
7	P24MCA17	NCMC	Research Methodology and IPR (Online Course)	-	0	-	-	--
8	P24MHU18	NCMC	Employability Enhancement Skills-1 (EES-1) *	0 : 2 : 0 : 2	0	50	-	100
9	P24MCA19	NCMC	Mathematics for MCA Students[#]	3 : 0 : 0 : 3	0	50	-	100
Total				**25**	**18**	**400**	**300**	**800**

Note: **BSC**-Basic Science Courses, **PCC**: Professional core. **IPCC**-Integrated Professional Core Courses, **PCC(PB):** Professional Core Courses (Project Based), **PCCL**-Professional Core Course lab**, NCMC**- Non Credit Mandatory Course, L-Lecture**, P**-Practical**, T/SDA**-Tutorial / Skill Development Activities.

24MCA17 Research Methodology and IPR (**Online**) - **None Credit Mandatory Course (NCMC)** if students have not studied this course in their undergraduate program, then he /she has to take this course at http://online.vtu.ac.in and to qualify for this course is compulsory before completion of the minimum duration of the program (Two years), however, this course will not be considered for vertical progression.

[#] **Bridge Course** Non-Credit Mandatory Course 24MCA18-Mathematics for MCA Students: Students who have not taken Mathematics at the **10+2 or degree level** are required to study and pass this course in the 1st semester. However, this course/subject will not be considered for vertical progression.

***** Taught by Humanities Science and Management Department

II SEMESTER

Sl. No	Course Code	Course Type	Course Title	Hrs/Week	Credits	Examination		
						Marks		
				L : T : P :H		CIE	SEE	Total
1	P24MCA21	IPCC	Machine Learning and Data Analytics using python	3 : 0 : 2 : 5	4	50	50	100
2	P24MCA22	PCC	Object Oriented Programming using JAVA	4 : 0 : 0 : 4	4	50	50	100
3	P24MCA23	PCC	Data Structure and Algorithms	4 : 0 : 0 : 4	4	50	50	100
4	P24MCA24	PCC	Software Engineering	3 : 1 : 0 : 4	4	50	50	100
5	P24MCA25	PCC	Web Application Development	3 : 1 : 0 : 4	4	50	50	100
6	P24MCAL26	PCCL	Object Oriented Programming using JAVA Laboratory	0 : 0 : 2 : 2	1	50	50	100
7	P24MCAL27	PCCL	Data Structure and Algorithms Laboratory	0 : 0 : 2 : 2	1	50	50	100
8	P24MHU28	NCMC	Employability Enhancement Skills-2 (EES-2)*	0 : 2 : 0 : 2	0	50	-	100
9	P24MCA29	NCMC	Ability Enhancement Courses with Seminar-I	-	0	-	-	-
		Total		**27**	**22**	**400**	**350**	**800**

Note: **BSC**-Basic Science Courses, **PCC**: Professional core. **IPCC**-Integrated Professional Core Courses, **PCC(PB):** Professional Core Courses (Project Based), **PCCL**-Professional Core Course lab**, NCMC**- Non-Credit Mandatory Course, **L**-Lecture**, P**-Practical**, T/SDA**-Tutorial / Skill Development Activities

***** Taught by Humanities Science and Management Department

24MCA29- Ability Enhancement Courses with Seminar-I - **None Credit Mandatory Course (NCMC)**, Students have to select the Topic like ERP, R Programming, Scripting language, Web Development Application, etc. They have to develop a small prototype and demonstrate to all the class.

I and II Semester Course Types Description:

BSC: **Basic Science Courses**: Courses like Mathematics/ Science are the prerequisite courses that the concerned engineering stream board of Studies will decide. **PCC**: **Professional Core Course**: Courses related to the stream of engineering, which will have both CIE and SEE components, students have to qualify in the course for the award of the degree.

Integrated Professional Core Course (IPCC): Refers to a Professional Theory Core Course Integrated with practicals of the same course. The IPCC's theory part shall be evaluated by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper.

PCCL: Professional Core Course Laboratory: Practical courses whose CIE will be evaluated by the class teacher and SEE will be evaluated by the two examiners.

Skill development activities: Under Skill development activities in a concerning course, the students should

1.. Involve in research/testing/projects to understand their problems and help creative and innovative methods to solve the problem.
2. Involve in case studies and field visits/ fieldwork.
3. Accustom to the use of standards/codes etc., to narrow the gap between academia and industry.
4. Handle advanced instruments to enhance technical talent.
5. Gain confidence in the modeling of systems and algorithms for transient and steady-state operations, thermal study, etc.
6. Work on different software/s (tools) to simulate, analyze and authenticate the output to interpret and conclude.

All activities should enhance student's abilities to employment and/or self-employment opportunities, management skills, Statistical analysis, fiscal expertise, etc. Students and the course instructor/s are to be involved either individually or in groups to interact together to enhance the learning and application skills of the study they have undertaken. The students with the help of the course teacher can take up relevant technical activities that will enhance their skills. The prepared report shall be evaluated for CIE marks.

III SEMESTER

Sl. No	Course Code	Couse Type	Course Title	Hrs/Week	Credits	Examination		
						Marks		
				L : T : P : H		CIE	SEE	Total
1	P24MCAX311X	PEC	Specializations	3 : 0 : 0 : 3	3	50	50	100
2	P24MCAX311X	PEC	Specializations	3 : 0 : 0 : 3	3	50	50	100
3	P24MCAX311X	PEC	Specializations	3 : 0 : 0 : 3	3	50	50	100
4	P24MCA32	PROJ	Project Work	-	15	100	100	200
Total				**09**	**24**	**250**	**250**	**500**

Note: **PEC**-Professional Elective Courses, **PROJ**- Project, **L**-Lecture**, P**-Practical**, T/SDA**-Tutorial / Skill Development Activities (Hours are for Interaction between faculty and students)

Specializations: Students can ***opt*** three specializations from the listed below

Sl. No.	Specialization	Data Science and Analytics (A)	Sl. No.	Specialization	Web Application Development (B)
	Course code	Course Title		Course code	Course Title
1	P24MCAA311A	Data Mining and Visualization	1	P24MCAB311A	Web Development using Full Stack open
2	P24MCAA311B	Big Data Analytics	2	P24MCAB311B	Rich Internet Application Development
3	P24MCAA311C	Business Data Analytics	3	P24MCAB311C	Web Development Using PHP and MySQL
4	P24MCAA311D	Enterprise Resource Planning	4	P24MCAB311D	Enterprise Application Programming
5	P24MCAA311E	Exploratory Data Analysis	5	P24MCAB311E	Advances in Web Technologies
6	P24MCAA311F	Social Media Analytics	6	P24MCAB311F	Web Programming using Java

Sl. No.	Specialization	Network and System Administration (C)	Sl. No.	Specialization	Software Development and Systems (D)
	Course code	Course Title		Course code	Course Title
1	P24MCAC311A	Computer Networks	1	P24MCAD311A	Management Information Systems
2	P24MCAC311B	Network and Linux Administration	2	P24MCAD311B	Database Design & Applications
3	P24MCAC311C	TCP/ IP	3	P24MCAD311C	Software Architectures
4	P24MCAC311D	Unix Shell Programming	4	P24MCAD311D	Computer Organization and Software Systems
5	P24MCAC311E	Cloud Essentials	5	P24MCAD311E	Software Design and Patterns
6	P24MCAC311F	Introduction to ERP and SAP Basis Administration	6	P24MCAD311F	Object-oriented Analysis & Design

Sl. No.	Specialization	Computer Networks and Cloud (E)	Sl. No.	Specialization	AI and ML (F)
	Course code	Course Title		Course code	Course Title
1	P24MCAE311A	Computer Networks	1	P24MCAF311A	Introduction to Generative AI
2	P24MCAE311B	Data Storage Technologies and Networks	2	P24MCAF311B	Artificial Neural Networks
3	P24MCAE311C	Design and Operation of Data Centers	3	P24MCAF311C	Natural Language Processing
4	P24MCAE311D	Wireless and Mobile Communication	4	P24MCAF311D	Deep Learning Fundamentals
5	P24MCAE311E	Software Defined Networks	5	P24MCAF311E	Introduction to Machine Learning
6	P24MCAE311F	Cloud Computing	6	P24MCAF311F	Computer Vision

Sl. No.	Specialization	IoT (G)	Sl. No.	Specialization	Security (H)
	Course code	Course Title		Course code	Course Title
1	P24MCAG311A	Data Management for IoT	1	P24MCAH311A	Ethical Hacking
2	P24MCAG311B	Networked Embedded Applications	2	P24MCAH311B	Cyber Security
3	P24MCAG311C	Cross Platform Application Development	3	P24MCAH311C	Cryptography and Network Security
4	P24MCAG311D	IoT Technology and Applications	4	P24MCAH311D	Blockchain Technologies
5	P24MCAG311E	Communication and Networking Technologies in IoT	5	P24MCAH311E	Database & Web Application Security
6	P24MCAG311F	Software and Programming in IoT	6	P24MCAH311F	Mobile and Wireless Security

Project work is a significant component aimed at fostering research, practical application of knowledge, and innovation. The evaluation process generally follows these steps:

1. **Selection and Approval of Project Work**:
   - Topic Selection: Students propose project topics, often in consultation with their faculty advisor.
   - Approval Process: The proposed topic is submitted for approval by a project committee or department, ensuring alignment with academic standards and relevance.

