COURSE BOOK MCA I YEAR

AUTONOMOUS

KIET GROUP OF INSTITUTIONS

Connecting Life with Learning

www.kiet.edu
Delhi-NCR, Ghaziabad

CELEBRATING 25 YEARS OF EXCELLENCE

CURRICULUM STRUCTURE & SYLLABUS

Effective from the Session: 2024-25

1. Teaching Scheme of (MCA I Year) Department wise

1.1 Master of Computer Applications

MCA, 1st Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (Core)	MCA	ID101B	Programming for Computer Application	L	2	0	0	40	10	50	50	100	2
2	PC	Major (Core)	CSE	CS206L	Operating System	L	3	0	0	60	15	75	75	150	3
3	BS	SEC	ASH	MA205L	Mathematical Foundation of Data Science	L	3	0	0	60	15	75	75	150	3
4	PC	Major (Core)	CSE	CS207B	Software Engineering	L	3	0	0	60	15	75	75	150	3
5	PW	Value Added Courses	CSIT	ID103B	Design Thinking	L	1	0	0	40	10	50	—	50	1
6	HS	Value Added Courses	ASH	HS107L	Quantitative Aptitude & Logical Reasoning-I	L	2	0	0	40	10	50	50	100	2
Lab/Practical															
7	PC	Major (Core)	MCA	ID101P	Programming for Computer Application Lab	P	0	0	6	—	75	75	75	150	3
8	PC	Major (Core)	CSE	CS206P	Operating System Lab	P	0	0	2	—	25	25	25	50	1
9	ES	Minor Stream	EEE	EE101P	IoT and Embedded Systems Lab	P	0	0	2	—	25	25	25	50	1
10	PW	Value Added Courses	CSIT	ID104B	Innovation and Entrepreneurship	B	0	0	2	—	50	50	—	50	1
11	PW	Value Added Courses	MCA	ID102B	Mini Project-1	B	0	0	4	—	50	50	50	100	2
12	HS	Value Added Courses	ASH	HS101B	Communication Skills	B	0	0	4	80	20	100	—	100	2
Total Hours : 34 hrs.							**14**	**0**	**20**					**1200**	**24**

MCA, 2nd Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (Core)	MCA	ID202L	Software Testing	L	3	0	0	60	15	75	75	150	3
2	PC	Major (Core)	CSE	CS204B	Cloud Computing	L	2	0	0	40	10	50	50	100	2
3	HS	Value Added Courses	ASH	HS201L	Quantitative Aptitude & Logical Reasoning-II	L	2	0	0	40	10	50	50	100	2
Blended															
4	PC	Major (Core)	CSE	CS202B	Data Structures and Algorithms	B	2	0	4	80	20	100	100	200	4
5	PC	Major (Core)	CSE	CS203B	Database Management Systems	B	2	0	4	80	20	100	100	200	4
6	PC	SEC	CSE(AIML)	AI101B	Introduction to AI	B	2	0	2	60	15	75	75	150	3
7	PW	SEC	MCA	ID201B	Mini Project-2 (Full Stack Development)	B	2	0	6	100	25	125	125	250	5
Lab/Practical															
8	HS	Value Added Courses	ASH	HS202B	Campus to Corporate#	P	0	0	2	40	10	50	—	50	1
Total Hours : 33 hrs.							**15**	**0**	**18**					**1200**	**24**

2. Theory Courses Detail Syllabus

Course Code: ID101B	Course Name: Programming for Computer Application	L	T	P	C
		2	0	0	2

Pre-requisite:

- Basics of computers.
- Knowledge of any C compiler (not mandatory).

Course Objectives:

- Teach students the specific syntax and semantics of the C programming language, including how to write, compile, and debug C programs.
- Equip students with the skills to analyze problems and design algorithms.
- Introduce fundamental programming concepts such as variables, data types, operators, control structures (loops and conditionals), and functions.
- Provide a deep understanding of pointers, arrays and structures.
- Introduce essential data structures such as stacks, queues, and their implementations in C.

Course Outcome:

Students will be able to:	
CO1	Apply the concept of flow chart and branching.
CO2	Use the concept of looping structures and functions for a given problem.
CO3	Explore solution for the problems based on multi-dimensional arrays and pointers.
CO4	Demonstrate the operations using strings and structures.
CO5	Illustrate linear data structures like stacks and queues.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	-	3	-	-	-	3
CO2	3	3	-	1	-	-	-	3
CO3	3	3	-	1	-	-	-	2
CO4	3	3	-	1	-	-	-	2
CO5	3	3	-	1	-	-	-	3

Detailed Syllabus

Unit 1	**Introduction**	**6 hours**

Basics of programming: Approaches to problem solving, Concept of algorithm and flowchart.
Introduction to C: Structure of C program, compiling C program, link and run of C program, character set, tokens, keywords, identifiers, constants, variables, data types.
Operators and expressions: Operators, precedence, associativity and type conversion.
Conditional Program Execution: if, if-else, and nested if-else statements, switch statements, use of break and default with switch, comparison of switch and if-else.

Unit 2	**Lopping Structures and Functions**	**6 hours**

Loops and Iteration: for, while and do-while loops, nested loops, break and continue statement.
Functions: Introduction, types, declaration of a function, function calls, defining functions, function prototypes, passing arguments to a function, return values and their types, calling function by value.

Unit 3	**Arrays and Pointers**	**6 hours**

Arrays: Array notation and representation, declaring one-dimensional array, initializing arrays, accessing array elements, manipulating array elements, two-dimensional arrays, multidimensional arrays.

Pointers: Introduction, characteristics, * and & operators, pointer type declaration and assignment, call by reference, passing pointers to functions, array of pointers, pointers to functions, pointer to pointer.

Unit 4	Strings and Structures	6 hours

Strings: Introduction, initializing strings, accessing string elements, array of strings, passing strings to functions, string functions.
Structures: Introduction, initializing, defining, and declaring structure, accessing members, operations on individual members, operations on structures, structure within structure, array of structure, pointers to structure.

Unit 5	Linear Data Structure	6 hours

Stack: Introduction of stack, Memory Representation, Implementation of Stacks using arrays.
Applications of Stacks: Evaluation of postfix expression, infix to postfix.
Recursion: Recursion and its types, solving and storage representation of factorial of a number, nth Fibonacci number, Tower of Hanoi.
Queue: Introduction of Queue, Memory Representation Implementation of queue, deques, priority queues and circular queues using arrays.

	Total Lecture Hours	30 hours

Textbook:

1. Kanetkar Y., “Let Us C”, BPB Publications, 15th edition, 2016.
2. Schildt H., “C - The Complete Reference”, McGraw-Hill, 4th edition, 2017.

Reference Books:

1. Hanly J. R. and Koffman E. B., “Problem Solving and Program Design in C”, Pearson Education, 7th edition, 2013.
2. Gottfried B., “Schaum’s Outlines- Programming in C”, McGraw-Hill Publications, 2nd edition, 1996.
3. Kochan S.G., “Programming in C”, Addison-Wesley, 4th edition, 2014.
4. Greg Perry, "C Programming Absolute Beginner’s Guide", 3rd edition, Que Publishing, 2014.
5. Kernighan Brian W. "C Programming Language", 2nd edition, Pearson, 2015.
6. E Balagurusamy, Computer Concepts and Programming in C, 3rd edition, McGraw Hill Education (India) Private Limited, 2015.

Mode of Evaluation:

Evaluation Scheme

MSE@		CA					ESE	Total Marks
MSE1	MSE2	CA1	CA2	CA3	CA4 (Attendance)	CA5	50	100
20	20	2	2	3	3	3		
40		Best of 4 (10)						

@ Online exam on Code Tantra
CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: CS206L	Course Name: Operating System	L	T	P	C
		3	0	0	3

Pre-requisite: Concepts of Computer Fundamentals.

Course Objectives:

- To introduce basic concepts and functions of modern operating systems.
- To understand the concept of process, and process scheduling.
- To understand the concepts of process synchronization and dead lock management.
- To implement memory management algorithms.
- To implement I/O Management and Disk Scheduling.

Course Outcome:

Students will be able to:	
CO1	Discuss the role of operating systems and their types.
CO2	Apply the concept of CPU scheduling algorithms for execution of programs.
CO3	Apply the concept of inter-process communication, process synchronization and deadlock handling.
CO4	Discover memory management techniques.
CO5	Illustrate the concept of I/O management and file system.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	1	-	-	-	-	-	3
CO2	3	3	-	-	-	-	-	2
CO3	3	3	-	-	-	-	-	2
CO4	3	3	-	-	-	-	-	2
CO5	3	2	-	-	-	-	-	2

Detailed Syllabus

Unit 1	Introduction	9 hours

Operating System Overview: Operating System Objectives and Functions, Evolution of Operating Systems, Virtual Machines, Booting Process of Operating Systems.

Unit 2	Process Management and CPU Scheduling	9 hours

Process: Introduction to Process, Process States, Process Description Block, Process Control (Process creation, waiting for the process/processes, Loading programs into memory and Process Termination).

CPU Scheduling: Scheduling Criteria, Types of Scheduling, Scheduling Algorithms.

Unit 3	Process Synchronization and Deadlock	9 hours

Concurrency: Process Synchronization and Mutual Exclusion Principles of Concurrency, Requirements for Mutual Exclusion, Critical Section Problem, Dekker's Solution, Peterson's Solution, Semaphores, Hardware Support, Operating System Support (Semaphores and Mutex).

Classical Synchronization Problems: Inter Process Communication models, Readers/Writers Problem, Producer and Consumer problem.

Deadlocks: Introduction to Deadlock, Principles of Deadlock, Starvation, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery.

Unit 4	Memory Management	9 hours

Memory Partitioning: Fixed Partitioning, Fragmentations, Compaction, Dynamic Partitioning, Buddy System, Relocation, Paging, Segmentation.

Virtual Memory: Demand Paging, Page Replacement Algorithm, Thrashing, Cache Memory Organization, Locality of Reference, Cache mapping techniques.

Unit 5	Input/Output and Files	9 hours

I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, I/O Buffering, Disk Scheduling.

File Management: File Organization and Access, Access Matrix, File Directories, File Sharing, Record Blocking, Secondary Storage Management.

Total Lecture Hours	45 hours

Textbook:

3. *Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", John Wiley & Sons, Inc., 9th Edition, 2012.*
4. *William Stallings, "Operating System: Internals and Design Principles", Prentice Hall, 8th Edition, 2014.*

Reference Books:

7. Sumitabh Das, "UNIX: Concepts and Applications", McGraw Hill, 4th Edition, 2017
8. Tom Adelstein and Bill Lubanovic, "Linux System Administration", O'Reilly Media, Inc., 1st Edition, 2007.
9. Harvey M. Deitel, "Operating Systems", Prentice Hall, 3rd Edition, 2003.
10. Meeta Gandhi, Rajiv Shah, Tilak Shety, Vijay Mukhi, "The C Odyssey: Windows", BPB Publication, 6th edition, 2004
11. Yashwant Kanitkar, Unix Shell Programming, BPB Publications, 1st Edition, 2003.

Mode of Evaluation:

Evaluation Scheme								
MSE		CA					ESE	Total Marks
MSE1	MSE2	CA1	CA2	CA3	CA4 (Attendance)	CA5	75	150
30	30	4	4	4	3	4		
60		Best of 4 (15)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: MA205L	**Course Name:** Mathematical Foundation of Data Science	**L**	**T**	**P**	**C**
		3	0	0	3

Pre-requisite: Linear Algebra, Probability and statistics.

Course Objectives:

- Equip students with a strong mathematical groundwork essential for AI, machine learning, and data science.
- Able to solve real-world problems by using statistics, probability, Random number generation with data pre-processing.

