Sanjay Ghodawat University Kolhapur



School of Engineering and Technology

Department of Artificial Engineering and Data Science

B. Tech. Artificial Engineering and Data Science

Curriculum Structure and Syllabus (R26)

Semester: I-II

Programme Structure R26 for B.Tech.: First Year Artificial Intelligence and Data Science

Semester I: Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total credits
UADPC106	Web Application Development I	3	-	-	3	-	-	3
UADPC107	Computer Programming	3	-	-	3	-	-	3
UADPC103	Introduction to Generative AI	1	2	-	-	2	-	2
USTBS105	Mathematics for Computer Science	2	-	-	2	-	-	2
UADAE103	Quantitative Aptitude	3	-	-	3	-	-	3
UHSAE006	Communication English Foundation	1	2	-	-	2	-	2
UADPC108	Web Application Development I Laboratory	-	4	-	-	2	-	2
UADPC105	Computer Programming Laboratory	-	4	-	-	2	-	2
UHSVE001 **OR** USTBS206	Constitution of India **OR** Modern Physics	2	-	-	2	-	-	2
	Total	**15**	**12**	**-**	**13**	**8**	**-**	**21**

Semester I: Evaluation Scheme

Course Code	Course Name	Theory Marks					Practical Marks		Total
		Internal Assessment				SEE	Term work / Tutorial	Practical Oral/ POE	
		T1	T2	FET	Total				
UADPC106	Web Application Development I	10	10	5	25	50	-	-	75
UADPC107	Computer Programming	10	10	5	25	50	-	-	75
UADPC103	Introduction to Generative AI	-	-	-	-	-	50	-	50
USTBS105	Mathematics for Computer Science	-	10	5	15	35	-	-	50
UADAE103	Quantitative Aptitude	10	10	5	25	50	-	-	75
UHSAE006	Communication English Foundation	-	-	-	-	-	50	-	50
UADPC108	Web Application Development I Laboratory	-	-	-	-	-	25	25	50
UADPC105	Computer Programming Laboratory	-	-	-	-	-	25	25	50
UHSVE001 **OR** USTBS206	Constitution of India **OR** Modern Physics	10	-	5	15	35	-	-	50
								Total	**525**

Programme Structure R26 for B.Tech.: First Year Artificial Intelligence and Data Science

Semester II: Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total Credits
UADPC201	Web Application Development II	3	-	-	3	-	-	3
UADPC202	Database Management System	3	-	-	3	-	-	3
UADPC203	Data Structures	4	-	-	4	-	-	4
UADPC204	Building LLM Applications	1	2	-	-	2	-	2
UADAE201	Numerical Ability	2	-	-	2	-	-	2
UHSAE007	Communicative English Advanced	1	2	-	-	2	-	2
UADPC205	Database Management Systems Laboratory	-	2	-	-	1	-	1
UADPC206	Web Application Development II Laboratory	-	2	-	-	1	-	1
UADPC207	Data Structures Laboratory	-	4	-	-	2	-	2
UHSVE001 **OR** USTBS206	Constitution of India **OR** Modern Physics	2	-	-	2	-	-	2
Total		**16**	**12**	-	**14**	**8**	-	**22**

Semester II: Evaluation Scheme

Course Code	Course Name	Theory Marks					Practical Marks		
		T1	T2	FET	Total	SEE	TermWork/ tutorial	Practical Oral/POE	Total
UADPC201	Web Application Development II	10	10	5	25	50	-	-	75
UADPC208	Database Management System	10	10	5	25	50	-	-	75
UADPC202	Foundation of Data Structures	10	10	5	25	75	-	-	100
UADPC204	Building LLM Applications	-	-	-	-	-	50		50
UADAE201	Numerical Ability	-	10	5	15	35	-	-	50
UHSAE007	Communicative English Advanced	-	-	-	-	-	50	-	50
UADPC210	Database Management Systems Laboratory	-	-	-	-	-	25	-	25
UADPC209	Web Application Development II Laboratory	-	-	-	-	-	25	-	25
UADPC207	Foundation of Data Structures Laboratory	-	-	-	-	-	25	25	50
UHSVE001 **OR** USTBS206	Constitution of India **OR** Modern Physics	-	10	5	15	35	-	-	50
								Total	**550**