2. **Project Execution**:
   - Research and Development: Students carry out research, experiments, or development work as per the project plan.
   - Periodic Reviews: Regular progress reviews are conducted by faculty to monitor the project's progress and provide feedback.
   - Documentation: Students maintain a detailed record of their methodology, data, results, and analysis.

3. **Submission of the Project Report:**
   - Format and Guidelines: The report must follow the prescribed format by the university or department.
   - Plagiarism Check: The report is often checked for plagiarism to ensure originality.

4. **Evaluation Process:**
   - Internal Evaluation: Faculty members from the department review the project report and presentation for content quality, innovation, and depth of research.
   - External Evaluation: An external examiner, often an industry expert or academician from another institution, reviews the project.
   - Viva Voce Examination: The student defends their project work before a panel comprising internal and external examiners. This assesses their understanding, analytical ability, and application of the project work.

5. **Grading Criteria (Guidelines only)**

- Report Quality: Depth of research, organization, and clarity of the document.
- Presentation Skills: Effectiveness in communicating key aspects of the project.
- Technical Merit: Innovation, accuracy, and the applicability of the research.
- Viva Performance: Understanding of the subject, responses to questions, and ability to discuss the work effectively.

6. **Final Outcome:**

- Marks Allocation: Typically, evaluation is a blend of internal (guided by the department) and external (examiner's input) assessments, distributed over the report, presentation, and viva.
- Pass Requirement: Students must meet a minimum threshold to pass, as per University / Department.

This structured evaluation ensures a comprehensive assessment of the student's practical and research capabilities, preparing them for further research orprofessional practice.

IV SEMESTER

Sl. No	Course Code	Couse Type	Course Title	Hrs/Week	Credits	Examination		
						Marks		
				L : T : P: H		CIE	SEE	Total
1	P24MCA41	PEC	(Online Courses)12 weeks duration	-	3	-	-	100
2	P24MCA42	TS	Technical Seminar	-	2	100	-	100
3	P24MCA43	INT	Research Internship /Industry-Internship / StartupInternship	-	11	100	100	200
Total				**-**	**16**	**200**	**100**	**400**

Note: **PEC**-Professional Elective Courses, **TS** – Technical Seminar, **INT**- Industry/ Research Internship leading to the project work /startup

24MCA41: One Online NPTEL course can be cumbersome so one research article may be accepted/Published. A patent can be also one option.

TS: Technical Seminar: Students can present the seminar based on the new technologies in the seminar by all postgraduate students of the program shall be mandatory. The CIE marks awarded for the Seminar shall be based on the evaluation of the Report, Presentation skill, and performance in the Question-and-Answer session in the ratio 50:25:25. Seminar shall be considered as a head of passing and shall be considered for the award of degree. Those, who do not take up/ complete shall be declared as fail in the seminar course and have to complete the same during the subsequent semester

Industry Internship: The main objective of the industry internship is to ensure that the intern is exposed to a real-world environment and gain practical experience. Often, it may be a practical exposure to the theory that has been learned during the academic period. The industry internship helps students understand of analytical concepts and tools, hone their skills in real-life situations, and build confidence in applying the skills learned.

Research Internship: A research internship is an opportunity for students or early career professionals to gain hands-on experience in conducting research under the guidance of a mentor or within a research team. These internships can take place in academic institutions, research organizations, government agencies, or private companies

Research /Industry Internship: In the third-semester Students have to be in touch with a guide/mentor/coordinator and regularly submit the reportreferred to the progress internship. Based on the progress report the Guide/Mentor/coordinator has to enter the CIE marks. he/she has to attend the SEE at the parent Institute.

Assessment Details for I, II, III & IV Semester's Courses: (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 50% of the maximum marks. Minimum passing marks in SEE is 40% of the maximum marks of SEE. A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 50% (50 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Assessment Details for Integrated Professional Core Courses:

Assessment Details										
Assessment	**Weightage**	**Marks**	**Event Break UP**							
			Theory (A)				**Practical (B)**			
CIE	50%	50	**CIE I**	**CIE II**	**Assignment/ Skill Development**	**Average of CIE I & CIE II +Assignment**	**CIE**	**Record of Experiment write-up & Execution**	**CIE + Record Marks**	**Average of both A and B**
			40	40	10	50	40	10	50	50
SEE	50%	50	**Questions to Set: 5**		**Questions to Answer: 5**		**-**			

Assessment Details for Professional Core Courses & Basic Science Course:

Continuous Internal Evaluation (CIE):

Assessment Details						
Scheme	**Weightage**	**Marks**	**Event Break UP**			
CIE	50%	50	**CIE I**	**CIE II**	**Assignment**	**Average of CIE I & CIE II + Assignment**
			40	40	10	50
SEE	50%	50	**Questions to Set: 5**		**Questions to Answer: 5**	

Scheme of SEE Question Paper:

Scheme of SEE Question Paper (100 Marks)		
Duration:3hrs	**Marks:100**	**Weightage:50%**
• The question paper should cover the entire contents of the syllabus. • The question paper will be set for 100 marks and marks scored will be scaled down proportionately to 50 marks. • Total questions to be set are FIVE (with a maximum of 3 sub-questions), which have internal choice for any THREE units and remaining TWO units' questions are compulsory. • The students should answer 5 full questions • Each unit carries equal marks of 20.		

Assessment Details for Professional Core Laboratory Courses

Continuous Internal Evaluation (CIE):

Assessment Details						
Scheme	**Weightage**	**Marks**	**Event Break UP**			
CIE	**50%**	**50**	**CIE I**	**CIE II**	**Record of Experiment write-up & Execution**	**Average of CIE I & CIE II + Record**
			40	**40**	10	50
SEE	**50%**	**50**	write-up 15%, Conduction procedure and result in 70%, Viva voce 15% of maximum marks			

SEMESTER-I

Programming and Problem Solving in C			
Course Code	P24MCA11	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:2	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 10 hours Lab	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives (CLOs):

- Implement the constructs of C Language.
- Construct C Programs using basic programming constructs
- Develop C programs using arrays and strings
- Organize modular applications in C using functions
- Integrate pointers and structures in C applications and Execute input/output and file handlingin C

UNIT-I

08 Hours

BASICS OF C PROGRAMMING Introduction to programming paradigms – Applications of C Language - Structure of C program - C programming: Data Types - Constants – Enumeration Constants - Keywords – Operators: Precedence and Associativity - Expressions - Input/Output statements, Assignment statements – Decision making statements - Switch statement – Looping statements – Preprocessor directives - Compilation process

UNIT-II

08 Hours

ARRAYS AND STRINGS Introduction to Arrays: Declaration, Initialization – One dimensional array – Two dimensional arrays - String operations: length, compare, concatenate, copy – Selection sort, linear and binary search.

UNIT-III

08 Hours

FUNCTIONS AND POINTERS Modular programming - Function prototype, function definition, function call, Built-in functions (string functions, math functions) – Recursion, Binary Search using recursive functions –Pointers – Pointer operators – Pointer arithmetic – Arrays and pointers – Array of pointers – Parameter passing: Pass by value, Pass by reference.

UNIT-IV

08 Hours

STRUCTURAL AND UNION Structure - Nested structures – Pointer and Structures – Array of structures – Self-referential structures – Dynamic memory allocation - Singly linked list – typedef – Union - Storage classes and Visibility.

UNIT-V

08 Hours

FILE PROCESSING Files – Types of file processing: Sequential access, Random access –Sequential access file - Random access file - Command line arguments.

LAB EXPERIMENTS: **10 Hours**

1. Simulation of a Simple Calculator.
2. Implement Binary Search on Integers
3. Sort the given set of N numbers using Bubble sort.
4. Implement Matrix multiplication and validate the rules of multiplication.
5. An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges.
6. Write functions to implement string operations such as compare, concatenate, and find string length. Use the parameter passing techniques.
7. Implement structures to read, write and compute average- marks of the students, list the students scoring above and below the average marks for a class of N students.
8. Write a C program to copy a text file to another, read both the input file name and target file name.

TEXT BOOKS:

1. ReemaThareja, "Programming in C", Oxford University Press, Second Edition, 2016.
2. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015.

REFERENCES:

1. Paul Deitel and Harvey Deitel, "C How to Program with an Introduction to C++", Eighth edition, Pearson Education, 2018
2. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.
3. Byron S. Gottfried, "Schaum's Outline of Theory and Problems of Programming with C",McGraw-Hill Education, 1996.
4. Pradip Dey, Manas Ghosh, "Computer Fundamentals and Programming in C", Second 5. Edition, Oxford University Press, 2013.Anita Goel and Ajay Mittal, "Computer Fundamentals and Programming in C", 1st Edition,Pearson Education, 2013.
5. Anta Goel and Ajay Mittal, "Computer Fundamentals and Programming in C, 1st Edition Pearson Education, 2013.

Skill Development Activities Suggested

The students with the help of the course teacher can take up technical activities which will enhance their skill, or the students should interact with industry (small, medium and large), understand their problems or foresee what can be undertaken for study in the form of research/testing/projects, and for creative and innovative methods to solve the identified problem. The prepared report shall be evaluated for CIE marks.

Note: The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper shall include questions from the practical component.