Course Outcome:

Students will be able to:	
CO1	Summarize statistical analysis and its types to perform various operations.
CO2	Analyze the probability distribution on a variety of data.
CO3	Apply statistical analysis and perform appropriate statistical tests.
CO4	Analyze linear method for regression analysis.
CO5	Analyze various techniques for data pre-processing and random number generation

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	1	1	-	-	-	1
CO2	3	2	2	1	-	-	-	2
CO3	3	2	2	1	-	-	-	2
CO4	3	2	2	1	-	-	-	2
CO5	3	2	1	1	-	-	-	1

Detailed Syllabus

Unit 1	**Data Science and Descriptive Statistics**	**9 hours**

Introduction to Data Science: Overview of data science, importance of mathematics in data science.
Introduction to Statistics: Types of statistical analysis: Descriptive statistics, types of statistical analysis- inferential statistics, relation between statistics and machine learning, understanding the types of data, sampling techniques.
Hands on: Types of data
Descriptive Statistics- Measure of central tendency, Measure of Dispersion- range & interquartile range, measures of skewness and kurtosis.
Hands on: BigMart Sales Analysis

Unit 2	**Probability**	**9 hours**

Probability: Basic probability concepts, measures of probability, conditional probability, product rule of dependent and independent event, Bayes' theorem.
Hands on: Bayes' theorem
Discrete and continuous probability distribution, expectation and variance, central limit theorem.
Hands on: Central Limit Theorem.

Unit 3	**Inferential Statistics**	**9 hours**

Inferential statistics: Introduction, sampling & confidence interval, margin of error, hypothesis testing & its steps. Z-test, t-test, ANOVA test & Chi-squared test.
Hands on: Hypothesis Testing & Steps of Hypothesis testing- ZTest/T Test
Hands on: ANOVA and chi-square ARIMA and SARIMA.
Correlation, inference procedure for correlation coefficient, bivariate correlation, multiple correlations, linear regression with normal equation.
Hands on: PUBG Game Analysis/ BigMart Sales Analysis.

Unit 4	**Linear Methods for Regression Analysis**	**9 hours**

Regression: Overview of Regression Analysis, Simple Linear Regression, Multiple linear regression, Dimensionality Reduction Techniques-Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).
Hands on: PCA

Unit 5	**Pseudo-Random Number Generation**	**9 hours**

Pseudo-Random Number Generation: Introduction, Bivariate and Multivariate.
Hands on: Random Number generation, bi- variate and multi-variate.
Data Pre-processing: Overview, data cleaning: missing values, noisy data, data cleaning as a process.

Data Integration: Redundancy and correlation analysis, data value conflict detection and resolution, data reduction, transformation and discretization.
Hand on: Data Pre-processing.

	Total Lecture Hours	**45 hours**

Textbook:

1. Sheldon M. Ross, “Introduction to Probability and Statistics for Engineers and Scientists”, Academic Press, 5th edition
2. S.C. Gupta & V.K. Kapoor, “Fundamentals of Mathematical Statistics”, Sultan Chand & Sons
3. Han, J., Kamber, M., & Mining, D. (2012). Concepts and techniques. Morgan kaufmann, 340, 94104-3205.

References:

1. https://www.udemy.com/course/mathematical-foundation-for-machine-learning-and-ai/
1. https://www.udemy.com/course/master-linear-algebra-and-probability-2-in-1-bundle/

Mode of Evaluation:
Continuous Assessment (CA): Video Assignments/ Assignment / Quiz / Project / Seminar/ MOOC Course etc.

Evaluation Scheme								
MSE		**CA#**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5***	75	150
30	30	4	4	4	3	4		
60		15						

*The marks will be given if the student is participating in any Hackathon, National International research paper or any other National/International Activity.
#Best 4 out of 5 will be considered.

Course Code: CS207B	**Course Name:** Software Engineering	**L**	**T**	**P**	**C**
		3	0	0	3

Pre-requisite: None

Course Objectives:

- Introduce basic software engineering concepts for software development process.
- Analyze the system requirement specifications.
- Design the system using modeling tools.
- Explore the basic concept of software testing and debugging.
- Understand the various project management activities.

Course Outcome:

Students will be able to:	
CO1	Understand the fundamental concepts of the software development process.
CO2	Apply the concept of requirement engineering in the SRS document.
CO3	Demonstrate the concepts of software design.
CO4	Elaborate concept of software maintenance and software project management.
CO5	Illustrate the fundamental concepts of Agile models.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	2	-	-	-	-	-	-	1
CO2	2	3	2	-	1	-	-	3
CO3	3	3	3	3	2	-	-	3
CO4	3	2	1	1	1	-	-	3
CO5	2	1	1	-	1	3	-	2

Detailed Syllabus

Unit 1	**Introduction**	**9 hours**

Introduction to Software Engineering: Software Components, Software Characteristics, Conventional Vs Software Engineering approach.

Software Development Life Cycle (SDLC) Models: Waterfall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Rapid Application Development. **Quality Assurance**: Software Quality Assurance Models, SEI-CMM Model, ISO Models.		
Unit 2	**Software Requirement Specifications**	**9 hours**
Software Requirements: Functional and Non-Functional Requirements, User Requirements, System Requirements, and Interface Requirements. **Requirement Engineering Process:** Feasibility Study, Requirement Elicitation, Requirement Analysis, Requirement Management, Introduction to Software Requirements Specifications (SRS), Structure of SRS (IEEE Format).		
Unit 3	**Software Design & Testing**	**9 hours**
Introduction to Design: Basic Concepts, Top-down & Bottom-Up Approach, Coupling, Cohesion, Structure Charts. **Introduction to Software Testing:** Software Errors, Faults and Failures Concepts of Verification and Validation. **Levels of Testing:** Unit Testing, Integration Testing, System Testing, Acceptance Testing. White Box Testing: Test coverage, code coverage, condition coverage, branch coverage, cyclomatic complexity; Black-Box Testing: Equivalence classes, boundary value tests, Testing for Non-Functional Requirements Load, Volume, Performance, Code Inspection, Test Plan, Test case, Test suite.		
Unit 4	**Software Project Management**	**9 hours**
Maintenance: Introduction, Perfective, Preventive, Corrective, Adaptive. **Software Project Management Activities:** Project planning and Tracking, Cost Estimation Management (Constructive Cost Model (COCOMO)), Scheduling Management (GANTT and CPM/PERT), **Configuration Management Activities**: Identification and Establishment, Version Control, Change Control, Configuration Auditing.		
Unit 5	**Agile Process Model**	**9 hours**
Agile Model: Introduction to Agile, principles of agile manifesto, overview of Various Agile methodologies - Scrum, Extreme Programing, Lean, and Kanban **Key aspects of Scrum**: roles - Product Owner, Scrum Master, Team, Manager in scrum and product backlog **Scrum process flow**: Product Backlog, Sprint Review, sprints backlog, scrum meetings, sprint review, sprint retrospective, Burn down chart. **Scaled agile frameworks:** SAFe, Scrum@Scale, Disciplined Agile.		
	Total Lecture Hours	**45 hours**

Textbook:

4. Roger S. Pressman, Software Engineering-A practitioner's Approach, 7th Edition, McGraw-Hill International, 2010.
5. K.K Aggarwal & Yogesh Singh, Software Engineering, 3rd Edition, New Age International Publishers, 2007.
6. Pankaj Jalote, "Software Engineering", 1st edition, Wiley, 2013.
7. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, 1st edition, Pearson Education, 2009.

Reference Books:

1. Sommerville, Software Engineering, 8th Edition, Pearson Education, 2009.
2. Aditya P Mathur, Fundamentals of Software Testing; 1st Edition, Pearson, 2011.

Mode of Evaluation:

Evaluation Scheme								
MSE		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**		
30	30	4	4	4	3	4	75	150
60		Best of 4 (15)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: ID103B	**Course Name:** Design Thinking	**L**	**T**	**P**	**C**
		1	0	0	1

Pre-requisite: Not Applicable

Course Objectives:

- To expose the student with state-of-the-art perspectives, ideas, concepts, and solutions related to the design and execution of projects using design thinking principles.
- To prepare the mindset and discipline of systemic inspiration driven by a desire to identify new sources of ideas, and new models especially outside their regular working atmosphere.
- To propose a concrete, feasible, viable and relevant innovation project/challenge.

Course Outcome:

Students will be able to:	
CO1	Understand the basic requirements of a good design.
CO2	Empathies and ideate the solutions to problems in his environment
CO3	Prototype and test the developed solutions.
CO4	Apply the principles of design thinking on developing innovative solutions to the real-world problems.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

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

Detailed Syllabus

Unit 1	**Fundamentals of Design Thinking**	**4 hours**

Concept of Design Thinking, Need of Design Thinking, Goal of Design thinking (Desirability, feasibility and viability), Design thinking Process model, Design thinking tools.

Activities: Identify an Opportunity, Scope of the Project, Explore the possibilities and prepare a design brief.

Unit 2	**Empathize and Define**	**4 hours**

Design thinking phases, how to empathize, Role of empathy in design thinking, the purpose of empathy maps, Things to be done prior to empathy mapping, Activities during and after the session, Understanding empathy tools: Customer Journey Map, Personas. Define-Methods of Define Phase: Storytelling.

Activities: Apply the methods of empathizing and Define Phases Finalize the problem statement.

Unit 3	**Ideation**	**4 hours**

Challenges in idea generation, Visualize, Empathize, and Ideate method, Importance of visualizing and empathizing before ideating, Applying the method, Create Thinking, Generating Design Ideas, Lateral Thinking, Analogies, Brainstorming, Mind mapping, Ideation Tools: How Might We? (HMW), Storyboard, Brainstorming. What is design innovation? A mindset for innovation, and asking "What if?" asking "What wows?" and "What works?"

Activities: Apply the methods of Ideate Phase: Generate Innovative solution ideas.

Unit 4	**Prototyping and Testing**	**3 hours**

What is a prototype? - Prototyping as a mindset, prototype examples, prototyping for products; Why we prototype? Fidelity for prototypes, Process of prototyping- Minimum Viable prototype. Testing prototypes with users, collect feedback; iterate and improve the ideas.

Activities:

1. Prototype: Apply the Methods of the Prototype Phase - Create prototypes for selected ideas.
2. Testing: Collect feedback; iterate and improve the ideas Present your solution using the Storytelling method.

Total Lecture Hours	**15 hours**

Reference Books:

1. E. Balagurusamy and Bindu Vijayakumar, Design Thinking (A Beginner's perspective), McGraw-Hill, 2024.
2. Michael Lewrick, Patrick Link, and Larry Leifer, The Design Thinking Playbook, Wiley, 2018.
3. Christian Müller-Roterberg, Design Thinking for Dummies, Wiley, 2021.
4. Don Norman, The Design of Everyday Things, Navol Books Trading, Edition 2022
5. James Robert Rossman and Mathew D. Duerden, Designing Experiences, Columbia Business School Pub, 2019.
6. Roger Martin, "The Design of Business:Why Design Thinking is Next CompetitiveAdvantage", Harvard Business Press, 2009.
7. Idris Mootee, Design Thinking for Strategic Innovation,2013, John Wiley & Sons Inc., 2014.

Additional Learning Resources:

1. https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process
2. https://www.ibm.com/design/thinking/page/toolkit
3. https://www.interaction-design.org/literature/article/define-and-frame-your-designchallenge-by-creating-your-point-of-view-and-ask-how-might-we
4. https://nptel.ac.in/courses/109/104/109104109/

5. https://nptel.ac.in/courses/110106124/

Mode of Evaluation:

Evaluation Scheme							
MSE	**CA**					**ESE**	**Total Marks**
	CA1	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	NA	50
40	2	2	3	3	3		
40	Best of 4 (10)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: HS107L	**Course Name:** Quantitative Aptitude & Logical Reasoning-I	**L**	**T**	**P**	**C**
		2	0	0	2

Course Objectives:

- Students will gain a strong foundation in basic mathematical concepts, and become proficient in performing calculations, solving equations, and manipulating numerical data.
- Students will learn to think critically, make deductions, and draw logical conclusions based on given information. This will enable them to approach complex problems systematically and derive accurate solutions.
- Students will learn how to interpret and analyze data presented in various formats such as tables, graphs, and charts. This skill is particularly useful for tasks involving statistical analysis and decision-making.
- It will help students to improve their speed and accuracy in solving problems. Through practice and exposure to different types of questions, students will learn time-saving techniques and develop strategies to effectively manage time during exams or real-world scenarios.
- It will help students to prepare for various competitive exams, such as aptitude tests for job placements, college admissions tests for higher education (GATE/CAT/GRE/GMAT), and government entrance exams. The course will provide the necessary knowledge and practice to perform well in these exams.