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC106	Web Application Development I	3	-	-	3	-	-

Course Code	Course name	Evaluation scheme (In semester)					Semester end exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min pass	
UADPC106	Web Application Development I	10	10	5	-	25	50	40%	75

WEB APPLICATION DEVELOPMENT I

Course Description

This course introduces the fundamental concepts of web and frontend development using HTML and CSS. It covers the structure of web pages, HTML elements, semantic components, and multimedia integration. Students learn CSS styling techniques including text properties, the CSS Box Model, selectors, and specificity. The course also explores layout design using Flexbox and CSS Grid, along with responsive design using media queries and positioning. GenAI tools are introduced to support faster learning and modern development practices.

Course Objectives

1. To introduce the fundamentals of web development using HTML and CSS.
2. To develop the ability to design structured and styled web pages using modern HTML5 and CSS features.
3. To enable students to apply layout techniques such as Flexbox and Grid for building responsive web interfaces.
4. To familiarize students with CSS selectors, positioning, and responsive design principles.
5. To provide hands-on experience in developing responsive websites and portfolio pages using modern web technologies and GenAI tools.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Identify**[1] the core concepts of web development, HTML structure, GenAI tools, and foundational CSS syntax used in frontend development.

CO2 **Describe**[2] CSS text properties, the CSS Box Model, HTML list elements, and styling rules for structuring and formatting web content.

CO3 **Apply**[3] CSS Box Model, Flexbox, and Grid layout techniques to design structured and well-aligned web pages with appropriate spacing and positioning.

CO4 **Analyze**[4] HTML5 semantic elements, multimedia features, hyperlinks, and advanced CSS concepts to evaluate their role in accessible web development.

CO5 **Design**[6] and implement responsive web interfaces using CSS Grid, Flexbox, media queries, and positioning techniques that function effectively across different devices and screen sizes.

Course Contents

Module	Unit	Description	Hours
1.0		**INTRODUCTION TO WEB DEVELOPMENT & HTML FUNDAMENTALS**	9
	1.1	GenAI in Web Development: Introduction to GenAI, Building First Frontend Project, Leveraging Gen AI for Accelerated Learning, HTML Basics: HTML Structure, HTML Elements.	
	1.2	CSS Fundamentals: Container Element, CSS Syntax, class attribute, Text-align Property, Text Color, Background-color.	
2.0		**CSS STYLING & BOX MODEL**	9
	2.1	CSS Text Properties: Multiple Rules in Ruleset, Multiple Rulesets, Font-family, Font-size, Text-decoration, Font-style.	
	2.2	CSS Box Model: Height, Width, Background-image, Background-size, Viewport, Border Properties, Padding, CSS Colors in Hex Code, HTML Elements & Lists: Image Element, Void Elements, Margin, Ordered Lists, Unordered Lists, List-style-type.	
3.0		**HTML5, SEMANTICS & ADVANCED CSS**	9
	3.1	Web Infrastructure & Hyperlinks: Internet, IP Address, AWS Route S3, Anchor Element, Navigate within Same Document, Image as Link, Break, Horizontal Rule.	
	3.2	HTML5 Features: HTML5 Versions, Multimedia Support (Video & Audio Elements), Semantic Elements, Non-Semantic Elements, New Input Types, Location Support, More CSS Concepts: Margin Properties, Padding Properties, Margin-auto, Span Element, Width & Height Percentages, Gradients (Linear, Radial).	
4.0		**CSS SELECTORS, SPECIFICITY & FLEXBOX**	9
	4.1	CSS Selectors: Inherited & Non-inherited Properties, Simple Selectors (type/tag, class, id), Pseudo Classes, Pseudo Elements, Combinators, Attribute Selectors.	
	4.2	CSS Specificity & Sizing: Specificity, Cascading, !important & Overriding, Sizing Elements, Handling Overflow, Min-Size & Max-size, Meta Element, Box Sizing & Flexbox: Box Sizing Property, Universal Selector, Display Flex, Flex-direction, Justify-content, Align-items, Flex-wrap, Align-self, Order, Flex-grow, Flex-shrink.	
5.0		**RESPONSIVE DESIGN, GRID LAYOUT & POSITIONING**	9
	5.1	Media Queries & Responsive Design: Media Types, Media Features, Logical Operators, Combining Multiple Media Features, Building Responsive Web Pages, CSS Grid Layout: Grid, Grid-template-columns, Grid-template-rows, Gap, Fr unit, Repeat() function, Grid-row, Grid-column, Auto.	
	5.2	Grid-template-areas, Grid-area, Aligning Rows and Columns, Aligning Grid Items, Aligning Specific Grid Item, Media Queries with Grid, Auto-fit, Minmax, Flexbox vs Grid, CSS Positioning: Position Property, Changing Element's Position, Position Property Values, Scrolling Elements, Ancestors, Positioned Ancestors, Fixed Vs Absolute, TailwindCSS, Building a Responsive Website using GenAI.	