Course Outcomes:
On completion of this course, the students should be able to:

CO1: Demonstrate knowledge on C Programming constructs (L5)
CO2: Develop simple applications in C using basic constraints (L4)
CO3: Design and implement applications using arrays and strings (L5)
CO4: Develop and implement modular applications in C using functions (L4)
CO5: Develop applications in C using structures and pointers (L4)

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1		-	-	-	-
CO2	-	-	3	2	-	-	-	-
CO3	-	-	3		-	-	-	-
CO4	-	-	-	2	-	-	-	-
CO5	-	-	-	-	3	-	-	-
3-Strong; 2-Medium; 1-Low								

Discrete Mathematics and Graph Theory			
Course Code	P24MCA12	CIE Marks	50
Teaching Hours/Week (L:P:T)	2:1:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Course Learning Objectives (CLOs)

- Analyze basic concepts of mathematical logic for analyzing propositions and proving theorems
- Apply sets and their operations algebraically to solve real-world problems.
- Examine the basics of graph theory and their various properties.
- Model problems using graphs and to solve these problems algorithmically.
- Apply graph theory concepts to solve real world problems in Computer Networking, Telecommunication Infrastructure, Transportation etc.

UNIT-I

08 Hours

Basic Structures: Sets, set operations, Principle of Inclusion, Exclusion and Pigeonhole principle, functions, matrices.

UNIT-II

08 Hours

The foundations of logic and proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference Introduction to Proofs

UNIT-III

08 Hours

Introduction to Graphs: Application of graphs – finite, infinite and bipartite graphs – Incidence and Degree – Isolated vertex, pendant vertex and Null graph. **Paths and circuits –** Isomorphism, sub-graphs, walks, paths and circuits, connected graphs, disconnected graphs and components.

UNIT-IV

08 Hours

Eulerian and Hamiltonian graphs: Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Travelling salesman problem. **Directed graphs** – types of digraphs, Digraphs and binary relation.

UNIT-V

08 Hours

Graph Colouring: Chromatic number, Chromatic polynomial, Matchings, Coverings, Four colour problem

Textbooks:

1. Kenneth H Rosen, "Discrete Mathematics and its Applications", McGraw Hill Publications, 6th edition
2. Narsingh Deo, Graph theory with the applications to engineering & Computer Science, Dovers Publications, 2016
3. J.A. Bondy and U.S.R. Murty. Graph theory with Applications, Springer, 1 st edition, 2008.

Reference Books:

1. J. K Sharma "Discrete Mathematics", Mac Millian Publishers India, 3rd edition, 2011
2. Garry Chartand and Ping Zhang, Introduction to Graph Theory, Tata McGraw-Hill, 2006.
3. Frank Harary, Graph Theory, Narosa Publishing House, Latest edition.
4. Discrete Mathematical Structures –by Dr. D.S. Chandrashekaraih.

Course Outcomes:

On completion of this course, the students should be able to:

CO1: **Understand** basic concepts of mathematical logic for analyzing propositions and proving theorems and Use sets and its operations algebraically for solving real world problems.

CO2: **Understand** the basics of graph theory and their various properties.

CO3: **Model** problems using graphs and to solve these problems algorithmically

CO4: **Analyze** Eulerian and Hamiltonian graphs, solve optimization problems like TSP, and explore directed graphs and binary relations.

CO5: **Derive and analyze** the chromatic polynomial for graphs, examining how colorings change with available Colors.

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	1	-	-	-	-	-
CO2	3	2	2	-	-	-	-	-
CO3	3	2	2	-	-	-	-	-
CO4	3	2	2	-	-	-	-	-
CO5	3	2	1	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Database Management Systems (DBMS)			
Course Code	P24MCA13	CIE Marks	50
Teaching Hours/Week (L:P: T)	4:0:0	SEE Marks	50
Total Hours of Pedagogy	50	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives (CLOs)

- Analyze the basic concepts and the applications of database systems.
- Evaluate the different issues involved in the design and implementation of Database System.
- Explain the basic concepts of relational data model, entity relationship model, relational database design, relational algebra and database language SQL .
- Design and build a simple database system and demonstrate competence with the fundamental tasks involved with modelling, designing, and implementing a DBMS
- Apply the Normalization concepts to normalize the database and describe the concepts of DBMS transactions, concurrency control and crash recovery.

UNIT-I

Introduction **10 Hours**

Introduction: Purpose of Database System, Views of data, data models, database management system, threeschema architecture of DBMS, components of DBMS. E/R Model Conceptual data modelling motivation, entities, entity types, attributes relationships, relationship types, E/R diagram notation, examples.

Data Models: Introduction to the Relational Model Structure Database Schema, Keys Schema Diagrams. Database design Other Models, ER diagrams ER Model Entities, Attributes and Entity sets Relationships and Relationship sets ER Design Issues Concept Design Conceptual Design with relevant Examples. Relational Query Languages, Relational Operations

UNIT-II

10 Hours

Relational Algebra-Selection and projection set operations renaming Joins Division Examples of Algebra overviews Relational calculus Tuple Relational Calculus (TRC) Domain relational calculus (DRC).

Overview of the SQL Query Language-Basic Structure of SQL Queries, Data types, Creating a database, create a table, drop the database, drop table, select table, insert a record, update record, delete a record, order by, group by, triggers,Set Operations, Aggregate Functions , Nested Sub queries, Views, Procedures.

UNIT-III

Relational Model and Relational Algebra **10 Hours**

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions and dealing with constraint violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping.

UNIT-IV

SQL **10 Hours**

SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL, Basic queries in SQL, more complex SQL Queries. Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL; Additional features of SQL;

Database programming issues and techniques; Embedded SQL, Dynamic SQL; Database stored procedures and SQL / PSM.

UNIT-V

10 Hours

Recovery and Atomicity- Log Based Recovery Recovery with Concurrent Transactions Check Points - Buffer Management Failure with loss of nonvolatile storage.

Text Books:

1. Database System Concepts, Silberschatz, Korth, Mc Graw hill, 7th edition.
2. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, TATA McGrawHill 3rd Edition.
3. Fundamentals of Database Systems, Elmasri and Navathe, 6th Edition, 2011, Pearson Education, ISBN-13: 9780136086208.

Reference Books:

1. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson, Eight Edition.
2. Rob,Coronel,"Database Systems", Seventh Edition, Cengage Learning
3. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL,Shah, PHI.

Course Outcomes:

On completion of this course, the students should be able to:

CO1: **Demonstrate** the basic elements of a relational database management system.
CO2: Ability to **identify** the data models for relevant problems
CO3: Design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respect data
CO4: Develop a database application using relation schema with the help of normalization
CO5: Utilize transaction processing, concurrency control and crash recovery.

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	-	-	-	-	-	-
CO2	3	1	-	-	-	-	-	-
CO3	3	2	1	1	-	-	-	-
CO4	1	1	-	-	-	-	-	-
CO5	1	1	1	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Operating System			
Course Code	P24MCA14	CIE Marks	50
Teaching Hours/Week (L:P: T)	2:1:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Course Learning Objectives (CLOs)

- Define operating systems and its concepts.
- Identify the services provided by the operating system and learning different operating structure.
- Understand process management and coordination with classic synchronization problems.
- Evaluate various CPU scheduling algorithm problems and analyze methods for handling deadlocks.
- Discuss different memory management techniques such as contiguous, paging, segmentation and virtualmemory.
- Explain file system and its implementation, mass storage and its scheduling

UNIT-I

08 Hours

Introduction to Operating Systems, System Structure What operating systems do, Operating System Operations, Computing Environments, Operating System Services, System Calls, Types of System Calls, System Programs, Operating System Structure, System Boot Process Concept Process Concept, Process Scheduling, Interprocess Communication

UNIT-II

08 Hours

Process Scheduling Basic Concepts, Scheduling Criteria, Scheduling Algorithms Synchronization Background, The Critical Section Problem, Mutex Locks, Semaphores, Classic Problems of Synchronization: Readers-Writers Problem, Dining Philosophers Problem using Semaphores

UNIT-III

08 Hours

Deadlocks: System model, Deadlock Characterization, Methods for handling deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock

UNIT-IV

08 Hours

Memory Management Strategies Basic Hardware, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory Management Background, Demand Paging, Page Replacement

UNIT-V

08 Hours

File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing,Protection, File-System Structure, Directory Implementation, Allocation Methods, Disk Scheduling.

Text Book:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating Systems Principles, 10th Edition, Wiley – India, 2019.

Reference Books:

1. D M Dhamdhere: Operating Systems – A Concept Based Approach, 3rd Edition, Tata McGraw – Hill, 2017.
2. Harvey M Deital: Operating Systems, 3rdEdition, Addison Wesley, 1990.

Course Outcomes:

On completion of this course, the students should be able to:

CO1: Explain The basics of Operating Systems, services and structure.
CO2: Understand Process concept, coordination and Synchronization, Methods for handling deadlocks.
CO3: Utilize Process Scheduling, Synchronization, deadlock avoidance, prevention and detection, diskscheduling, Memory management algorithms of operating system.
CO4: Identify Memory management techniques.
CO5: Discuss File concepts, allocation methods, Linux operating system.

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	1	-	1	-	-	-
CO2	3	2	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-
CO4	3	3	-	-	-	-	1	-
CO5	3	2	-	-	-	-	2	-
3-Strong;2-Medium;1-Low								

Web Technologies			
Course Code	P24MCA15	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Course Learning Objectives (CLOs)

- Creating the small web page using HTML and CSS
- Developing the interactive web pages using JavaScript
- Create web pages using angular JS framework

UNIT-I

Web browsers, web servers, MIME, URL, HTTP **08 Hours**
Introduction to HTML5 tags, Basic syntax and structure, text markups, images, lists, tables, progress, Media tags-audio and video, forms, span and div tags.