Course Outcome:

Students will be able to:	
CO1	Analyze problems, identify relevant information, and apply appropriate mathematical methods to reach solutions.
CO2	Analyze tasks and activities by following a chain of thought process and find logical solutions to a problem.
CO3	Analyze trends, patterns, and relationships within the data.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	-	-	-	-	-	-
CO2	2	2	-	-	-	-	-	-
CO3	1	1	-	-	-	-	-	-

Detailed Syllabus

Unit 1	**Speed Math & Number System**	**6 hours**

Addition, Subtraction, Multiplication and Division shortcuts, Square and Square roots, Cubes and Cube roots, Comparing fractions, Classification of Numbers, Divisibility Rules, Cyclicity and finding Unit digit, Remainder Theorem, Factorization, Finding HCF and LCM, Word problem based on HCF and LCM, Arithmetic and Geometric Progression, Averages, Surds & Indices, Simplification.

Unit 2	**Series, Coding and Decoding, Ranking and Order**	**6 hours**

Number and Letter Series, Number and Letter Analogy, Coding & Decoding, (Type-I)-: Basics of A-Z and Z-A, the position of alphabets including Reverse Order, the concept of 27, Short-cuts to learn positions of alphabets, (Type-II)-: Ascending and Descending calculation of position of alphabets, (Type-III)-: Fictitious Coding, (Type-IV)-: Substitution Method applied in Coding, (Type-V)-: finding position and identifying the Characters in Coded words, Ranking and Order based problems.

Unit 3	**Percentage, Profit, Loss & Discount (PPLD), Interest**	**6 hours**

Introduction & Definition of Percentage, Conversion of a fraction to percentage and vice versa, Growth and depreciation, Word problem based on percentage, Application of percentage in Change in Area, Perimeter, and Volume of different Geometrical shapes, Definition and Introduction of P & L, Application of P & L based on percentage, Introduction of discount, Discount Series, single discount,

Application based word problems on Discount, Introduction of Concept related to SI and CI, Generalized way to find the difference between SI and CI for 2 year and 3 year.		
Unit 4	**Ratio, Proportion and Variation, Mixture and Alligation**	**6 hours**
Definition of Ratio and Proportion, Type of Proportion (Direct Proportion, Inverse Proportion, Continued Proportion), Definition of Variance, Problem-related to age, coins, expenditure, and saving, etc, Introduction of Mixture and Alligation, Repletion process, Application on profit and loss, Time, Speed and Distance, Percentage used in Mixture and allegation.		
Unit 5	**Direction Sense, Data Arrangement, and Data Interpretation**	**6 hours**
Direction Sense based Problems, Linear Arrangement, Circular Arrangement, Multi-dimensional Arrangement, Floor based puzzles, Introduction to Data Interpretation (DI), Questions based on Tabular charts, Bar Graphs, Pie charts, Line Graphs, and Mix graphs etc., Questions based on missing data.		
	Total Lecture Hours	**30 Hours**

Textbooks:
1. “Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publication.
2. “A Modern Approach to Verbal & Non-Verbal Reasoning” by R.S. Aggarwal, S. Chand Publication.

Reference Books:
1. Arun Sharma, How to Prepare for Quantitative Aptitude for the CAT, 10th Edition, TMH Publication, 2022.
2. Arun Sharma, How to Prepare for Logical Reasoning for the CAT, 7th Edition, TMH Publication, 2024.
3. Arun Sharma, "How to Prepare for Data Interpretation for the CAT, 8th Edition, TMH Publication, 2024.

Mode of Evaluation:

Evaluation Scheme								
MSE		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	50	100
20	20	2	2	3	3	3		
40		Best of 4 (10)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Lab Course Code: ID101P	**Lab Course Name:** Programming for Computer Application Lab	**L**	**T**	**P**	**C**
		0	0	6	3

Pre-requisite: Concepts of Computer Fundamentals.

Course Objectives:
- Read, understand and trace the execution of programs written in C language.
- Write the C code for a given algorithm.
- Implement programs with arrays, pointers, strings and structures.
- Write programs that perform operations on data structures like stacks and queues.

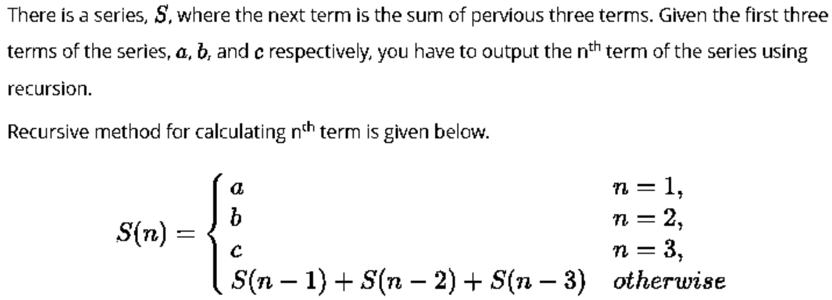
Course Outcome:

Students will be able to:	
CO1	Apply the concept of flow chart and branching.
CO2	Use the concept of looping structures and functions for a given problem.
CO3	Explore solution for the problems based on multi-dimensional arrays and pointers.
CO4	Demonstrate the operations using strings and structures.
CO5	Illustrate linear data structures like stacks and queues.

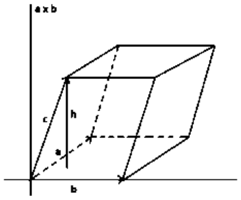
CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	3	-	3	-	-	-	3
CO2	3	3	-	1	-	-	-	3
CO3	3	3	-	1	-	-	-	2
CO4	3	3	-	1	-	-	-	2
CO5	3	3	-	1	-	-	-	3

<table>
<tr><th colspan="2">List of Experiments</th></tr>
<tr><td><b>Week-1</b></td><td><b>Problem 1:</b>
Program to make use of various input-output functions (printf & scanf).
<b>Problem 2:</b>
Program to swap two numbers using a third variable.
<b>Problem 3:</b>
Program to swap two numbers without using third variable.
<b>Problem 4:</b>
Program to demonstrate implicit and explicit type casting.
<b>Problem 5:</b>
Design a standard calculator for performing arithmetic operations like addition, subtraction, multiplication, division and modulus.</td></tr>
<tr><td><b>Week-2</b></td><td><b>Problem 6:</b>
Given a positive integer denoting n, do the following<ul><li>If 1&lt;= n <=9, print the lowercase English word corresponding to the number(e.g one for 1, two for 2 etc.)</li><li>If n &gt; 9, print greater than 9</li></ul><b>Problem 7:</b>
Given a quadratic equation in the form of $ax^2+bx+c=0$. Find the roots of quadratic equation when the value of discriminant (b^2-4ac) is either greater than or equal to zero.
<b>Problem 8:</b>
The sum of angles of a triangle is 180 degrees. Write a C program to check whether a triangle can be formed with the given values for the angles. Make sure the values of angle input by the user must be positive.</td></tr>
<tr><td><b>Week-3</b></td><td><b>Problem 9:</b>
Write a program in C to calculate and print the electricity bill of a given customer. The customer ID, name, and unit consumed by the user should be captured from the keyboard to display the total amount to be paid to the customer. If bill exceeds Rs. 400 then a surcharge of 15% will be charged and the minimum bill should be of Rs. 100.
The charge is as follow:
<table>
<tr><th>Units Consumed</th><th>Charge/unit</th></tr>
<tr><td>upto 199</td><td>@1.20</td></tr>
<tr><td>200 and above but less than 400</td><td>@1.50</td></tr>
<tr><td>400 and above but less than 600</td><td>@1.80</td></tr>
<tr><td>600 and above</td><td>@2.00</td></tr>
</table>
<b>Problem 10:</b>
Print a pattern of numbers from 1 to n as shown below. Each of the numbers is separated by a single space.
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4</td></tr>
<tr><td><b>Week-4</b></td><td>The Fibonacci sequence is a type series where each number is the sum of the two that precede it. The Fibonacci sequence is given by 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, and so on.
<b>Problem 11:</b>
Write the function to find the nth term of the sequence.
<b>Program 12:</b>
Write the function to generate the Fibonacci sequence up to N terms.
<b>Program 13:</b>
Write the function to calculate the sum of N terms of sequence.</td></tr>
</table>

Week-5	**Problem 14:** Write the program to check whether a number is prime or not. **Problem 15:** Write a program to generate prime numbers between a given range. **Problem 16:** Write a program to check whether a number can be expressed as sum of two prime numbers. **Problem 17:** Write a program to find sum of digits of an input integer number. **Problem 18:** Write a program to reverse an input integer number. **Problem 19:** Write a program to find sum of digits of an input integer number. **Problem 20:** Write a program to check an input integer number is palindrome or not. **Problem 21:** Write a program to check an input integer number is Armstrong number or not.
Week-6	**Problem 22:** Write a function int max_of_four(int a, int b, int c, int d) which reads four arguments and returns the greatest of them. **Problem 23:**  There is a series, S, where the next term is the sum of pervious three terms. Given the first three terms of the series, a, b, and c respectively, you have to output the nth term of the series using recursion. Recursive method for calculating n^{th} term is given below. $S(n) = \begin{cases} a & n = 1, \\ b & n = 2, \\ c & n = 3, \\ S(n-1) + S(n-2) + S(n-3) & otherwise \end{cases}$
Week-7	**Problem 24:** Write a program to implement a Function with no arguments and no return type. **Problem 25:** Write a program to implement a Function with arguments and no return type. **Problem 26:** Write a program to implement a Function with no arguments and return type. **Problem 27:** Write a program to implement a Function with arguments and return type. **Problem 28:** The Tower of Hanoi is a mathematical puzzle. It consists of three rods and N disks. The task is to move all disks to another rod following certain rules: • Only one disk can be moved at a time. • Only the uppermost disk can be moved from one stack to the top of another stack or to an empty rod. • Larger disks cannot be placed on top of smaller disks. Write a recursive procedure to implement the same.
Week-8	**Problem 29:** Create an array and read the values from stdin. Iterate the array calculating the sum of all elements. Print the sum and average of elements. **Problem 30:** You are given an array of integers, marks, denoting the marks scored by students in a class. • The alternating elements marks$_0$, marks$_2$, marks$_4$ and so on denote the marks of boys. • Similarly, marks1, marks3, marks5 and so on denote the marks of boys. The array name, marks, works as a pointer which stores the base address of that array. In other words, marks contains the address where marks0 is stored in the memory.

	Write a program to • Find the sum of marks obtained by boys in a subject. • Find whether a boy or a girl student scored the maximum marks.
Week-9	**Problem 31:** Given an array of size n, reverse it. Example: if array, arr = [1,2,3,4,5], after reversing it, the array should be, arr = [5,4,3,2,1] **Problem 32:** Given a string **s** consisting of alphabets and digits, find the frequency of each digit in the given string. **Problem 33:** Define a void update(int *a, int *b). It receives two integer pointers, int* a and int* b. Set the value of a to their sum, and b to their absolute difference. There is no return value, and no return statement is needed. **Problem 34:** Program to demonstrate call by value and call by reference.
Week-10	**Problem 35:** Program to find the sum and average of the contents of an integer array with the help of pointer. **Problem 36:** Program to demonstrate Pointer arithmetic. **Problem 37:** Write a user-defined function to find the length of a string **Problem 38:** Write a user-defined function to copy a string into another string.
Week-11	**Problem 39:** Write a user-defined function to Reverse a given string. **Problem 40:** Write a user-defined function to Concatenate two given strings. **Problem 41:** Program to demonstrate all library functions of <string.h>. **Problem 42:** Given a string “KIET GROUP OF INSTITUTIONS”. Store the string in the character array of appropriate size. Now perform the following operations. • Count number of vowels and consonants. • Count number of white spaces. • Count the frequency of each alphabet. **Problem 43:** You are given n triangles, specifically, their sides ai, bi, and ci. Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different. *Hint:* The best way to calculate **A** area of a triangle with sides a, b and c is Heron's formula: $A=\sqrt{s(s-a)(s-b)(s-c)}$ **Here, s is the semi-perimeter of a triangle; and is represented as:** $s=\frac{a+b+c}{2}$ **Problem 44:** You are transporting some boxes through a tunnel, where each box is a parallelepiped, and is characterized by its length, width and height. The height of the tunnel 41 feet and the width can be assumed to be infinite. A box can be carried through the tunnel only if its height is strictly less than the tunnel's height. Find the volume of each box that can be successfully transported to the other end of the tunnel. *Note: Boxes cannot be rotated.*