Text Books

1. Terry Felke-Morris, Web Development and Design Foundations with HTML5, 9th Edition, Pearson, 2023.
2. Jennifer Niederst Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, 5th Edition, O'Reilly Media, 2018.

Reference Books

1. Thomas A. Powell, HTML & CSS: The Complete Reference, McGraw-Hill.
2. Eric A. Meyer & Estelle Weyl, CSS: The Definitive Guide, 4th Edition, O'Reilly Media.
3. Keith Grant, CSS in Depth, Manning Publications.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. Fifth module will be assessed for 5 marks separately.
3. Tutorial should be assessed based on accuracy, completeness, conceptual understanding, problem-solving approach, clarity of explanation, and presentation quality.
4. Term work assessment shall be based on the overall performance of the student with every tutorial graded from time to time.
5. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC107	Computer Programming	3	-	-	3	-	-

Course Code	Course Name	Evaluation scheme (In semester)					Semester end exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min pass	
UADPC107	Computer Programming	10	10	5	-	25	50	40%	75

COMPUTER PROGRAMMING

Course Description

This course introduces the fundamentals of Python programming and problem solving. It covers core concepts such as variables, data types, operators, input/output operations, and type conversion. Students learn to control program flow using conditional statements and loops. The course also explores Python data structures including lists, tuples, sets, and dictionaries, along with functions and argument types. Additionally, students are introduced to the Python standard library and module usage for building efficient programs.

Course Objectives

1. To introduce the fundamentals of Python programming including variables, data types, operators, and input/output operations.
2. To enable students to apply conditional statements and loops for solving basic programming problems.
3. To develop the ability to work with Python data structures such as lists, tuples, sets, and dictionaries.
4. To familiarize students with functions, argument types, and Python standard libraries.
5. To provide hands-on experience in developing Python programs for solving computational and data-processing tasks.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] Python fundamentals including variables, data types, indentation, input/output operations, and the order of execution.

CO2 **Describe**[2] relational and logical operators and explain conditional branching using if, if-else, and if-elif-else statements to control program flow.

CO3 **Implement**[3] while loops, for loops, nested loops, and control statements (break, continue, and pass) to solve iterative and repetitive programming problems.

CO4 **Analyze**[4] list and tuple operations, including indexing, slicing, modification, and built-in methods, to evaluate their suitability for managing collections of data.

CO5 **Develop**[6] Python programs using functions, sets, dictionaries, and standard libraries by importing modules and applying built-in functions to solve data-processing tasks.