UNIT-II

08 Hours

Introduction to CSS, Levels of CSS, Selectors, Font, color and Text Properties, BOX Model, Introduction to JavaScript, JavaScript variables, operators, Conditional and loop statements in JavaScript, Functions and Arrays in JavaScript

UNIT-III

08 Hours

Event Handling and Document Object model in JavaScript, handling strings and working with window object

UNIT-IV

08 Hours

Introduction to AngularJS, Expressions, Modules, Directives, Model, Data binding, Controllers, Scopes, Filters

UNIT-V **08 Hours**

Services, Tables, Select box, Forms, Events, Validations

Text books:

1. Web Programming By Chris Bates , Wiley Publication
2. Robert W. Sebesta: Programming the World Wide Web, 8th Edition, Pearson education, 2015.
3. Angular JS By Krishna Rungta

Reference Books:

1. HTML5 Black Book by Dreamtech.

Web links:

- https://www.w3schools.com/angular/default.asp
- https://www.tutorialspoint.com/angular8/index.ht

Course Outcomes:
On completion of this course, the students should be able to:

CO1: Explain the fundamental concepts of web. (L2)
CO2: Construct the web pages using HTML and CSS. (L3)
CO3: Develop program using dynamic JavaScript for a given problem. (L3)
CO4: Build interactive web pages using AngularJS. (L3)
CO5: Develop the single window applications using AngularJS. (L3)

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	-	-	-	-	-
CO2	2	3	2	-	-	-	-	-
CO3	2	2	2	-	-	-	-	-
CO4	3	1	2	2	-	-	-	-
CO5	2	2	2	-	-	-	-	-
3-Strong;2-Medium;1-Low								

DBMS and Web Technologies Laboratory			
Course Code	P24MCAL16	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:0:2	SEE Marks	50
Total Hours of Pedagogy	10 Hours	Total Marks	100
Credits	01	Exam Hours	03

Course Learning Objectives (CLOs)

- Create SQL queries for the small projects.
- Create database objects that include tables, constraints, indexes, and sequences.
- Demonstrate client/server application for a given problem

1. Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries. BRANCH (Branchid, Branchname, HOD)
   STUDENT (USN, Name, Address, Branchid, sem)
   BOOK (Bookid, Bookname, Authorid, Publisher, Branchid)
   AUTHOR (Authorid, Authorname, Country, age)
   BORROW (USN, Bookid, Borrowed_Date)
   Execute the following Queries:
   i.List the details of Students who are all studying in 2nd sem MCA.
   ii.List the students who are not borrowed any books.
   iii.Display the USN, Student name, Branch_name, Book_name, Author_name, Books_Borrowed_Date of 2nd sem MCA Students who borrowed books.
   iv.Display the number of books written by each Author.
   v.Display the student details who borrowed more than two books.
   vi.Display the student details who borrowed books of more than one Author.
   vii.Display the Book names in descending order of their names.
   viii.List the details of students who borrowed the books which are all published by the same publisher.

2. Consider the following schema:
   STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA)
   Execute the following queries:
   i. Update the column total by adding the columns mark1, mark2, mark3.
   ii. Find the GPA score of all the students.
   iii. Find the students who born on a particular year of birth from the date_of_birth column.
   iv. List the students who are studying in a particular branch of study.
   v. Find the maximum GPA score of the student branch-wise.
   vi. Find the students whose name starts with the alphabet “S”.
   vii. Findthe students whose name ends with the alphabets “AR”.
   viii. Delete the student details whose USN is given as 1001

3. Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries .Consider a Cricket Tournament “ABC CUP” organized by an organization. In the tournament there are many teams are contesting each having a

Teamid,Team_Name, City, a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain. Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers, age. A player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name,Address (involves city,area_name,pincode). A team can play many matches. Each match played between the two teams in the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player.

Execute the following Queries:

i. Display the youngest player (in terms of age) Name, Team name, age in which he belongs of thetournament.
ii. List the details of the stadium where the maximum number of matches were played.
iii. List the details of the player who is not a captain but got the man_of _match award at least in twomatches.
iv. Display the Team details who won the maximum matches.
v. Display the team name where all its won matches played in the same stadium.

4. A country wants to conduct an election for the parliament. A country having many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state,Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter_id, having the Name, age, address (involves Houseno,city,state,pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidates are uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party.Thereare many parties. Each party is uniquely identified by using Party_id, having Party_Name,Party_symbol. A candidate can contest from many constituencies under a same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belongs to different parties. Each voter votes only one candidate of his/her constituencty.

   Queries:

   i. List the details of the candidates who are contesting from more than one constituency which are belongs to different states.
   ii. Display the state name having maximum number of constituencies.
   iii. Create a stored procedure to insert the tuple into the voter table by checking the voter age. If voter's age is at least 18-year-old, then insert the tuple into the voter else display the "Not an eligible voter msg"
   iv. Create a stored procedure to display the number_of_voters in the specified constituency. Where theconstituency name is passed as an argument to the stored procedure.
   v. Create a TRIGGER to UPDATE the count of "Number_of_Voters" of the respective constituency in Constiuency table , AFTER inserting a tuple into the "VOTERS"table.

5. Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places . Each Tourist place is identified by using tourist_place_id, having a name, belongs to a state, Number of kilometers away from the 02.03.2021 updated 52/ 104 capital city of that state,history. There are many Tourists visits tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple emailids. A tourist visits many Tourist places, it is also required to record the visted_date in the database. A

tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either in the same date or at different dates.

Queries:

i. List the state name which is having maximum number of tourist places.
ii. List details of Tourist place where maximum number of tourists visited.
iii. List the details of tourists visited all tourist places of the state "KARNATAKA"
iv. Display the details of the tourist place visited by the tourists of all country.e details of the tourists visited at least one tourist place of the state, but visited all states toursitplaces.
v. Display the details of the tourist place visited by the tourists of all country.

6. Create an XHTML page that provides information about your department. Your XHTML page must use the following tags: a) Text Formatting tags b) Horizontal rule c) heading d) Links e) Images f) Tables (Use of additional tags encouraged).

7. Develop and demonstrate a XHTML file that includes Javascript script for the following problems:
   a) Input : A number n obtained using prompt Output : The first n Fibonacci numbers
   b) Input : A number n obtained using prompt Output : A table of numbers from 1 to n and their squares using alert

8. Develop and demonstrate, using JavaScript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible. Modify the above document so that when a text is moved from the top stacking position, it returns to its original position rather than to the bottom

9. Develop, test and validate an XHTML document that includes text box, checkboxes and radio button along with submit and reset button.

10. Develop dynamic JavaScript code to create a html page to create online exam paper format with 4 optional answers.

Course Outcomes:
On completion of this course, the students should be able to:

CO1: Design entity-relationship diagrams to solve given database applications.
CO2: Exhibit SQL queries for data retrieval based on requirements.
CO3: Formulate SQL queries for the given problem
CO4: Construct the web pages using HTML and CSS
CO5: Develop program using dynamic JavaScript for a given problem

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	-	-	-	-
CO2	3	2	2	1	-	-	-	-
CO3	2	1	1	1	-	-	-	-
CO4	3	1	2	2	-	-	-	-
CO5	2	2	2	-	-	-	-	-
3-Strong;2-Medium;1-Low								

Employability Enhancement Skills -1			
Course Code	P24MHU19	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:2:0	SEE Marks	-
Total Hours of Pedagogy	30 Hours	Total Marks	100
Credits	0	Exam Hours	03

Course Learning Objectives (CLOs)

- Introduction to basic quantitative aptitude concepts and problem-solving techniques.
- Advanced quantitative aptitude topics and practice.
- Enhance reasoning and analytical skills through verbal reasoning exercises.
- Develop effective communication skills and understanding of professional etiquettes.
- Focus on personal branding and future career planning.

UNIT - I

06 hours

Introduction to basic quantitative aptitude concepts and problem-solving techniques.

Agenda Points:

1. Introduction to Quantitative Aptitude
    - Definition and importance
    - Topics overview (numbers, geometry, algebra)
2. Number Systems
    - Types of numbers, prime numbers, factors
    - Activities: Solving basic number system problems
3. Arithmetic
    - Percentages, ratios, and proportions
    - Activities: Real-life application problems

Hands-On Practice:

- Group exercises on problem-solving on the above concepts

Expected Outcomes:

- Understanding of basic numerical concepts
  Ability to solve arithmetic problems accurately

UNIT - II

6 hours

Advanced quantitative aptitude topics and practice.

Agenda Points:

1. Algebra
    - Basic equations and inequalities
    - Activities: Solving equations as a team
2. Geometry
    - Basic shapes, area, and volume calculations
    - Activities: Geometry problem sets
3. Data Interpretation
    - Understanding graphs and tables
    - Activities: Analyzing data sets for insights

Hands-On Practice:
- Problem-solving competition
- Group discussions on different approaches to problems

Expected Outcomes:
- Enhanced skills in solving algebraic and geometric problems
- Ability to interpret and analyze data

Deliverables:
- Problem-solving competition results and worksheets
- Aptitude Assessment – 45 min – google form based MCQ's

UNIT - III

06 hours

Enhance reasoning and analytical skills through verbal reasoning exercises.