<table>
<tr><td></td><td colspan="2">
</td></tr>
<tr><td><b>Week-12</b></td><td><b>Problem 45:</b>
Write a program to read and print employee details using structure. To store 10 employee details, we will use an array of structures. Each element in the array will represent a single employee.
Each Structure i.e. Employee contains:<table><tr><td>Name</td><td>Id</td><td>Salary</td><td>Experience</td></tr></table>Now, input the details of each employee from the user and perform the following operations.<ul><li>Output all employee details by iterating through the array using the ‘for’ loop.</li><li>Details of employee who is getting the highest salary.</li><li>Increment the salary of employees whose experience is more than 10 years by 10000 and display those names.</li></ul><b>Problem 46:</b>
Complex numbers are of the form A+iB, where A is known as Real part of complex number and B is known as Imaginary part of complex number. To add or subtract two complex numbers, just add or subtract the corresponding real and imaginary parts. For instance, the sum of 6 + 3i and 4 + 2i is 10 + 5i. Write a program to add and multiply two complex number using structure.</td><td></td></tr>
<tr><td><b>Week-13</b></td><td><b>Problem 47:</b>
Write a program to implement stack using array. Push the elements 1, 2, 3, 4 into the stack. Then perform the pop operation and display the contents of the stack. It will be displayed 3, 2, 1.
<b>Problem 48:</b>
Write a program to implement linear queue using array. Enter the elements 1, 2, 3, 4, 5 into the queue and delete the elements from the queue.
<b>Problem 49:</b>
Write a program to find the factorial of an input number solve using recursion:
<b>Problem 50:</b>
Write a program to implement Binary Search using recursion.</td><td></td></tr>
<tr><td colspan="2"></td><td><b>90 hours</b></td></tr>
<tr><td colspan="2"><b>Mode of Evaluation:</b>
<table>
<tr><th colspan="5">Evaluation Scheme</th></tr>
<tr><th colspan="3">CA</th><th>ESE</th><th>Total Marks</th></tr>
<tr><th>CA1</th><th>CA2</th><th>CA3</th><td rowspan="3">75</td><td rowspan="3">150</td></tr>
<tr><td>25</td><td>25</td><td>25</td></tr>
<tr><td colspan="3">75</td></tr>
</table></td><td></td></tr>
</table>

Lab Course Code: CS206P	**Lab Course Name:** Operating System Lab	**L**	**T**	**P**	**C**
		0	0	2	1

Pre-requisite: Concepts of computer fundamentals.

Course Objectives:

- To introduce basic concepts of UNIX/Linux operating systems.
- To learn various file and process management schemes in operating systems.

- To write programs in Linux environment using system calls.

Course Outcome:

Students will be able to:	
CO1	Discuss the role of operating systems and their types.
CO2	Apply the concept of CPU scheduling algorithms for execution of programs.
CO3	Apply the concept of inter-process communication, process synchronization and deadlock handling.
CO4	Analyze various memory management techniques.
CO5	Illustrate the concept of I/O management and file system.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	1	-	-	-	-	-	3
CO2	3	3	-	-	-	-	-	2
CO3	3	3	-	-	-	-	-	2
CO4	3	3	-	-	-	-	-	2
CO5	3	2	-	-	-	-	-	2

List of Experiments

Week-1

Familiarization of Linux, and its Interface: Installation, Basic of accessing the Linux starting and shutting process, logging in and logging out.

Week-2

File Basic Commands:
Utility Commands: find, bc, date, cal, time, exit, clear etc.
File Commands: touch, cat, cat >, cat >>, rm, cp, mv, more, less etc.

Week-3

Directory Basic Commands:
Directory Commands: pwd, mkdir, ln, ls, cd, cd /, etc.
Permissions Commands: umask, chmod, chgrp, chown etc.

Week-4

Linux Process Commands: ps, fg, bg, top, kill, nice, etc.

Write a C program to perform process creation using fork() system call.
The fork system call is used for creating a new process in Linux, and Unix systems, which is called the child process, which runs concurrently with the process that makes the fork() call (parent process). After a new child process is created, both processes will execute the next instruction following the fork() system call**.**

- Example of fork() in C
- Parent Child Relationship

Week-5

Write a C program to implement the various process scheduling mechanisms such as FCFS scheduling.
Given n processes with their burst times, the task is to find average waiting time and average turnaround time using FCFS scheduling algorithm. First in, first out (FIFO), also known as first come, first served (FCFS), is the simplest scheduling algorithm. Consider the set of 5 processes whose arrival time and burst time are given below:

Process Id	Arrival time	Burst time
P1	3	4
P2	5	3
P3	0	2
P4	5	1
P5	4	3

Week-6

Write a C program to implement the various process scheduling mechanisms such as SJF scheduling.
Given n processes with their burst times, the task is to find average waiting time and average turnaround time using SJF algorithm with both pre-emptive and non-pre-emptive mode. Consider the set of 5 processes whose arrival time and burst time are given below:

Process Id	Arrival time	Burst time
P1	3	1
P2	1	4

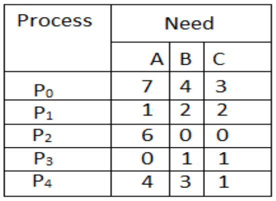
P3	4	2
P4	0	6
P5	2	3

Week-7

Linux Process Synchronization: pipe and tee.
Write a program to implement wait() system call for Process Synchronization.

Week-8

Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
The banker's algorithm is a resource allocation and deadlock avoidance algorithm that simulates resource allocation for predetermined maximum possible amounts of all resources before performing an "s-state" check to look for potential activities and determining whether allocation should be permitted to continue**.** Considering a system with five processes P0 through P4 and three resources of type A, B, C. Resource type A has 10 instances, B has 5 instances and type C has 7 instances. Suppose at time t0 following snapshot of the system has been taken:



Process	Need		
	A	B	C
P_0	7	4	3
P_1	1	2	2
P_2	6	0	0
P_3	0	1	1
P_4	4	3	1

Week-9

Write C programs to simulate FIFO Page Replacement Algorithm.
In operating systems that use paging for memory management, page replacement algorithm is needed to decide which page needed to be replaced when new page comes in. Whenever a new page is referred and not present in memory, page fault occurs, and Operating System replaces one of the existing pages with newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce number of page faults.
Enter the number of pages: 20
Enter the page number: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1.
Enter the number of frames: 3.

Week-10

Write C programs to simulate LRU Page Replacement Algorithm.
In operating systems that use paging for memory management, page replacement algorithm is needed to decide which page needed to be replaced when new page comes in. Whenever a new page is referred and not present in memory, page fault occurs, and Operating System replaces one of the existing pages with newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce number of page faults. In Least Recently Used (LRU) algorithm is a Greedy algorithm where the page to be replaced is least recently used. The idea is based on locality of reference, the least recently used page is not likely.
Let say the page reference string 7 0 1 2 0 3 0 4 2 3 0 3 2.
Initially we have 4-page slots empty.

Week-11

Linux Disk Commands: du, df, fdisk, stat, free etc.

Week-12

Linux User administration Commands: useradd, userdel, groupadd, groupdel, su, etc.

Week-13

Write a C program to implement Linux system calls related to file management.
The system call is a way for programs to interact with the operating system. When the program makes a system call at that time it makes a request to the operating system's kernel. Here, we will discuss the system calls for file management in Linux system. There are four system calls for file management,

- open()
- read()
- write()
- close()

30 hours

Mode of Evaluation:

Evaluation Scheme				
CA			ESE	Total Marks
CA1	CA2	CA3	25	50
5	10	10		
25				

Lab Course Code: EE101P	Lab Course Name: IoT and Embedded System lab	L	T	P	C
		0	0	2	1

Pre-requisite: Not Applicable

Course Objectives:

- The course aims to provide exposure to the applications of IoT in smart cities and industrial applications.
- It aims to train the students to the basic concepts of the Embedded C.
- It aims to train the students to the basic concepts of the Controller.
- This course is designed to give the students hands-on experience with the Software and Hardware concepts.

Course Outcome:

Students will be able to:	
CO1	Understand the basic concepts of sensors and transducers.
CO2	Understand basics of embedded system and different IoT boards.
CO3	Apply basic operations and programming techniques of IoT devices.
CO4	Apply smart technology knowledge through case studies.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	-	-	2	-	-	-	2
CO2	2	-	2	2	2	-	-	2
CO3	3	-	3	3	3	-	-	2
CO4	3	2	3	3	3	-	-	2

List of Experiments (Indicative & not limited to)

1. Understanding the Architecture and Pin Configuration of ESP8266 and Arduino Boards.
2. Hands-On Introduction to commonly used real world IoT Sensors.
3. Analyze Digital signal data acquisition using Arduino and ESP8266.
4. Explore Digital signal generation using Arduino and ESP8266.
5. Analyze Analog signal data acquisition using Arduino.
6. Explore Analog signal generation using Arduino.
7. Real-Time Data Logging Using ESP8266 and Arduino.
8. Designing a Lighting Control System using LDR.
9. Designing a Multi-Sensor Alert System Using Touch, IR, PIR and Arduino.
10. Object Detection Using Ultrasonic Sensors with Arduino and ESP.

Beyond Syllabus:

11. Real-Time Data Logging Using ThingSpeak with Arduino.
12. Building IoT Applications with Blynk: Monitoring Temperature and Humidity with DHT11 Sensor.
13. Building IoT Applications with Blynk: Smart Home Automation Using ESP8266 and Blynk.
14. Building a Soil Health Monitoring system using NPK sensor.
15. Designing Water Quality Monitoring System.

30 hours

Textbook:

1. Raj Kamal , "Internet of Things: Architecture and Design Principles", McGraw Hill Education (India) Private Limited CHENNAI, 2022.

2. Waldemar Nawrocki , "Measurement Systems and Sensors" , Artech House Boston , London, 2021.
3. K. Krishnaswamy and S. Vijayachitra ,"Industrial Instrumentation", New Age International Publishers, 2020.
4. D. Patranabis, "Sensors and Transducers" , PHI Learning Pvt. Ltd. Delhi, 2003.

Reference Books:

1. Murty D.V.S, “Transducers and Instrumentation”, 2nd Edition, PHI, 2008.
2. Rajkumar Buyya and Amir Vahid Dastjerdi , "Internet of Things: Key Applications and Protocols" Elsevier, 2016.
3. Arsheep Bahga & Vijay Madisetti, Internet of Things: A Hands-on approach, Orient Blackswan Pvt. Ltd. - New Delhi, 2015.
4. Pethuru Raj and Anupama C. Raman. “The Internet of Things: Enabling technologies, platforms, and use cases”. Auerbach Publications, 2017.
5. by Donald Norris, The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and Beagle Bone Black, McGraw-Hill Education,2015.

Mode of Evaluation:

Evaluation Scheme				
CA			**ESE**	**Total Marks**
CA1	**CA2**	**CA3**	25	50
5	10	10		
25				

Lab Course Code: ID104B	**Lab Course Name:** Innovation and Entrepreneurship	**L**	**T**	**P**	**C**
		0	0	2	1

Pre-requisite: NA

Course Objectives:

1. To cultivate an innovative mindset among students by introducing them to various types of innovation, success stories, and the importance of creativity in problem-solving and entrepreneurship.
2. To equip students with practical tools for idea generation and commercialization by teaching them techniques for brainstorming, creativity, and developing structured business models using the Business Model Canvas.
3. To enhance students' understanding of market dynamics by providing them with skills to conduct market research, understand customer segmentation, and validate the feasibility of their business ideas through data-driven insights.
4. To foster hands-on learning through prototype development workshops where students can transform their innovative ideas into minimum viable products (MVP) and prepare for investor pitching.

To provide real-world pitching experience by organizing Demo Day presentations where students can pitch their ideas to industry experts and investors, receive feedback, and explore potential opportunities for funding or mentorship.

Course Outcome:

Students will be able to:	
CO1	Demonstrate an understanding of the various types of innovation, their importance in personal and professional growth, and how to apply innovative thinking to solve real-world problems.
CO2	Gain the ability to generate and refine innovative ideas through creative techniques and utilize the Business Model Canvas to structure viable business concepts.
CO3	Develop the skills to conduct comprehensive market research, identify and segment target customers, and validate their business ideas based on market insights and data analysis.
CO4	Transform their innovative ideas into tangible prototypes (Minimum Viable Products) and will acquire the ability to craft and deliver compelling pitches for potential investors and stakeholders.
CO5	Effectively present their business ideas to industry experts and investors, apply feedback to improve their ideas, and explore opportunities for securing funding or mentorship.