Course Contents

Module	Unit	Description	Hours
1.0		**PYTHON BASICS**	9
	1.1	Python Fundamentals, Variables, Data types, Order of Execution, Indentation, Input/Output Operation.	
	1.2	Operators and Expressions: Arithmetic, Comparison, and Operator Precedence, Type Conversion: Checking Data Type, String to Integer, Integer to String.	
2.0		**CONTROL STRUCTURES AND CONDITIONAL STATEMENTS**	9
	2.1	Relational Operators, Logical Operators.	
	2.2	Conditional Branching Statements: if, if-else, Nested if, if-elif-else.	
3.0		**LOOPS AND CONTROL STATEMENTS**	9
	3.1	Loops: while, for loops, Nested loops.	
	3.2	Control Statements: continue, break, pass, String Comparison: String Comparison, Character by character comparison, and best practices.	
4.0		**LISTS AND TUPLES**	9
	4.1	Lists: Creation of Lists, Accessing List Items, Concatenation, Slicing, Extended Slicing, Indexing.	
	4.2	Lists: Modifying, Splitting, Joining Lists, Negative Indexing, Tuples: Creating Tuples, Accessing Elements, Tuple Methods.	
5.0		**FUNCTIONS, SETS, AND DICTIONARIES**	9
	5.1	Functions: Defining Functions, Calling Functions, Returning Values, Built-in Functions, Function Arguments: Keyword Arguments, Positional Arguments, Default Values, Passing Mutable Objects.	
	5.2	Set: Creating a Set, Operations on Sets: Union, Intersection, Difference, Symmetric Difference, Dictionaries: Creating a Dictionary, Accessing Items, Dictionary Methods, Working with Standard Library: Importing module, Importing from a Module, Aliasing Imports.	

Text Books

1. Matthes, E., Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 3rd ed., No Starch Press, 2023.
2. Downey, A. B., Think Python: How to Think Like a Computer Scientist, 3rd ed., O'Reilly Media, 2024.

Reference Books

1. Lutz, M., Learning Python: Powerful Object-Oriented Programming, 6th ed., O'Reilly Media, 2025.
2. Ramalho, L., Fluent Python: Clear, Concise, and Effective Programming, 2nd ed., O'Reilly Media, 2022.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. Fifth module will be assessed for 5 marks separately.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
USTBS105	Mathematics for Computer Science	2	-	-	2	-	-

Course Code	Course name	Evaluation scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term Work	Total	Marks	Min Pass	
USTBS105	Mathematics for Computer Science	-	10	5	-	15	35	40%	50

MATHEMATICS FOR COMPUTER SCIENCE

Course Description

This course introduces mathematical concepts used in computing systems. It covers number systems, base conversions, signed an d floating-point representations, and bitwise operations. Students learn number theory concepts such as GCD, LCM, modular arithmetic, and error detection techniques. The course also introduces combinatorics and probability for analyzing real-world problems. In addition, it explores matrices, mathematical series, and set theory concepts including relations and functions used in computational problem solving.

Course Objectives

1. To introduce different number systems and their representations in computing systems.
2. To develop problem-solving skills using modular arithmetic, GCD/LCM, and number theory concepts.
3. To understand combinatorics and probability for analyzing computational and real-world problems.
4. To familiarize students with matrices, series, and their applications in computing.
5. To apply concepts of set theory, relations, and functions in organizing and analyzing data.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] the types of number systems, base conversions, arithmetic operations, bitwise operators, and integer representations used in computing.

CO2 **Describe**[2] signed and unsigned integer representations, including sign magnitude, 1's complement, and 2's complement methods, and explain their properties.

CO3 **Apply**[3] arithmetic techniques including GCD/LCM, Euclidean algorithms, modular arithmetic, and multiplicative inverses to solve computational problems.