Agenda Points:
1. Introduction to Verbal Reasoning
    - Importance in exams and interviews
    - Types of verbal reasoning problems
2. Practice Exercises
    - Analogies, syllogisms, and critical reasoning
    - Activities: Pair work for solving reasoning puzzles

Hands-On Practice:
- Group discussions on reasoning strategies
- 45 minutes - Mock verbal reasoning tests – Google form based

Expected Outcomes:
- Improved analytical thinking and reasoning capabilities

Deliverables:
- Results from mock tests and group discussions

UNIT – IV

07 hours

Develop effective communication skills and understanding of professional etiquettes.

Agenda Points:
1. English Communication Basics
    - Importance of effective communication
    - Key elements of verbal and non-verbal communication
2. Written Communication Etiquettes
    - Email etiquette and formal writing skills
    - Activities: Drafting professional emails
3. Professional Etiquettes
    - Understanding workplace behavior
    - Activities: Role-playing scenarios

Hands-On Practice:
- Practicing written communication through peer feedback
- Role-playing to demonstrate professional etiquettes

Expected Outcomes:

- Ability to communicate effectively in English
- Understanding of professional behavior in workplace settings

Deliverables:

- Drafted emails and feedback from peers
- Students will be able to write an effective email to the department about various topics related to their career
- Students will be able to write an effective email to HR department seeking a job opportunity

UNIT - V

07 hours

English Communication and Professional Skills

Agenda Points:

1. SWOT Analysis
    - Understanding personal strengths, weaknesses, opportunities, and threats
    - Activities: Group SWOT analysis discussions
2. Goal Setting
    - Setting SMART goals for personal and professional growth
    - Activities: Individual goal-setting exercises
3. Resume Building and LinkedIn Profile Creation
    - Key components of impactful resumes
    - Tips for optimizing LinkedIn profiles
    - Activities: Drafting a resume and LinkedIn profile

Hands-On Practice:

- Group sharing of personal goals and SWOT analysis
- Peer reviews of resumes and LinkedIn profiles

Expected Outcomes:

- Clarity on personal and professional goals
- Comprehensive resume and LinkedIn profile ready for use

Deliverables:

- Completed SWOT analysis, personal goals, resume, and LinkedIn profile drafts

Reference Books:

1. "Word Power Made Easy New Revised and Expanded Edit ion", First Edition, Norman Lewis, GoyalPublisher.
2. "Essential English Grammar", Raymond Murphy, Cam bridge University Press, new edition

Course Outcomes:
On completion of this course, the students should be able to:

CO1: Demonstrate understanding of basic numerical concepts by accurately solving arithmetic problems.
CO2: Apply advanced skills to solve algebraic and geometric problems effectively.
CO3: Develop analytical thinking and reasoning capabilities to evaluate complex problems and construct logical solutions.
CO4: Demonstrate the ability to communicate effectively in English through written, verbal, and non-verbal methods
CO5: Create a comprehensive resume and LinkedIn profile tailored for professional opportunities

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	-	-	-	2	-	3
CO2	-	3	-	-	3	-	-	2
CO3	-	3	-	-	3	-	-	1
CO4	-	-	2	-	3	2	-	2
CO5	-	-	2	2	3	-	2	2
3-Strong;2-Medium;1-Low								

Mathematics Bridge Course for MCA Program			
Course Code	P24MCA19	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	-
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	0	Exam Hours	

Course Learning Objectives (CLOs)

The mandatory learning course, Mathematics Bridge Course for MCA aims to provide basic concepts of Sets, Relations, Logic, Matrices & Determinants, Sequences & Series and Probability Theory.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

- The lecture method (L) need not be only the traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
- Use of Video/Animation to explain the functioning of various concepts.
- Encourage collaborative (Group Learning) Learning in the class.
- Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking.
- Adopt Problem-Based Learning (PBL), which fosters students' Analytical skills, and develops design thinking skills such as the ability to design, evaluate, generalize, and analyse information rather than simply recall it.
- Show the different ways to solve the same problem and encourage the students to come up with creative ways to solve them.
- Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

UNIT - I

Set Theory **08 hours**

Introduction. Representation of sets, Types of Sets, Finite set, Infinite set, equivalent set, disjoint set, Subset, Power set. Venn diagram. Set operations: Union, Intersection, Complement of a set, Difference, Symmetric Difference. Laws of set theory. Cartesian product of sets, Relations and properties.

UNIT - II

Logic and Propositions **08 hours**

Logic Statement, Propositions, Connectives, Basic Logic Operations: Conjunction, Disjunction, Negation, Implication and Double Implication. Truth table, Logical Equivalence/Equivalent Statements, Tautologies and Contradictions**.**

UNIT - III

Matrices and Determinants **08 hours**

Matrix Introduction, Types of matrices, Scalar multiplication, Addition of matrices, Product of matrices. Transpose of a matrix, Symmetric and Skew Symmetric matrix, Rank of a matrix, Determinant of a matrix. Singular matrix.

UNIT - IV

Sequence and Series **08 hours**

Introduction, Sequences, Series, Arithmetic Progression, Sum of Finite number of terms in A.P, Arithmetic Means, Geometric Progression, sum to n terms of G.P, Geometric Mean, relation between A.M and G.M.

UNIT – V

Probability Theory **08 hours**

Probability: Introduction, random experiments, sample space, events and algebra of events. Definitions of Probability – classical, and axiomatic. Conditional Probability, laws of addition and multiplication, independent events, theorem of total probability, Bayes' theorem and its applications.

Assessment Details (CIE)

The weightage of Continuous Internal Evaluation (CIE) is 100%. The minimum passing mark for the CIE is 40% of the maximum mark (100). A student shall be deemed to have satisfied the academic requirements if the student secures not less than 40% (40 Marks out of 100) in the CIE.

Reference Books:

1. "Kenneth H Rosen, "Discrete Mathematics and its Applications", McGraw Hill publications, 7th edition.
2. Walpole Myers Ye "Probability and Statistics for engineers and Scientist" Pearson Education, 8th edition.
3. Richard A Johnson and C. B Gupta "Probability and statistics for engineers" Pearson Education.
4. J.K Sharma "Discrete Mathematics", Mac Millan Publishers India, 3rd edition,2011.
5. Higher Engineering Mathematics, B. S. Grewal, Khanna Publishers, 43rd Edition, 2015.

Web links and Video Lectures (e-Resources):

- http://.ac.in/courses.php?disciplineID=111
- http://www.class-central.com/subject/math(MOOCs)
- http://academicearth.org/
- VTU EDUSAT PROGRAMME-20

Course Outcomes:

On completion of this course, the students should be able to:

CO1: **Widen** the knowledge of Basic concepts in Set Theory and Apply the fundamentals of set theory and Relations to the given problem

CO2: **Understand** mathematical reasoning to read, comprehend and construct mathematical arguments

CO3: **Understand** the Basic Concepts in Matrices and Formulate the problems in Matrix expression.

CO4: **Determine** the sum of the first n terms of an arithmetic and Geometric series

CO5: Get the **basic** concepts of probability and find the probability of simple and compound events

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	1	-	-	-	-	-
CO2	3	2	2	-	-	-	-	-
CO3	3	2	2	-	-	-	-	-
CO4	3	2	2	-	-	-	-	-
CO5	3	2	1	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

SEMESTER – II

Machine learning and Data analytics using Python			
Course Code	P24MCA21	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:2	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 10 hours Lab	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives (CLOs)

- Understand foundational concepts in machine learning and data analytics.
- Gain proficiency in Python for data analysis and machine learning tasks.
- Learn and apply various machine learning algorithms and techniques.
- Develop skills in data preprocessing, visualization, and model evaluation.
- Prepare students for industry roles involving data-driven decision making and predictive modeling.

UNIT – I

Introduction to Machine Learning and Python: 8 Hours

Introduction to Machine Learning: Definition and importance of machine learning, Types of machine learning: Supervised, unsupervised, and reinforcement learning, Applications of machine learning in various domains.

Python for Data Analysis: Introduction to Python programming, Python libraries for data analysis: NumPy, Pandas, Matplotlib, Data manipulation and visualization using Pandas and Matplotlib.

Data Preprocessing: Data cleaning and transformation, Handling missing values and outliers, Feature scaling and normalization.

UNIT – II

Supervised Learning: 8 Hours

Regression: Linear regression, Polynomial regression, Model evaluation metrics: MAE, MSE, RMSE. Classification: Logistic regression, K-Nearest Neighbors (KNN), Decision Trees and Random Forests, Model evaluation metrics: Accuracy, precision, recall, F1-score, ROC-AUC.

Model Training and Evaluation: Train-test split and cross-validation, Hyper parameter tuning using GridSearchCV, Overfitting and underfitting.

UNIT – III

Unsupervised Learning: 8 Hours

Clustering: K-Means clustering, Hierarchical clustering, Evaluation of clustering results. Dimensionality Reduction: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) t-Distributed Stochastic Neighbor Embedding (t-SNE).

Association Rule Learning: Apriori algorithm, Market Basket Analysis, Evaluation metrics for association rules

UNIT – IV

Advanced Machine Learning Techniques: 8 Hours

Ensemble Methods: Bagging and Boosting, Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoost).