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	1	1	2	1	1	1	1	2
CO2	2	2	2	1	1	1	1	2
CO3	1	2	2	1	2	1	1	2

CO4	1	2	2	1	2	1	1	2
CO5	1	2	2	1	2	2	1	2

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

List of Activities/Experiments

Unit-1	**Innovation & Creativity:** • Resource Person: Technical Expert/ Innovator/Entrepreneur • Content Overview: Introduction to Innovation, the importance of Innovation in life, Type of Innovation, Stages of Innovation, success stories, and opportunities available to students.	**07 Hours**
Unit-2	**Idea/ Innovation Generation, Commercialization & Business Model Canvas Workshop:** • Resource Person: Innovation Coaches/Startup Mentors • Content Overview: Techniques for brainstorming, creativity exercises, introduction to the Business Model Canvas, and developing business concepts.	**07 Hours**
Unit-3	**Market Research and Validation Workshop:** • Resource Person: Market Research Analysts/Marketing Professors • Content Overview: Conducting market research, understanding target customers, market segmentation, and validating business ideas.	**08 Hours**
Unit-4	**Prototype Development & Pitching Workshop:** • Resource Person: Product Developers/Venture Capitalists • Content Overview: Creating a minimum viable product (MVP), hands-on prototyping, crafting, and delivering a compelling pitch.	**08 Hours**
	Total Lecture Hours	**30 Hours**

Mode of Evaluation:

Evaluation Scheme				
CA			**ESE**	**Total Marks**
CA1	**CA2**	**CA3**	NA	50
10	20	20		
50				

M.C.A.

Department of Computer Applications

Detailed Syllabus (2024-25)

Lab Course Code: ID102B	**Lab Course Name:** Mini Project-I	**L**	**T**	**P**	**C**
		0	0	4	2

Pre-requisite: Concepts of Computer Fundamentals.

Course Objectives:

- Understand the role of HTML, CSS, and JavaScript in web development.
- Use CSS frameworks like Bootstrap or Tailwind to design responsive web pages.
- Understand role of frontend application development using React JS.

Course Outcome:

Students will be able to:	
CO1	Understand the working of web browser and web application.
CO2	Apply CSS on a web application.
CO3	Construct an interactive web page with the help of Java script.
CO4	Prepare interactive web page using advance Java script (ES6).
CO5	Change existing static website into an interactive one through the use of APIs to retrieve data.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	-	-	1	1
CO2	3	2	2	3	-	-	1	1
CO3	3	2	2	3	1	2	1	1
CO4	3	2	2	3	1	2	1	1
CO5	3	2	1	3	2	3	1	2

List of Experiments

Week 01	Design a static page which explains all the basic tags you study in HTML.
Week 02	Design the following static web pages required for our college web site. a) HOME PAGE: The static home page must contain three frames. b) LOGIN PAGE c) COURSE DETAILS: This page should contain the details of all the courses available. d) REGISTRATION PAGE
Week 03	Build a portfolio website, so you can highlight your best web work and upload it on Github.
Week 04 & 05	Develop and demonstrate the usage of inline, internal and external style sheet using CSS.
Week 06	Write JavaScript to validate the following fields of the Registration page. a) **Name** (The name must contain letters, and its length must be at least 6 characters.). b) **Password** (Password length must be at least 6 characters.). c) **E-mail address** (should not contain invalid characters and must adhere to the standard format name@domain.com) d) **Mobile Number** (Phone number must contain 10 digits only). **Last Name and Address** (Cannot be Empty).
Week 07	Create an HTML page for the following: - a) Take an input from textbox and find whether it is even or odd, on button click. b) Take an input from textbox and find its factorial on button click. c) Take an input from textbox and print its multiplication table. d) Extracts a number between 0 and 999 from a text field and displays it in words.
Week 08	Create an HTML page, including any necessary JavaScript to create a simple calculator similar to available on your windows. It should not accept five-digit numbers, alphabetic characters, or special characters.
Week 09	Create an HTML page with a select box containing a list of five countries. When a user selects a country, its capital should be displayed immediately after. Add CSS to modify property values of the capital's font (color, bold, and font size).
Week 10	Write a JavaScript program to iterate over all the properties of an object in reverse, running a callback for each one.
Week 11	Write a JavaScript program to convert an asynchronous function to return a promise.
Week 12	Write a program to read JSON file and display it on web page. Write a program fetch method in Rest API.
Week 13	E-Commerce Website development using REACT.
	60 Hours

Text Book:

1. Julie C. Meloni, HTML, CSS, and JavaScript All in One, Sams Teach Yourself Pearson Publication, 3rd Edition, 2020.

Mode of Evaluation:

Evaluation Scheme				
CA			**ESE**	**Total Marks**
CA1	**CA2**	**CA3**	50	100
10	20	20		
50				

Lab Course Code: HS101B	**Lab Course Name:** Communication Skills	**L**	**T**	**P**	**C**
		0	0	4	2

Course Objectives:

- To orient the students to the IPA, the RP system and GIP variants of English speech.
- To groom them well for standard pronunciation.
- To compete with other professionals at the international level.

Course Outcome:

Students will be able to:	
CO1	Analyze pronunciation and its application in communication.
CO2	Apply stress intonation and articulation in language for effective communication.
CO3	Understand the usage of vocabulary in real-life situations at the workplace**.**

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

List of Experiments

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	-	-	-	-	3	-	-	2
CO2	-	-	-	-	3	-	-	2
CO3	-	-	-	-	3	-	-	2

Week-1 (4 hours)	Linguistic Competence: Correct English Usages Through Orell Smart Software & worksheet Practice
Week-2 (4 hours)	Advanced Vocabulary Building Through Orell Smart Software & worksheet Practice
Week-3 (4 hours)	Structure of Syllables – Word Stress and Rhythm– Weak Forms and Strong Forms in Context. Practice: Basic Rules of Word Accent - Stress Shift - Weak Forms and Strong Forms in Context.
Week-4 (4 hours)	Comprehension Skills Based on Listening & Reading Practical on a Model Audio – Visual Usage
Week-5 (4 hours)	Application writing through email
Week-6 (4 hours)	Letter writing through email
Week-7 (4 hours)	Technical Paper review
Week-8 & 9 (8 hours)	Professional Introduction using SWOT analysis
Week-10 & 11 (8 hours)	Business idea pitching (connected with Ideathon event)
Week-12 & 13 (8 hours)	Technical idea pitching (connected with Ideathon event)
	Total Hours: 60

Mode of Evaluation (Lab):

Evaluation Scheme						
MSE		**CA**			**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	NA	100
40	40	6	7	7		
80		20				

MCA-2nd Sem

Course Code: CS202B	**Course Name:** Data Structures and Algorithms	**L**	**T**	**P**	**C**
		2	0	4	4

Pre-requisite: Understanding of programming language like C / C++ with all basic concepts.

Course Objectives:

- Provide the knowledge of basic data structures and their implementations.
- Understand importance of data structures in context of writing efficient programs.
- Develop skills to apply appropriate data structures in problem solving.

Course Outcome:

Students will be able to:	
CO1	Describe basic data structures such as arrays.
CO2	Illustrate data structures like linked list, stacks and queues.
CO3	Compare incremental and divide-and-conquer approaches of sorting and searching algorithms.
CO4	Demonstrate the properties of graphs & trees and implement various operations.
CO5	Analyze designing approaches such as greedy, dynamic programming and backtracking.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	-	1	-	-	-	1
CO2	3	3	3	1	-	-	-	2
CO3	3	3	3	1	-	-	-	2
CO4	3	3	3	1	-	-	-	2
CO5	3	3	3	1	-	-	-	3

Detailed Syllabus

Unit 1	**Introduction and Recursion**	**18 hours**

Basics: Data types, Classification of Data Structures, Storage structure of linear array and multidimensional array, matrices and sparse matrices.

Notion of Algorithm: Asymptotic Notations, Mathematical Analysis of Non-Recursive and Recursive algorithms algorithm.

Recursion: Recursion and its types, solving and storage representation of factorial of a number, nth Fibonacci number, Tower of Hanoi.

Practical:

1. Write a program to implement two sum and 3 sum problem.
2. Write a program to find out, is there any duplicate value in array or not.
3. Write a program to find the subarray with the largest sum and return its sum.
4. Write a program to implement Factorial, Fibonacci series and Tower of Hanoi using recursion.

Unit 2	**Linked List and Hashing**	**18 hours**

Linked List: Introduction of Linked List, Memory Representation, Implementation of singly linked list and doubly linked list with various insertion, traversing and deletion operation.

Stacks and Queues: Dynamic implementation of stack & queues and polish notation.

Hashing: Concept of hashing, hash table, hash function and Collison removal techniques.

Practical:

1. Write a program that uses functions to perform the following operations on singly linked list i) Creation ii) Insertion iii) Deletion iv) Traversal.
2. Write a program to reverse a linked list.
3. Write a program that uses functions to perform the following operations on doubly linked list i) Creation ii) Insertion iii) Deletion iv) Traversal.
4. Write a program that implement Stack (its operations) using linked list.
5. Write a program that implement Queue (its operations) using linked list.
6. Write a program to implement polynomial addition and multiplication using linked list.

Unit 3	**Sorting and Searching Techniques**	**18 hours**

Sorting and Searching: Insertion Sort, Linear Search.

Divide and Conquer Algorithms: Binary Search, Merge sort, Quick sort.

Heaps: Introduction to different kinds of Heaps, Heap sort.
Non comparison sorting techniques: Counting Sort and Bucket Sort.
Practical:

1. Write a program that implements the following algorithm i) Bubble sort ii) Selection sort (iii) Linear search
2. Write a program to implement the following algorithm (i) Insertion Sort (ii) Binary Search (iii) Merge Sort (iv) Quick Sort
3. Write a program to implement Heap Sort.

Unit 4	**Trees and Graphs**	**18 hours**

Trees: Introduction to trees, Memory representation of Binary trees using array and linked list, Binary tree traversals.
BST: Introduction of BST, Implementation of various operation like insertion, deletion.
Advanced Tree concepts: AVL tree and B tree.
Graphs: Introduction to graphs, Basic terminology of Graphs, Memory Representations, Graph Traversal Algorithms like BFS and DFS.
Practical:
Write a program to perform the following operations in Binary Search Tree: a) Insertion, b) Deletion, c) Searching and d) Traversal.

Unit 5	**Algorithm Design Paradigms**	**18 hours**

Introduction of different problem-solving approaches: Top-down and Bottom-up approaches.
Greedy Algorithms: Minimum Spanning Tree using Prim's and Kruskal's, Single Source Shortest Path using Dijkstra Algorithm.
Dynamic Programming: Longest Common Subsequence, Knapsack problem.
Backtracking: N-Queen Problem, 15 Puzzle problem.
Practical:

1. Implementation of Prim's and Kruskal's Algorithms.
2. Implementation of Longest Increasing Subsequence.
3. Implementation of Longest Common Subsequence.

Total Lecture Hours	**90 hours**

Textbook:

8. Lipschutz Seymour, Data Structures, Revised First Edition, Schaum's Outline Series, McGraw Hill Education India, 2017.
9. Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson Education, 2012.

Reference Books:

2. Richard F. Gilberg and Behrouz A. Fourouzan, Data structures-A pseudocode approach with C, 2nd Edition, Cengage Learning, 2005.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI Learning Private Limited, 2012.
4. Jean-Paul Tremblay, Paul G. Sorenson, An Introduction to Data Structures with Application, 2nd Edition, Mcgraw Hill Computer Science Series, 2001.
5. Dave P. H., H.B.Dave, Design and Analysis of Algorithms, 2nd Edition, Pearson Education, 2017.
6. Lipschuts S., "Theory and Problems of Data Structures", Schaum's Series, 2017.

Mode of Evaluation:

Evaluation Scheme								
MSE		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	100	200
40	40	5	5	5	5	5		
80		Best of 4 (20)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: CS203B	**Course Name:** Database Management Systems	**L**	**T**	**P**	**C**
		2	0	4	4

Pre-requisite: Concepts of any Programming Language.

Course Objectives:

- Learn the difference between traditional file-based systems and modern database systems.
- Understand the relational data model and its concepts, including tables, tuples, attributes, and keys.
- Design E-R Diagrams and to convert it into relational model.
- Understand normalization and the use of normal forms to reduce data redundancy and improve data integrity.
- Apply SQL and PL/SQL on Oracle Database.

Course Outcome:

Students will be able to:	
CO1	Understand the concepts of ER modeling, relational data models, and database architecture.
CO2	Apply relational algebra and SQL commands for database creation and data retrieval.
CO3	Understand ER-to-relational mapping and enforce integrity constraints in databases.
CO4	Apply normalization techniques to optimize relational database design.
CO5	Apply SQL and PL/SQL concepts to solve database problems.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	-	-	-	-	2
CO2	3	2	2	3	-	-	-	2
CO3	3	2	2	3	-	-	-	2
CO4	3	3	3	-	-	-	-	2
CO5	3	3	3	3	-	-	-	2
PO Target	3	2.4	2.4	3	-	-	-	2

Detailed Syllabus

Unit 1	**Introduction to DBMS and ER Modeling**	**18 hours**

Basics: Data, Information, Database, Database Management Systems. Database Systems Vs File Systems, Data Models, Instances & Schemas, Database Architecture: Levels of Abstraction and Data Independence.

Conceptual Data Modeling (E-R Modeling): Entity Types, Entity Sets, Attributes, Relationship Types, Roles and Mapping Constraints, Weak Entity.

Practical:

- Case Studies: Design E-R diagrams for scenarios (e.g., Banking, Library, Airline).
- Practical Assignment 1: Analyze and create E-R diagrams for a given use case.
- Setting up databases in MySQL/Oracle.
- Practical Assignment 2: Use of DDL Commands.
- Practical Assignment 3: Use of DML Commands (SELECT, INSERT, UPDATE, DELETE).

Unit 2	**Relational Data Model and SQL Basics**	**18 hours**

Relational Model Concepts: Tables, Tuples, Attributes, Keys. Integrity Constraints: Entity Integrity, Referential Integrity, Domain Constraints. ER-to-Relational Mapping.

Practical:

- Practical Assignment 4: SELECT with WHERE and DISTINCT clauses.
- Practical Assignment 5: Column Alias, LIKE, BETWEEN, IN clauses.
- Practical Assignment 6: Implementation of constraints in SQL.

Unit 3	**Relational Algebra and Advanced SQL**	**18 hours**

Relational Algebra: Operations (Select, Project, Union, Intersection, Cartesian Product, Join). Query Representation in Relational Algebra.

Introduction to SQL Joins, Subqueries, and Nested Queries, Built-in Functions

Practical:

- Practical Assignment 7: String & Numeric Functions.
- Practical Assignment 8: Implementation of Joins.
- Practical Assignment 9: Implementation of Subqueries.

Unit 4	**Functional dependency & Database Normalization**	**18 hours**

Concept of Functional Dependencies: Inference Rules, Closure of Attributes, FD Equivalence & Minimal Cover.

Normalization Concepts: 1NF, 2NF, 3NF, BCNF. Lossless Decomposition and Dependency Preservation.

Practical:

- Practical Assignment 10: Implementation of Aggregate Functions, Order by, GROUP BY, and HAVING.

Unit 5	**PL/SQL Programming and Advanced SQL**	**18 hours**

Introduction to PL/SQL: Procedures, Functions, Triggers. Cursors, Exception Handling Views, & Indexing.

Practical:

- Practical Assignment 11: Implementation of Date & Conversion Functions.

- Practical Assignment 12: Implementation of PL/SQL.
- Practical Assignment 13: Implementation of Cursor, Function, Trigger, Views & Indexes.

Total Lecture Hours	**90 hours**

Textbook:

1. Elmasri, Navathe, “Fundamentals of Database Systems”, 6th Edition, Addition Wesley, 2010.
2. Korth, Silbertz, Sudarshan, “Database Concepts”, 6th Edition, Tata McGraw Hill, 2010.
3. Ivan Bayross, “SQL PL/ SQL”, 4th Edition, BPB Publications, 2010.

Reference Books:

1. Bipin C. Desai, “An Introduction to Database Systems”, Galgotia Publications, 1981.
2. Majumdar & Bhattacharya, “Database Management System”, TMH, 2017.
3. Date C J, “An Introduction to Database Systems”, 8th Edition, Pearson Education, 2003.
4. Ramakrishnan Raghu, Gehrke Johannes, “Database System Concepts”, 2nd Edition, Tata McGraw Hill, 1999.
5. Leon & Leon, “Database Management Systems”, 1st Edition, Tata McGraw Hill, 2008.

Mode of Evaluation:

Evaluation Scheme								
MSE		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	100	200
40	40	5	5	5	5	5		
80		Best of 4 (20)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: ID202L	**Course Name:** Software Testing	**L**	**T**	**P**	**C**
		3	0	0	3

Pre-requisite: Concept of any Programming Language

Course Objectives:

- To provide a comprehensive understanding of the fundamental concepts, principles, and importance of software testing in the software development lifecycle (SDLC).
- To equip students with knowledge of test design techniques, including black-box, white-box, and experience-based methods, for creating effective test cases.
- To familiarize students with various types of testing (manual and automated) and their applications in functional and non-functional areas.
- To introduce popular testing tools like Selenium, Bugzilla and demonstrate their practical applications in test automation.

Course Outcome:

Students will be able to:	
CO1	Understand software testing concepts, principles, and the testing lifecycle.
CO2	Apply black-box and white-box testing techniques to validate software functionality.
CO3	Understand levels of testing and regression testing techniques for ensuring software quality.
CO4	Apply test management strategies, including test planning and risk analysis, to optimize the testing process.
CO5	Apply automation testing tools like Selenium for functional testing.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	-	-	-	-	-	-	-
CO2	3	2	2	-	-	-	-	2
CO3	3	-	-	-	-	-	-	2
CO4	3	-	-	-	-	-	-	2
CO5	3	-	-	3	-	-	-	2
PO Target	3	2	2	3	-	-	-	2

Detailed Syllabus		
Unit 1	**Introduction to Software Testing**	**9 hours**
Introduction: Some popular worldwide software failure. What is software testing, Testing Objective, Role of a tester, Skills required by tester. Terminologies in testing: Error, Fault, Failure, Verification, Validation, Test Cases, Test Suits, Test Oracles, Testing Vs Debugging, Testing, Quality Assurance and Quality Control, Limitations of testing, Testing Life Cycle (Phases: Requirement analysis, test planning, test case design, test execution, defect reporting, and closure). Principles of Testing.		
Unit 2	**White Box and Black Box Testing**	**9 hours**
Static vs Dynamic Testing, Black Box Testing Techniques: Boundary Value Analysis (Generalizing Boundary Value Analysis, Robust Boundary Value Testing, Worst-Case Boundary Value Testing, Robust Worst-Case Testing), Equivalence Class Testing, Decision Table based Testing, Cause-Effect Graphing Technique, State Transition Technique White Box Testing – Need, Logic Coverage Criteria, Statement Coverage, Branch Coverage, Condition Coverage, Loop Coverage, Path Coverage, Graph Matrices, Cyclomatic Complexity, Data Flow Testing, Mutation Testing.		
Unit 3	**Testing Levels**	**9 hours**
Levels of Testing: Unit Testing, Integration Testing, System Testing and Acceptance Testing Regression Testing: Progressive Vs Regressive Testing, Objectives of Regression Testing, Regression testing Techniques Experience Based Testing Techniques: Error Guessing, Exploratory Testing, Checklist Based Testing.		
Unit 4	**Test Management**	**9 hours**
Test Management: Organization Structures for Testing Teams, Test Planning, Test case minimization, Test Case Prioritization, Risk Analysis. Debugging: Debugging Process, Debugging Techniques, Debuggers.		
Unit 5	**Automation Testing**	**9 hours**
Automation and Testing Tools: Need for automation, Testing Tool Classification, benefits and Risks of Test automation, Overview of some commercial testing tools, Introduction to Selenium, Functional testing using Selenium, Automation Testing using Bugzilla.		
	Total Lecture Hours	**45 hours**

Textbook:

1. Yogesh Singh, “Software Testing”, Cambridge University Press, New York.
2. Naresh Chauhan, “Software Testing Principles and Practices”,

Reference Books:

1. Wllian E Perry, “Effective Methods for Software Testing”, Wiley
2. Roger S Pressman, Software Engineering – a Practitioners Approach”, McGraw Hill Education.
3. Aditya P Mathur, “Foundation of Software Testing”, Pearson Publications.

Mode of Evaluation:

Evaluation Scheme

MSE		CA					ESE	Total Marks
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	75	150
30	30	4	4	4	3	4		
60		Best of 4 (15)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: CS204B	**Course Name:** Cloud Computing	**L**	**T**	**P**	**C**
		2	0	0	2

Pre-requisite: Basic internet knowledge, general IT technical knowledge, and an understanding of IT business concept

Course Objectives:

The objective of this course is to impart a comprehensive understanding of cloud computing concepts and AWS core services. It aims to equip students with the technical expertise to architect and deploy secure, scalable cloud solutions while ensuring cost optimization and efficient management.

Course Outcome:

Students will be able to:	
CO1	Understand cloud computing concepts, AWS global infrastructure, pricing models, and core services.
CO2	Describe AWS security measures, including Identity and Access Management (IAM), and establish secure virtual private clouds (VPC) using Amazon VPC.
CO3	Demonstrate the use of services like Amazon EC2, AWS Lambda, and AWS Elastic Beanstalk for deploying scalable applications.
CO4	Optimize AWS storage and database services, including Amazon S3, EBS, EFS, Glacier, RDS, and DynamoDB, for diverse use cases.
CO5	Design cost-efficient, reliable, and scalable cloud architecture using AWS principles, Elastic Load Balancing, CloudWatch, and Auto Scaling.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	2	-	-	-	2
CO2	2	3	-	3	-	-	2	-
CO3	2	-	3	3	2	-	-	3
CO4	3	2	3	3	-	-	-	3
CO5	2	3	3	3	-	2	-	3

Detailed Syllabus

Unit 1	**Cloud Fundamentals**	**6 hours**

Cloud Concept Overview: Introduction to Cloud Computing, Advantages of the Cloud, Introduction to AWS, and Moving to the AWS Cloud.
Assignment Activity: Sample Exam Question and Knowledge Check.
Cloud Economics and Billing:
Fundamentals of Pricing, Total Cost of Ownership, Delaware North Case Study, AWS Organizations, AWS Billing and Cost Management, Billing Dashboards, and Technical Support Models.
Activity: Simple Monthly Calculator and Support Plan Scavenger Hunt
Assignment Activity: Sample Exam Question and Knowledge Check.
AWS Global Infrastructure Overview: Introduction to AWS Global Infrastructure, AWS Services, and Service Categories.
Demo: AWS Global Infrastructure
Activity: AWS Management Console Clickthrough
Assignment Activity: Sample Exam Question and Knowledge Check.

Unit 2	**Cloud Security and Networking**	**6 hours**

Cloud Security: AWS Shared Responsibility Model, AWS Identity and Access Management (IAM), Securing a New AWS Account, Securing Accounts & Data, and Working to Ensure Compliance.
Activity: AWS Shared Responsibility Model.
Demo: AWS IAM Console.
Lab: Introduction to AWS IAM.
Assignment Activity: Sample Exam Questions and Knowledge Check.
Networking and Content Delivery: Networking Basics, Amazon VPC, VPC Networking, VPC Security, Route 53, and CloudFront.
Activity: Label this Diagram
Demo: Amazon VPC Console.
Lab: Build a VPC and Launch a Web Server, and Design a VPC
Assignment Activity: Sample Exam Question and Knowledge Check.

Unit 3	**AWS Compute Services**	**6 hours**

Compute Services Overview: Amazon EC2 (Parts 1, 2, and 3), Amazon EC2 Cost Optimization, Container Services, AWS Lambda, and AWS Elastic Beanstalk.
Demo: Amazon EC2
Demo: Amazon EC2 Part Console
Lab: Introduction to Amazon EC2

Activity: Amazon EC2 Versus Managed Services, AWS Lambda Activity, and AWS Elastic Beanstalk Activity.
Assignment Activity: Sample Exam Question and Knowledge Check.