Support Vector Machines (SVM): Linear and non-linear SVM, Kernel trick, Model evaluation and tuning.
Neural Networks and Deep Learning: Introduction to neural networks, Building and training neural networks using TensorFlow and Keras, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

UNIT – V

Data Analytics and Real-World Applications: 8 Hours

Exploratory Data Analysis (EDA): Data visualization techniques, Statistical analysis and hypothesis testing, Identifying patterns and insights from data.
Time Series Analysis: Introduction to time series data, Time series forecasting using ARIMA and Prophet, Evaluating time series models.
Integrating Machine Learning Models: Deployment of machine learning models, Building web applications with Flask and Django, Case studies on real-world applications of machine learning.

LAB EXPERIMENTS

10 Hours

1. Python programs to show the usage of Python Libraries for ML application such as Pandas, Matplotlib and Seaborn. Read the training data from a .CSV file.
2. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
3. Write a program to implement the binary logistic Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
4. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions.
5. Write a program to implement SVM algorithm to classify the iris data set. Print both correct and wrong predictions.
6. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
7. Write a program to demonstrate Regression analysis with residual plots on a given data set.
8. Write a program to compute summary statistics such as mean, median, mode, standard deviation and variance of the given different types of data.
9. Write a program to implement k-Means clustering algorithm to cluster the set of data stored in .CSV file.

Text Books:

1. Practical Machine Learning for Data Analysis Using Python Abdulhamit Subasi.
2. Advance Machine Learning with Python by John Hearty.

Reference Books:

1. Machine Learning using Python, Manaranjan Pradhan, U Dinesh Kumar, Wiley India Pvt. Ltd., 2019(Chapters: 1, 2, 4, 5, 6.3, 6.5, 7)
2. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf, 2013.
3. Learning with Python: How to Think Like a Computer Scientist Paperback – Allen Downey , Jeffrey Elkner, 2015.

4. Jake Vander plas, “Python Data Science Handbook: Essential tools for working with data”, O‘Reilly Publishers, I Edition.
5. Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems, Aurelien Geron, O'Reilly Publisher , I edition, 2017

Web links and Video Lectures (e-Resources):
https://youtu.be/7eh4d6sabA0?si=JWHNJRVS6NhQOPYe
https://youtu.be/kqtD5dpn9C8?si=LBep4HWaMFRrAPsD
https://youtu.be/4SJ7bEILPJk?si=5LurvjzUOuCew1W9

Skill Development Activities Suggested
The students with the help of the course teacher can take up technical activities which will enhance their skill or the students should interact with industry (small, medium and large), understand their problems or foresee what can be undertaken for study in the form of research/testing/projects, and for creative and innovative methods to solve the identified problem. The prepared report shall be evaluated for CIE marks.

Course outcome:
After completion of this course the student will be able to

CO1: Understanding of machine learning concepts and techniques. L2
CO2: Perform data preprocessing and exploratory data analysis using Python. L4
CO3: Develop and evaluate learning models using Python libraries. L5
CO4: Apply learning model machine for real-world data problems. L3
CO5: Integrate data analytics and machine learning models into practical applications. L5

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	-	-	-	-	-	2
CO2	-	2	3	-	-	-	-	-
CO3	-	-	3	2	-	-	-	-
CO4	-	-	3	2	-	-	-	2
CO5	-	-	3	2	-	-	-	2
3-Strong;2-Medium;1-Low								

Object Oriented Programming using Java			
Course Code	P24MCA22	CIE Marks	50
Teaching Hours/Week (L:P:T)	4:0:0	SEE Marks	50
Total Hours of Pedagogy	50	Total Marks	100
Credits	4	Exam Hours	3

Course Learning Objectives (CLOs)

- This subject will help to improve the analytical skills of object-oriented programming Formal introduction to Java programming language
- Overall development of problem solving and critical analysis

UNIT - I

10 hours

The History and Evolution of Java: The Byte code, Features of Java An overview of Java: Object Oriented Programming, Structure of a Java program, Data Types and Variables, Type conversion and casting, Arrays

Classes: Fundamentals, Declaring Objects, Assigning Object Reference Variables, Methods, Constructors, this Keyword, Garbage Collection, Stack application

Methods and Classes: Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Access Control, static, final, Command-Line Arguments

UNIT - II

10 hours

Inheritance: Basic concepts, Member Access and Inheritance, Practical Example Inheritance types, super, constructors, Method Overriding, Dynamic Method Dispatch, Abstract Classes, final with inheritance.

String Handling: String Constructor, String length, Special string Operations, CharacterExtraction, String comparison, Modifying a string, String Buffer

Generics: About Generics, A simple Generic Example, General class with Two Type Parameters, General form of generic class

UNIT - III

10 hours

Packages and Interfaces: Packages, Packages and member access, importing packages, Interfaces, Default interface methods, Use static methods in an interface, Private Interface methods.

Exception handling: Fundamentals, Exception types, uncaught exceptions, try and catch, multiple catch clauses, nested try statements, throw, throws, finally, Java's built-in exceptions, User-defined exceptions.

UNIT - IV

10 hours

Multithreaded Programming: Java thread model, main thread, creating thread, creating multiple

threads, isalive( ) and Join( ), thread priorities, synchronization

Input/Output: Exploring java.io - The I/O Classes and Interfaces, The Byte Streams

UNIT - IV

10 hours

Event Handling: Two Event Handling Mechanisms, The Delegation Event Model, EventsEvent Sources, Event Listeners, Event Classes- The MouseEventClass, Event Listener Interfaces-The MouseListener Interface, the MouseMotionListener Interface, Delegation

Event Model – Handling Mouse Events.

AWT: Working with Windows, Graphics and Text

AWT Classes, Window Fundamentals, Working with Frame Windows, Graphic

Text Book::

1. Java the Complete Reference Eleventh Edition by Herbert Schildt ,TataMcGraw-hill Edition, 2019

Reference Text Books:

1. Introduction to JAVA Programming 9th Edition by Y. Daniel Liang , Pearson education, 2012
2. Programming in JAVA 5.0 1st Edition by James P Cohoon, Jack W Davidson, TATA McGraw hill,2006

Web links and Video Lectures (e-Resources):

1) https://ia800303.us. archive.org/ 26/items/ JavaJavaJavaObjectorientedProblemSolving/ jjj-os.pdf
2) http://people.reed.edu/ ~jerry/121/materials/ artsciencejava.pdf
3) https://upload.wikimedia. org/wikipedia/ commons/e/e7/ Java_Programming.pdf
4) https://onlinecourses.swayam2.ac.in/aic20_sp1 3/preview
5) https://onlinecourses.swayam2.ac.in/aic20_sp1
6) https://www.classcentral.com/course/coursera-object-oriented-programming-in-java-4212

Skill Development Activities Suggested

The students with the help of the course teacher can take up technical activities which will enhance their skill or the students should interact with industry (small, medium and large), understand their problems or foresee what can be undertaken for study in the form of research/testing/projects, and for creative and innovative methods to solve the identified problem. The prepared report shall be evaluated for CIE marks.

Course Outcomes:

On completion of this course, the students should be able to:

CO1: Understand the basic principles of the object-oriented programming
CO2: Demonstrate an introductory understanding of graphical user interfaces, multi-threaded programming.
CO3: Apply the knowledge of Java concepts to find the solution for a given problem.
CO4: Analyze the given Java application for correctness/functionalities.
CO5: Develop Java programs / applications for a given requirement.

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	-	3	-	-	-	-	
CO2	2	-	-	-	-	-	-	3

CO3	2	-	-	3	-	-	-	-
CO4	-	2	3	-	-	-	-	-
CO5	-	2	-	3	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Data Structure and Algorithms			
Course Code	P22MCA23	CIE Marks	50
Teaching Hours/Week (L:P:T)	4:0:0	SEE Marks	50
Total Hours of Pedagogy	50 hours	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives (CLOs)

- To understand and implement fundamental data structures.
- To develop efficient algorithms for solving problems.
- To analyze the time and space complexity of algorithms.
- To gain practical experience in applying data structures and algorithms to real-world problems.
- To prepare students for industry roles requiring strong foundations in data structures and algorithmic thinking.

UNIT-I

Introduction to Data Structures and Algorithms: 10 Hours

Basic Concepts: Definition and importance of data structures, Abstract Data Types (ADTs),
Algorithm analysis: Time and space complexity, Big O notation. Arrays: Definition and operations: Insertion, deletion, traversal, Multidimensional arrays, Applications of arrays.
Linked Lists: Singly linked list: Creation, insertion, deletion, traversal, Doubly linked list and circular linked list, Applications of linked lists.

UNIT-II

Stacks, Queues, and Recursion: 10 Hours

Stacks: Definition and operations: Push, pop, peek, Applications: Expression evaluation, backtracking, function calls. Queues: Definition and operations: Enqueue, dequeue, front, rear, Types: Circular queue, priority queue, double-ended queue (deque), Applications of queues. Recursion: Definition and principles of recursion, Recursive algorithms: Factorial, Fibonacci series, Tower of Hanoi, Analysis of recursive algorithms.

UNIT-III

Trees and Graphs: 10 Hours

Trees: Definition and terminology: Root, leaf, internal node, height, depth, Binary trees: Traversal (preorder, inorder, postorder), creation, insertion, deletion, Binary search trees (BST), AVL trees, B-trees. Graphs: Definition and terminology: Vertices, edges, adjacency list, adjacency matrix, Graph traversal algorithms: Depth-first search (DFS), breadth-first search (BFS),Shortest path algorithms;Dijkstra's algorithm,Floyd-Warshall algorithm.