Unit 4	Storage and Database Services	6 hours

Storage Services: Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier.
Activity: Storage Technology Selection.
Demo: Demo of AWS Storage Consoles (EBS, S3, EFS, Glacier)
Lab: Working with EBS.
Assignment Activity: Sample Exam Question and Knowledge Check.
Database Services:
Amazon RDS, Amazon DynamoDB, Amazon Redshift, Amazon Aurora.
Demo: Database Consoles (RDS, DynamoDB).
Activity: Database Case Study**.**
Lab: Build a Database Server.
Assignment Activity: Sample Exam Question and Knowledge Check.

Unit 5	Cloud Architecture and Monitoring	6 hours

Cloud Architecture: AWS Well-Architected Framework Design Principles, Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, Reliability and High Availability, and AWS Trusted Advisor.
Activity: Interpret AWS Trusted Advisor Recommendations.
Assignment Activity: Sample Exam Question and Knowledge Check.
Automatic Scaling and Monitoring: Elastic Load Balancing, Amazon CloudWatch, Scaling and Load Balancing Architectures.
Activity: Elastic Load Balancing Activity and Amazon CloudWatch Activity.
Lab: Scale and Load Balance Your Architecture.
Assignment Activity: Sample Exam Question and Knowledge Check.

Total Lecture Hours	30 hours

Textbook:

1. https://aws.amazon.com/training/awsacademy/.
2. Ben Piper and David Clinton, "AWS Certified Solutions Architect Study Guide: Associate (SAA-C03) Exam," Sybex, 2022.

Reference Books:

1. Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, "Cloud Computing: Concepts, Technology & Architecture," Prentice Hall, 2013.
2. Alberto Artasanchez, "AWS for Solutions Architects: Design Your Cloud Infrastructure by Implementing DevOps, Containers, and Amazon Web Services," Packt Publishing, 2021.
3. Karen Tovmasyan, "Mastering AWS CloudFormation: Learn How to Automate Your Cloud Infrastructure with the Power of AWS CloudFormation," Packt Publishing, 2022.

Mode of Evaluation:

Evaluation Scheme								
MSE		CA					ESE	Total Marks
MSE1	MSE2	CA1	CA2	CA3	CA4 (Attendance)	CA5*	50	100
20	20	2	2	3	3	3		
40		Best of 4 (10)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.
#Best 4 out of 5 will be considered.

Course Code: AI101B	Course Name: Introduction To AI	L	T	P	C
		2	0	2	3

Pre-requisite: Python Programming knowledge and an understanding of Statistics concept.

Course Objectives:
To provide students with a understanding of AI principles and applications, gain insights into computer vision, natural language processing and Gen AI, explore ethical considerations, and acquire hands-on skills in implementing AI solutions for real-world scenarios.

Course Outcome:

Students will be able to:	
CO1	Acquire the basic understanding of the fundamental concepts of artificial intelligence (AI) to implement search algorithms, and game playing strategies.
CO2	Illustrate the insights of data pre-processing techniques and its visualization.
CO3	Give a basic understanding of Machine Learning, NLP and computer vision to solve real-world problems.
CO4	Apply concepts of uncertainty in AI, decision-making frameworks, and reinforcement learning techniques to solve real-world problems.
CO5	Understand the fundamentals of ANN, Generative AI, ChatGPT, and AI ethics while exploring the future potential of AI applications.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	3	2	2	-	1	2
CO2	-	3	3	3	2	-	-	2
CO3	3	3	3	3	3	-	2	3
CO4	3	3	3	3	3	2	2	3
CO5	3	-	3	2	3	-	3	3

Detailed Syllabus

Unit 1	**Introduction to AI**	**12 hours**

Discussion on Course Outcomes and Introduction to AI, Motivation and Role of Artificial Intelligence: AI in various domains (healthcare, finance, etc.), AI from Turing Test to Humanoids, Various Approaches to AI: Symbolic AI, Sub-symbolic AI, Hybrid Approaches, AI Concepts, Terminology, and Application Areas

Agents and Environments: Intelligent agents, Problem-solving agents

Types of AI: Search-Based Systems: Problem formulation, Search algorithms (e.g., BFS, DFS), Rule-Based Systems: Expert systems, Knowledge representation, Learning-Based Systems: Machine learning basics, and Types of learning (supervised, unsupervised).

Adversarial Search and Games: Optimal Decisions in Games, Minimax Algorithm, and Alpha-Beta Pruning

Constraint Satisfaction Problem: Constraint Propagation, Backtracking Search, and Local Search

Lab Hands-On:

1. Basic Python scripts for AI concepts
2. Tic-Tac-Toe Solver using Minimax Algorithm (Google Colab)
3. Introduction to N-Queens and Sudoku Solver (using Backtracking Search) on Google Colab

Unit 2	**Understanding Data**	**12 hours**

Introduction to Data: History of Data, Data Storage, and the Importance of Data.

Data Acquisition Techniques: Methods of data collection.

The Stages of Data Processing: Overview of Data Pre-processing, Data collection, cleaning, Handling missing values, noisy data, Data Transformation, Normalization, and Scaling.

Data Visualization: The importance of visualizing data to understand patterns

Lab Hands-On:

1. Python scripts for data pre-processing using Google Colab
2. Data visualization with Matplotlib using Google Colab
3. Customer Segmentation data visualization on Google Colab

Unit 3	**Domains of AI**	**12 hours**

Overview of Machine Learning (ML): Supervised Learning: Classification, Regression and Unsupervised Learning: Clustering, Dimensionality Reduction

Overview of Natural Language Processing (NLP): Speech Recognition: Technologies and use cases, Natural Language Understanding: Text analysis, Semantic understanding, Natural Language Generation: Text generation, Chatbots, and Machine Translation: Basics of machine translation systems

Overview of Computer Vision: Image Formation: Pixels, Image representation, Image Classification: Convolutional Neural Networks (CNN), and Object Detection: Techniques and algorithms (e.g., YOLO, Faster R-CNN)

Lab Hands-On:
1. Demonstration of Speech-to-Text conversion through a working project on Google Colab
2. Object Detection in Images using pre-trained models on Google Colab
3. Image annotation and image classification using Google Colab

Unit 4	**Uncertainty in AI**	**12 hours**

Uncertainty in AI: Conditional Independence, Bayesian Networks, Bayes' Rule and Naive Bayes Model: Probability-based decision-making.
Simple Decision-Making: Utility Function and Decision Networks
Reinforcement Learning and Real-life examples: Active Learning: Exploration vs. Exploitation, Passive Learning: Reward feedback mechanism, and Model-Based Learning: Learning from a model of the environment.

Lab Hands-On:
1. Classification with Naive Bayes using Python on Google Colab

Unit 5	**Emerging AI**	**12 hours**

Overview of Artificial Neural Networks (ANN): Basic Structure of Neural Networks, Backpropagation and Deep Learning Overview.
Generative Adversarial Networks (GANs): GANs Basics: Generator vs. Discriminator and Applications of GANs.
Chatbots and Generative AI: Chatbot development using AI techniques and Overview of ChatGPT and other Generative Models.
Ethics of AI: Ethical considerations, Bias in AI models, and Responsible AI.
Future of AI: Trends, Challenges, and Opportunities.

Lab Hands-On:
1. Handwritten Digit Recognition using ANN (MNIST dataset) on Google Colab

Total Lecture Hours	**60 hours**

Textbook:
1. NORVIG, P . R. (2022): Artificial intelligence: A modern approach, 4th edition, Pearson
2. **Aurelien Geron (2023): Hands-On Machine Learning With Scikit-Learn, Keras & Tensorflow, 3rd Edition, O'Reilly**

Reference Books:
1. **https://www.ibm.com/think/topics /generative-ai.**
2. **https://www.cs e.iitd.ac.in/~maus am/cours es /cs l333/s pring2014/.**

Mode of Evaluation:

Evaluation Scheme								
MSE		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3***	**CA4** (Attendance)	**CA5***		
30	30	4	4	7	0	7	75	150
60		15						

CA3: Case Study 5 **(2 Marks)** & 2 Certification Courses **(5 marks)**

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.
#Best 4 out of 5 will be considered.

Certification Courses:
1. Introduction to AI (9 Hours) – LinkedIn Course: https://www.linkedin.com/learning/introduction-to-artificial-intelligence-2023/why-you-need-to-know-about-artificial-intelligence. (2 Marks)
2. The Ultimate Data Prep & EDA Course: https://www.udemy.com/course/the-a-to-z-of-data-preprocessing-for-data-science-in-python/. (3 Marks)

Or

1. AI for Everyone by Andrew Ng – Coursera: https://www.coursera.org/learn/ai-for-everyone?action=enroll. (5 Marks)

Course Code: HS201L	**Course Name:** Quantitative Aptitude & Logical Reasoning-II	**L**	**T**	**P**	**C**
		2	0	0	2

Course Objectives:

- Students will gain a strong foundation in basic mathematical concepts, and become proficient in performing calculations, solving equations, and manipulating numerical data.
- Students will learn to think critically, make deductions, and draw logical conclusions based on given information. This will enable them to approach complex problems systematically and derive accurate solutions.
- Students will learn how to interpret and analyze data presented in various formats such as tables, graphs, and charts. This skill is particularly useful for tasks involving statistical analysis and decision-making.
- It will help students to improve their speed and accuracy in solving problems. Through practice and exposure to different types of questions, students will learn time-saving techniques and develop strategies to effectively manage time during exams or real-world scenarios.
- It will help students to prepare for various competitive exams, such as aptitude tests for job placements, college admissions tests for higher education (GATE/CAT/GRE/GMAT), and government entrance exams. The course will provide the necessary knowledge and practice to perform well in these exams.

Course Outcome:

Students will be able to:	
CO1	Analyze problems, identify relevant information, and apply appropriate mathematical methods to reach solutions.
CO2	Analyze tasks and activities by following a chain of thought process and find logical solutions to a problem.
CO3	Analyze trends, patterns, and relationships within the data.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	2	2	-	-	-	-	-	-
CO2	2	2	-	-	-	-	-	-
CO3	1	1	-	-	-	-	-	-

Detailed Syllabus

Unit 1		**6 hours**

Blood Relation, Dice, Cube, Clock, and Calendar

Basic concepts, definition and terminology related to blood relationships, Conversation based blood relationships, Family Tree based problems, Coded relationships, Introduction of Dice, Cube and Cuboids, Learn standard dice and ordinary dice concept and various type of problems, Open and closed dice, Cube and Cuboids paint the face with different color and, cut the cube in different layer and then solve questions accordingly, Definition and Introduction of Concept and Relation of angle and time, Overtaking, overlapping, right-angle and straight Angle with respect to time, Error in clock (faster and slower), Correct time of clock, Mirror and Water Image of clock, Introduction of Calendar, Concept of Normal and Leap Year, Finding Odd days, Finding the day of the week of given date with and without reference.

Unit 2		**6 hours**

Time, Speed and Distance (TSD), Time and Work

Basic concepts, definition, and terminology related to TSD, Direct and Indirect relation of TSD, Unit Conversions, Average Speed, Relative Speed, Problem on Train, Problem on Boat & Stream, Race, and Games, Introduction to the Time and Work, Work and Wages, Efficiency based problems, Time and work-based problems, Pipe & Cistern based problems.

Unit 3		**6 hours**

Logical Deduction and Critical Reasoning

Syllogism, Argument – Identifying the Different Parts (Premise, assumption, conclusion), Course of Action, Cause and Effect, Assertion and Reason, Statement and Assumption, Logical Deduction.

Unit 4		**6 hours**

Permutation, Combination, and Probability

Definition and Introduction of permutation and combination, Fundamental principle of counting by “AND” or “OR” rule, Forming of Numbers, Words, and Team, Problems related to linear and nonlinear arrangement, Application based Geometry, Match, handshake, Chessboard, Whole number, and Natural number solution, etc., Distribution: - Identical & Distinct, Basic concepts, definition and terminology related to probability, Problem based on Dice, Coins, Playing cards, Marbles, etc., Conditional Probability – Bayes theorem, Question based on Binomial theorem.

Unit 5		**6 hours**

Analytical and Nonverbal Reasoning, Data Sufficiency,
Analytical Reasoning Problems, Puzzles, Mirror Image, Water Image, Paper cutting and folding, Embedded Figures, Non-verbal series, Group of Images, Number in figures, Problems based on Data Sufficiency.

Total Lecture Hours	**30 hours**

Textbook:
1. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publication, 2017.
2. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publication, 2018.