UNIT-IV

Sorting and Searching Algorithms: 10 Hours

Sorting Algorithms: Basic concepts and classification, Comparison-based sorting: Bubble sort, selection sort, insertion sort, quicksort, mergesort, heapsort, non-comparison-based sorting: Radix sort, counting sort. Searching Algorithms: Linear search and binary search, Search in linked lists, trees, and graphs, Hashing: Hash functions, collision resolution techniques (chaining, open addressing).

UNIT-V

Advanced Data Structures and Applications: 10 Hours

Advanced Data Structures: Heaps: Definition, operations, heap sort, applications, Trie: Definition, operations, applications in dictionary and spell-checking, Segment trees and Fenwick trees: Definition, operations, range queries. Algorithm Design Techniques: Divide and conquer, Greedy algorithms, Dynamic programming. Industry Applications: Realworld applications of data structures and algorithms, Best practices in data structure and algorithm implementation, Case studies of complex problem-solving using advanced data structures.

Text Books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
3. Anany Levitin: Introduction to the Design and Analysis of Algorithms, 2nd edition, Pearson Education,2003.

Reference Books:

1. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
2. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
3. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2 nd Ed, McGraw Hill, 2013
4. A M Tenenbaum, Data Structures using C, PHI, 1989
5. Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.
6. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009, Pearson.
7. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
8. Algorithms, Kenneth A Berman and Jerome L Paul, Cengage Learning India Pvt Ltd, 2002 Edition

Web links:

- https://www.youtube.com/watch?v=BBpAmxU_NQo
- https://www.youtube.com/watch?v=8hly31xKli0
- https://archive.nptel.ac.in/courses/106/106/106106127/

Course Outcomes:

On completion of this course, the students should be able to

CO1: Demonstrate an understanding of fundamental data structures and algorithms. (L2)
CO2: Implement and manipulate data structures such as arrays, linked lists, stacks, queues, trees, and graphs. (L3)
CO3: Develop algorithms for searching, sorting, and optimization problems. (L3)
CO4: Analyze the efficiency and correctness of algorithms. (L4)
CO5: Apply data structures and algorithms to solve complex problems in various domains. (L3)

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	-	-	-	-	-
CO2	2	3	2	-	-	-	-	-

CO3	2	2	2	-	-	-	-	-
CO4	3	2	2	2	-	-	-	-
CO5	2	2	2	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Software Engineering			
Course Code	P24MCA24	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:1:0	SEE Marks	50
Total Hours of Pedagogy	50	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives (CLOs)

- Understand the importance of various Software Engineering Lifecycle Models.
- Document the Software Requirements Specification (SRS) for the identified system.
- Gain knowledge of the System Analysis and Design concepts using UML.

UNIT-I

SOFTWARE PROCESS AND AGILE DEVELOPMENT: ` **10 Hours**

Introduction to Software Engineering, SoftwareProcess, Perspective and Specialized Process Models Introduction to Agility-Agile process-Extremeprogramming-XP Process-Case Study.

UNIT-II

REQUIREMENTS ANALYSIS AND SPECIFICATION: 10 Hours

Requirement analysis and specification, Requirements gathering and analysis, Software Requirement SpecificationFormal , system specification ,Finite State Machines, Object modelling using UML, Use case Model, Class diagrams, Interaction diagrams, Activity diagrams, State chart diagrams,Functional modelling Data Flow Diagram- CASE TOOLS.

UNIT-III

SOFTWARE DESIGN 10 Hours

Software design Design process Design concepts Coupling Cohesion Functional Independence Design patterns Model-view-controller Publish-subscribe Adapter Command Strategy Observer Proxy Facade Architectural stylesLayered - Client Server -Tiered - Pipe and filter- User interface

UNIT-IV

TESTING AND MAINTENANCE 10 Hours

Testing Unit testing Black box testing White box testing Integration and System testing Regression testing Debugging - Program analysis Symbolic execution Model Checking Case Study.

UNIT-V

PROJECT MANAGEMENT 10 Hours

Software Project Management- Software Configuration Management - Project Scheduling- DevOps: Motivation-Cloud as a platform-Operations- Deployment Pipeline:Overall Architecture Building and Testing-Deployment- Tools- Case Study.

Text Books:

1. Bernd Bruegge and Allen H. Dutoit, "Object-Oriented Software Engineering: Using UML, Patterns and Java", Third Edition, Pearson Education, 2009
2. David C Kung, "Object-Oriented Software Engineering: An Agile Unified Methodology", First Edition, Mc Graw-Hill International Edition, 2014.
3. Len Bass, Ingo Weber and Liming Zhu, "DevOps: A Software Architect's Perspective", Pearson Education, 2016.
4. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Learning Private Limited, 2009
5. Pankaj Jalote, "Software Engineering, A Precise Approach", Wiley India, 2010.

Reference Books:

1. Pankaj Jalote, Software Engineering Wiley India Pvt. Ltd. (2010)
2. Roger S Pressman: Software Engineering-A Practitioners approach, 6th edition, McGraw-Hill, 2010.
3. Hans Van Vliet: Software Engineering Principles and Practices, 3rd Edition, Wiley-India, 2010

Course Outcomes:

On completion of this course, the students should be able to

CO1: Describe the fundamental software process models and and the principles of requirement analysis.
CO2: Analyze Software Requirement Specification and different system models
CO3: Implement design patterns and architectural styles to construct software systems.
CO4: Utilize the software testing methodologies to ensure the quality and reliability of software.
CO5: Analyze the integration of software project management practices with DevOps principles to improve software delivery and operational efficiency

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	-	-	-	-	-	-
CO2	2	3	1	1	-	-	-	-
CO3	3	2	1	2	-	-	-	-
CO4	2	3	1	2	-	-	-	2
CO5	2	2	1	-	1	1	-	2
3-Strong; 2-Medium; 1-Low								

Web Application Development			
Course Code	P24MCA25	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:1:0	SEE Marks	50
Total Hours of Pedagogy	50	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives (CLOs)

- To understand the fundamental concepts and technologies of web application development.
- To gain proficiency in front-end and back-end web development.
- To learn and apply modern web frameworks and libraries.
- To develop skills in creating responsive and dynamic web applications.
- To Implement front-end and back-end technologies to expertise in web development.

UNIT-I

Introduction to Web Development and HTML5: 10 Hours

Web Development Basics: Introduction to web technologies and protocols, Client-server architecture, Overview of front-end and back-end development

HTML5 Fundamentals: HTML5 elements and attributes, Semantic HTML5 tags, Forms and input types, Multimedia elements (audio, video)

Advanced HTML5: Canvas and SVG for graphics, HTML5 APIs(Geolocation, Web Storage, Web Workers), Offline web applications using AppCache.

UNIT-II

CSS3 and Responsive Web Design: 10 Hours

CSS3 Basics: Introduction to CSS3, Selectors, properties, and values, Box model, layout, and positioning, Flexbox and Grid layouts.

Responsive Web Design: Media queries, Responsive design principles, Fluid grids and flexible images, Mobile-first design approach.

CSS Frameworks: Introduction to Bootstrap, Bootstrap components and utilities, Customizing Bootstrap with Sass.

UNIT-III

JavaScript and DOM Manipulation: 10 Hours

JavaScript Basics: Introduction to JavaScript, Variables, data types, and operators, Control structures (if-else, loops), Functions and scope

Document Object Model (DOM): DOM structure and manipulation, Event handling and event listeners, Creating and modifying DOM elements, Form validation using JavaScript

Advanced JavaScript: Asynchronous JavaScript (callbacks, promises, async/await), AJAX and Fetch API, Introduction to JavaScript libraries (e.g., jQuery).

UNIT-IV

Front-End Frameworks and AngularJS: 10 Hours

Introduction to Front-End Frameworks: Importance of front-end frameworks, Overview of popular frameworks (React, Angular, Vue)
Advanced AngularJS: Services and dependency injection, Routing and single-page applications (SPAs), Data binding and form handling, Custom directives and components

UNIT-V

Back-End Integration and Deployment: **10 Hours**

Back-End Development: Introduction to server-side programming, Overview of server-side languages (Node.js, PHP, Python), RESTful web services and APIs, Database integration (SQL, NoSQL)
Full-Stack Development: Integrating front-end and back-end technologies, Developing fullstack web applications, Case studies on full-stack applications
Deployment and Security: Web application deployment (cloud platforms, hosting services), Security best practices for web applications, Authentication and authorization, Performance optimization.

Text Books:

1. Web Programming By Chris Bates , Wiley Publications
2. Robert W. Sebesta: Programming the World Wide Web, 8th Edition, Pearson education, 2015.
2. HTML5 Black Book by Dreamtech
3. Angular JS By Krishna Rungta
4. Bootstrap essentials by Snig by Packt-open source .

Reference Books:

1. Deitel, P.J. Deitel, A. B. Goldberg: Internet & World Wide Web How to program, 3rd Edition, Pearson Education / PHI, 2004.