Reference Books:
1. "How to Prepare for Quantitative Aptitude for the CAT" by Arun Sharma, 10th Edition, TMH Publication, 2022.
2. "How to Prepare for Logical Reasoning for the CAT" by Arun Sharma, 7th Edition, TMH Publication, 2024.
3. "How to Prepare for Data Interpretation for the CAT" by Arun Sharma, 8th Edition, TMH Publication, 2024.

Mode of Evaluation:

Evaluation Scheme								
MSE		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	50	100
20	20	2	2	3	3	3		
40		Best of 4 (10)						

CA5: Participation in any Hackathon, National/International Activity or Research Paper Publication.

Course Code: ID201B	**Course Name:** Mini Project-2 (Full Stack Development)	**L**	**T**	**P**	**C**
		2	0	6	5

Pre-requisite: Concept of any programming language.

Course Objectives:

- Understand object-oriented programming concept using Java.
- Understand role of data structure algorithm using Java collection.
- Understand Backend application development using Spring Boot.

Course Outcome:

Students will be able to:	
CO1	Develop the object-oriented programming concepts using Java.
CO2	Illustrate exception handling, file handling, and multi-threading in Java.
CO3	Apply new java features to build java programs.
CO4	Analyse java programs with Collection Framework.
CO5	Test web and RESTful Web Services with Spring Boot using Spring Framework concepts.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	1	-	-	-	-	-	-	1
CO2	1	-	-	-	-	-	-	1
CO3	2	-	-	-	-	-	-	1
CO4	3	-	-	-	-	-	-	2
CO5	2	2	2	-	-	-	-	3

Detailed Syllabus

Unit 1	**Core Java & Object-Oriented Concepts**	**24 hours**

Introduction: Why Java, History of Java, JVM, JRE, Java Environment, Java Source File Structure, and Compilation. Fundamental.
Programming Structures in Java: Defining Classes in Java, Constructors, Methods, Access Specifies, Static Members, Final Members, Comments, Data types, Variables, Operators, Control Flow, Arrays & String.
Object Oriented Programming: Class, Object, Inheritance Super Class, Sub Class, Overriding, Overloading, Encapsulation, Polymorphism, Abstraction, Interfaces, and Abstract Class.
Packages: Defining Package, CLASSPATH Setting for Packages, Making JAR Files for Library Packages, Import and Static Import Naming Convention for Packages.

Unit 2	**Multi-Threading, File Handling with Exception Handling**	**24 hours**

Exception Handling: The Idea behind Exception, Exceptions & Errors, Types of Exception, Control Flow in Exceptions, JVM Reaction to Exceptions, Use of try, catch, finally, throw, throws in Exception Handling, In-built and User Defined Exceptions, Checked and Un-Checked Exceptions.
Multithreading: Thread, Thread Life Cycle, Creating Threads, Thread Priorities, Synchronizing Threads, Inter-thread Communication.

Unit 3	**Functional Programming using New Java Concepts**	**24 hours**

Java New Features: Functional Interfaces, Lambda Expression, Method References, Stream API, Default Methods, Static Method, Private Method, For Each Method, Java Module System, Local Variable Type Inference, Switch Expressions, Yield Keyword, Text Blocks, Records, Sealed Classes.

Unit 4	**DSA using Java**	**24 hours**

Java Collections Framework: Collection in Java, Collection Framework in Java, Hierarchy of Collection Framework, Iterator Interface, Collection Interface, List Interface, ArrayList, LinkedList, Vector, Stack, Queue Interface, Set Interface, HashSet, LinkedHashSet, SortedSet Interface, TreeSet, Map Interface, HashMap Class, LinkedHashMap Class, TreeMap Class, Hashtable Class, Sorting.

Unit 5	**Backend Programming using Java**	**24 hours**

Spring Framework: Spring Core Basics-Spring Dependency Injection concepts, Spring Inversion of Control, Bean Scopes- Singleton, Prototype, Request, Session, Application, Web Socket, Auto wiring, Annotations, Life Cycle Call backs, Bean Configuration styles.
Spring Boot: Spring Boot Build Systems, Spring Boot Code Structure, Spring Boot Runners, Logger, BUILDING RESTFUL WEB SERVICES, Rest Controller, Request Mapping, Request Body, Path Variable, Request Parameter, GET, POST, PUT, DELETE APIs, Build Web Applications.

Total Lecture Hours	**120 hours**

Textbook:

10. Java: The Complete Reference, Eleventh Edition, 11th Edition. by Herbert Schildt. Released, 2018.
11. Craig Walls, "Spring Boot in Action", 5th Edition, Manning Publication, 2015.

Reference Books:

4. Steven Holzner, "Java Black Book", Edition, Dreamtech, 2001.
5. Balagurusamy E, "Programming in Java", 5th Edition, McGraw Hill, 2017.
6. Java: A Beginner's Guide by Herbert Schildt, 9th Edition, Oracle Press, 2022.
7. Greg L. Turnquist "Learning Spring Boot 2.0 – 2nd Edition", Packt Publication, 2017.
8. AJ Henley Jr (Author), Dave Wolf, "Introduction to Java Spring Boot, Learning by Coding", Independently Published, 2019.

Mode of Evaluation:

			Evaluation Scheme			
MSE (100)		**CA (25)**			**ESE (125)**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**		
1. Project presentation (20 marks) 2. Synopsis Submission (15 marks)	1. Project Demonstration (20 marks) 2. Project report (10 marks) 3. Viva (15 marks) 4. Project submission on GitHub (5 marks)	Best 5 practicals out of 6: Lab Conduction (2 marks)	Best 5 practicals out of 6: Lab Conduction (2 marks)	Quiz (5 marks)	1. Project Demonstration (60 marks) 2. Viva (30 marks) 3. SDG Level Mapping (10 marks)	250

3. Viva (15 marks)					4. Real world project implementation (10 marks) 5. Quiz (15 marks)	
50	50	10	10	5	125	

List of Practical's:

Unit 1: Core Java with Object Oriented Programming

1. Write a Java program to create a Room class, the attributes of this class is roomno, roomtype, roomarea and ACmachine. In this class the member functions are setData and displayData. Use member function to set data and display that data using displayData() method.
2. Write a Java program to create a class named Shape and create three sub classes Circle, Triangle and Square, each class has two-member function named draw () and erase (). Implement this concepts using polymorphism.
3. Write a Java program to create class Number with only one private instance variable as a double primitive type, include the following methods isZero(), isPositive(), isNegative( ), isOdd( ), isEven( ), isPrime(), isAmstrong() in this class and all above methods should return boolean primitive type like for isPositive() should return "Positive = True".

Unit 2: Java Exception Handling, and Multithreading Programming

4. Create a class Customer having following members:
private String custNo
private String custName
private String category
Parameterized constructor to initialize all instance variables
Getter methods for all instance variables
Perform following validations in the constructor
- custNo must start with 'C' or 'c'
- custName must be atleast of 4 characters
- category must be either 'Platinum', 'Gold' or 'Silver '
When any of these validations fail, then raise a user defined exception InvalidInputException
Create a class TestCustomer having main method. Ask user to enter customer details. Create an object of Customer and perform validations. Print details of customer.
5. Create a class MyThread derived from Thread class and override the run method.
Create a class ThreadDemo having main method. Create 2 objects of MyThread class and observe the behavior of threads

Unit 3: Java New Features (Functional Programming in Java)

6. Write a Java program to implement a lambda expression to find the sum of two integers.
7. Write a Java program to implement a lambda expression to filter out even and odd numbers from a list of integers.
8. Write a Java program to remove all duplicate elements from a list using streams.

Unit 4: Java Collection (DSA in Java)

9. Write a Java program to create an array list, add some colors (strings) and print out the collection.
10. Write a Java program to iterate through all elements in an array list.
11. Write a Java program to insert an element into the array list at the first position.

Unit 5: Spring Boot (Java as a Backend)

12. Witte a CRUD Web Application using React JS or JavaScript frontend with Spring Boot as backend

Lab Course Code: HS202B	Lab Course Name: Campus to Corporate	L	T	P	C
		0	0	2	1

Course Objectives:

- Enable students to develop advanced employability skills, including verbal ability and effective communication, for industry readiness.
- Provide insights into the nuances of the recruitment process, professional ethics, and workplace communication dynamics to help students transition smoothly into the corporate environment.

Course Outcome:

Students will be able to:	
CO1	Demonstrate professional readiness by mastering verbal skills in grammar, vocabulary, and comprehension, enhancing digital and spoken communication, and addressing key recruitment components like Resumes and Visumes (video resumes).
CO2	Apply effective communication strategies in group discussions, interviews, and collaborative tasks to achieve clarity and confidence.
CO3	Evaluate professional situations and exhibit team dynamics, and lifelong learning skills for problem-solving and collaboration.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High):

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	-	2	-	2	2	1		2
CO2	-	1	-	1	2	-	1	2
CO3	-	-	-		2	-	1	2

S.No	**List of Practical**
1	**Understanding various job profiles in the VUCA & BANI world** **Overview-** Identifying skills required for different job profiles **Activity-**Research about the job profiles and team presentations.
2	**Resume crafting as per the ATS guidelines along with Cover letter and VISUME** **Overview -** The students will be given guidelines to prepare their resumes as per industry norms. **Activity:** Crafting a resume for employability purposes.
3	**Company-based Verbal Ability Papers-I** **Overview –** The students will be made to solve and discuss the previous year/company pattern-based VA Paper **Activity** – Solving Capgemini VA paper followed by discussion of solutions (1-hour), & 1-hour GD session with feedback
4	**Introduction to Interview** **(4 Ts, Dos and Don'ts) and Top 10 HR Questions** **Overview -**To familiarize and acquaint the students with various types of interviews along with the dos and don'ts (Virtual & offline). The students will be made to go through the most commonly asked questions in HR interviews and use the LinkedIn Interview Prep tool. **Activity:** Conduct a sample interview and show an appropriate video followed by a discussion in class. Discussion with the students will be held in class wherein they will be asked to give suitable answers to these questions and constructive feedback will be shared by the trainer for improvement.
5	**Company-based Verbal Ability Papers-II** **Overview –** The students will be made to solve and discuss the previous year/company pattern-based VA Paper **Activity** – Solving Cognizant VA paper followed by discussion of solutions (1-hour), & 1-hour GD session with feedback
6	**E-mail etiquette in the digital age** **Overview-** The students will be able to draft professional E-mails that are significant in the digital age along with the apt usage of ChatGPT **Activity-** E-mail writing on various scenarios
7	**GD Practice** **Overview -**The students will be put in 10 groups of 3 each and given Advanced Case study-based topics. They will then be given time to prepare and present their ideas and thoughts in the form of a discussion **Activity -**Team discussion in class along with constructive feedback
8	**Company-based Verbal Ability Papers-III** **Overview –** The students will be made to solve and discuss the previous year/company pattern-based VA Paper **Activity** – Solving TCS VA paper followed by discussion of solutions (1-hour), & 1-hour visume preparation & recording
9	**Essay/Paragraph Writing** **Overview-** The students will be introduced to the structure and elements of a well-written paragraph and essay. **Activity-** Paragraph/Essay writing on topics asked in recruitment rounds
10	**Company-based Verbal Ability Papers-IV** **Overview –** The students will be made to solve and discuss the previous year/company pattern-based VA Paper **Activity** – Solving Infosys VA paper followed by discussion of solutions (1-hour), & 1-hour visume preparation & recording
	30 hours

References:

1. Jeff Butterfield, (2019) Soft Skills for Everyone, 2nd Edition, Cengage Learning
2. Personality Development and Soft Skills, By Barun Mitra
3. Business Communication for Managers; Payal Mehra, Pearson Delhi, 2012
4. Uma Maheshwari, (2018), Soft Skills for Campus Placements, 1st Edition, Wiley India Private Limited.
5. Wren, P.C.; Martin, H.; Prasada Rao, N.D.V. (1973–2010). High School English Grammar & Composition. New Delhi, S. Chand. ISBN 81-219-2197-X.
6. Sherfield et. al. (2012) Developing Soft Skills, 4th Edition, Pearson Education

Mode of Evaluation (Lab):

Evaluation Scheme								
MSE@		**CA**					**ESE**	**Total Marks**
MSE1	**MSE2**	**CA1**	**CA2**	**CA3**	**CA4** (Attendance)	**CA5**	NA	50
20	20	2	2	3	3	3		
40		Best of 4 (10)						