Reference links: https://www.tutorialspoint.com/angularjs
https://www.tutorialspoint.com/angularjs/angularjs_tutorial.pdf

Course outcomes
On completion of this course, the students should be able to:

CO1: Outline the understanding of web technologies and protocols. (L2)
CO2: Develop web applications using HTML, CSS, JavaScript, and modern frameworks. (L3)
CO3: Apply responsive design principles using frameworks like Bootstrap.(L3)
CO4: Construct dynamic web applications using AngularJS. (L3)
CO5: Develop front-end and back-end technologies to create full stack web applications. (L3)

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	-	-	-	-	-
CO2	2	3	2	-	-	-	-	-
CO3	2	2	2	-	-	-	-	-
CO4	3	1	2	2	-	-	-	-

CO5	2	2	2	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Object Oriented Programming using JAVA Laboratory			
Course Code	P24MCAL26	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:0:2	SEE Marks	50
Total Hours of Pedagogy	10 Hours	Total Marks	100
Credits	1	Exam Hours	03

Course Learning Objectives (CLOs)

- Using java programming to develop programs for solving real-world problems.
- Reinforce the understanding of basic object-oriented programming concepts.

Experiments

1. Write a Java program to print the following triangle of numbers

   1 2

   1 2 3

   1 2 3 4

   1 2 3 4 5

2. Write a Java program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)

3. Write a Java program

   - To find the area and circumference of the circle by accepting the radius from the user.
   - To accept a number and find whether the number is Prime or not

4. Write a Java program to demonstrate a division by zero exception

5. Write a Java program to implement Inner class and demonstrate its Access protection.

6. Write a Java program to demonstrate Constructor Overloading and Method Overloading.

7. Write a JAVA program to demonstrate Inheritance. Simple Program on Java for the implementation of Multiple inheritance using interfaces to calculate the area of a rectangle and triangle.

8. Write a Java applet program, which handles keyboard event.

9. Write a Java Program to create a window when we press

- ✓ M or m the window displays Good Morning
- ✓ A or a the window displays Good After Noon
- ✓ E or e the window displays Good Evening
- ✓ N or n the window displays Good Night

10. Write a Java program to implement a Queue using user defined Exception Handling (also make use of throw, throws). a. Complete the following: b. Create a package named shape. c. Create some classes in the package representing some common shapes like Square, Triangle, and Circle. d. Import and compile these classes in other program.

Course Outcomes:
On completion of this course, the students should be able to:

CO1: Demonstrate the fundamental data types and constructs of Java Programming by writing executable/interpretable programs.
CO2: Illustrate the object-oriented principles with the help of java programs.
CO3: Develop reusable and efficient applications using inheritance concepts of java.
CO4: Learn the object-oriented concepts and its implementation in Java.

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	3	-	-	-	-	-
CO2	-		-	-	-	-	-	3
CO3	-	2	-	3	-	-	-	-
CO4	-	-	3	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Data Structures with Algorithms Laboratory			
Course Code	P24MCAL27	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:0:2	SEE Marks	50
Total Hours of Pedagogy	10 Hours	Total Marks	100
Credits	1	Exam Hours	03

Course Learning Objectives (CLOs)

- Evaluate the Expressions like postfix, prefix conversions.
- Implementing various data structures like Stacks, Queues, Linked Lists, Trees and Graphs.

LAB EXPERIMENTS

1. Implement a Program in C for converting an Infix Expression to Postfix Expression.
2. Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide)
3. Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display
4. Write a C program to simulate the working of a singly linked list providing the following operations: a. Display& Insert b. Delete from the beginning/end c. Delete a given element
5. Write a C program to Implement the following searching techniques a. Linear Search b. Binary Search.
6. Write a C program to implement the following sorting algorithms using user defined functions: a. Bubble sort (Ascending order) b. Selection sort (Descending order).
7. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm (C programming)
8. From a given vertex in a weighted connected graph, find shortest paths to other vertices Using Dijkstra's algorithm (C programming)
9. Using circular representation for a polynomial, design, develop, and execute a program in C to accept two polynomials, add them, and then print the resulting polynomial.
10. Check whether a given graph is connected or not using DFS method.

Course Outcomes:

On completion of this course, the students should be able to:

CO1: Implement the techniques for evaluating the given expression.(L3)
CO2: Design sorting / searching techniques and validate input/output for the given problem. (L3)
CO3: Apply data structures (namely Stacks, Queues, Circular Queues, Linked Lists, and Trees), its operations and algorithms. (L3)
CO4: Apply data structures(namely Linked Lists, and Trees),its operations and algorithms. (L3)
CO5: Implement the algorithm to find whether the given graphics connected or not and conclude on the performance of the technique implemented. (L2)

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	-	-	-	-	-
CO2	2	3	2	-	-	-	-	-
CO3	2	2	2	-	-	-	-	-
CO4	3	1	2	2	-	-	-	-
CO5	2	2	2	-	-	-	-	-
3-Strong; 2-Medium; 1-Low								

Employability Enhancement Skills-2 (EES-2)			
Course Code	P24MHU28	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:2:0	SEE Marks	-
Total Hours of Pedagogy	40 Hours	Total Marks	100
Credits	0	Exam Hours	03

Course Learning Objectives (CLOs)

- Strengthen foundational knowledge in Data Structures and Algorithms (DSA) using C.
- Enhance problem-solving skills through Python and Object-Oriented Programming (OOPs).
- Develop core skills in Database Management Systems (DBMS) with a focus on SQL and NoSQL concepts.
- Integrate programming knowledge to build and manage practical applications.

UNIT–I

07 Hours

Introduction to Data Structures and Algorithms (DSA): Basic Concepts: Definition, importance, and real-world applications of data structures. **Data Structures Overview**: Arrays: Basics and operations (insertion, deletion), Introduction to pointers in C.

Hands-On Practice: Implementing basic array operations in Introduction to git repository and online coding platforms (LeetCode, HackerRank).

Expected Outcomes:

- Clear understanding of basic data structures.
- Ability to implement array operations and manage version control.

UNIT–II

07 Hours

Intermediate Data Structures and Problem-Solving

Linked Lists: Singly linked lists: Creation, traversal, and deletion, Comparison with arrays.

Stacks: Basic operations (push, pop) and applications, **Problem-Solving Activities**: Coding exercises on linked list traversal and stack operations.

Expected Outcomes:

- Proficiency in working with linked lists.
- Understanding of stack operations and their practical applications.

UNIT–III

Object-Oriented Programming (OOPs) Using Python **06 Hours**

Python Overview: Syntax, data types, and control structures, **OOPs Principles**: Classes, objects,

encapsulation, and inheritance.

Hands-On Practice: Creating classes and implementing inheritance, Developing modular and reusable code structures.

Expected Outcomes:

- In-depth understanding of OOPs principles and Python programming.

UNIT–IV

6 Hours

Database Management Systems (DBMS) : Core Concepts: Overview of data models and relational databases, Basics of SQL: DML, DQL, DLL.

Expected Outcomes:

- Fundamental knowledge of relational and unstructured databases.
- Ability to write SQL queries for data management.

UNIT–V

CRUD operations: Core operations for interacting with databases—Create, Read, Update, and Delete.
Unstructured Databases: Introduction to NoSQL databases, Schema-less design for flexibility, Horizontal scalability to handle large data volumes.
Optimized for high-velocity data processing

Capstone Project: Developing a CRUD-based application using Python and SQL.
Exploring structured and unstructured data integration, **Assessment and Evaluation**: Written test and program coverage report.

Expected Outcomes:

- Hands-on experience in integrating programming and database skills.
- Improved logical thinking and practical application development.

Reference Books:

1. "Word Power Made Easy New Revised and Expanded Edition", First Edition, Norman Lewis, GoyalPublisher.
2. "Essential English Grammar" by Raymond Murphy, published by Cambridge University Press.

Course outcomes:
On completion of this course, the students should be able to:

CO1: Implement array operations and utilize version control systems effectively to manage code
CO2: Demonstrate proficiency in implementing and manipulating linked lists to solve Computational problems.
CO3: Analyze and apply Object-Oriented Programming (OOP) principles in Python to design and develop efficient software solutions
CO4: Design and execute SQL queries for effective data management and retrieval.
CO5: Demonstrate improved logical thinking to develop and implement practical applications Effectively.

.

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	-	-	-	1
CO2	3	3	3	3	-	-	2	1
CO3	2	3	3	3	-	-	2	1
CO4	3	3	3	3	3	2	2	2
CO5	3	3	3	3	3	2	2	2
3-Strong; 2-Medium; 1-Low								

Ability Enhancement Courses with Seminar-I			
Course Code	P24MCA29	CIE Marks	-
Teaching Hours/Week (L:P:T)	-	SEE Marks	-
Total Hours of Pedagogy	-	Total Marks	-
Credits	0	Exam Hours	-

Guidelines:

- A team of TWO students must develop the project.
- The team may implement a project of their choice based on emerging trends.
- The team must present their work in front of the Internal panel members and also submit a Brief Report (25 to30Pages) that must include the following:
  - Introduction
  - Requirements
  - Software Development Process Model Adopted
  - Analysis and Design Models
  - Implementation
  - Testing
  - References

Course outcomes:
On completion of this course, the students should be able to:

CO1: **Identify**, discuss and justify the technical aspects of the chosen project with a comprehensive and systematic approach.
CO2:**Analyse** and design the project.
CO3:**Apply** acquired knowledge for project development using modern tools and technology.
CO4:**Apply** debugging tools for testing the project
CO5:**Demonstrate** the project

Mapping with Program Outcomes:

CO's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	2	3	-	2	-
CO2	3	2	3	2	-	-	-	-
CO3	3	3	3	3	-	-	-	-
CO4	3	1	3	2	3	-	-	3

CO5	-	-	-	2	3	-	-	2
3-Strong; 2-Medium; 1-Low								