CO4 **Analyze**[4] counting principles, permutations, combinations, probability, conditional probability, and Bayes' theorem to evaluate real-world scenarios.

CO5 **Evaluate**[5] matrix operations, graph representations, AP, GP, HP, and AGP series, as well as set theory concepts including relations and functions, to solve data-oriented problems.

Course Contents

Module	Unit	Description	Hours
1.0		**NUMBER SYSTEMS AND OPERATIONS**	6
	1.1	Types of Number Systems: Binary, Decimal — Conversions between Number Systems — Basic Arithmetic operations in Decimal, Binary — Bitwise Operators: AND, OR, NOT, XOR, NAND, NOR, Shift Operations — Applications of Bitwise Operators in Optimization (Checking if a number is even or odd, Swapping two numbers without temporary storage, Counting the number of set bits in a number).	
	1.2	Representation of Signed & Unsigned Integers — Range of Signed and Unsigned Integers — Sign Magnitude form — 1's complement representation — 2's complement representation — Conversions between representations — Comparison of Sign Magnitude, 1's complement, 2's complement methods — Overflow, Underflow, and Wrap Around.	
2.0		**ARITHMETIC AND MODULAR MATHEMATICS**	6
	2.1	GCD and LCM, Euclidean Theorem for GCD (Subtraction Method and Modulo Method), Relation between GCD and LCM.	
	2.2	Modular Arithmetic and Properties, Multiplicative Inverse, Extended Euclidean Algorithm (without proof).	
3.0		**COMBINATORICS AND PROBABILITY**	6
	3.1	Basic Principles of Counting, Permutations: Arrangement with and Without Repetition, Circular Permutation, Applications in real-world scenarios: Scheduling tasks, Cryptography (AES Encryption), Combinations: With and Without Replacement.	
	3.2	Introduction to Probability: Basic concepts, Terminology, Rules, Simple Probability and Conditional Probability, Bayes Theorem, Real-life Applications: Medical Diagnostics and Spam Email Classification, Expected Value.	
4.0		**MATRICES AND SERIES**	6
	4.1	Matrix Introduction — Matrix Addition and Multiplication and their Properties — Determinants: Cofactor Expansion Method — Properties of Determinants — Adjoint and Inverse of a Matrix — Properties of Adjoint and Inverse — Matrix Application: Intro to Graphs, Directed and Undirected Graphs — Adjacency Matrix — Path finding using Matrix Multiplication.	
	4.2	Linear Transformations — Types of Linear Transformations — Arithmetic Progression — Geometric Progression — Harmonic Progression — Relation between Arithmetic Mean, Geometric Mean, and Harmonic Mean — AGP Series (including nth term, sum of n terms in finite and infinite conditions).	
5.0		**SETS AND BINARY OPERATIONS**	6
	5.1	Fundamentals of Set Theory, Set Operations: Union, Difference, Intersection, Complement, Venn Diagrams and Applications, Cartesian Products.	
	5.2	Relation, Types of Relations (Reflexive, Symmetric, Transitive, Antisymmetric, and Partial Order Relation), Functions, Functions Terminology (Domain, Range, Codomain), Types of Functions (Injective, Surjective, and Bijective).	

Text Books

1. Kenneth H. Rosen, Discrete Mathematics and Its Applications, 8th Edition, McGraw-Hill, 2019.
2. Susanna S. Epp, Discrete Mathematics with Applications, 5th Edition, Cengage, 2021.
3. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, 5th Edition, Pearson, 2023.

Reference Books

1. Kenneth H. Rosen & Dr Kamala Krithivasan, Discrete Mathematics and Its Applications, 8th Edition, McGraw-Hill.
2. Richard Johnsonbaugh, Discrete Mathematics, 8th Edition, Pearson.
3. Bernard Kolman, Robert Busby, and Sharon C. Ross, Discrete Mathematical Structures, 6th Edition, Pearson.

Internal Assessment (T1, T2 and FET)

1. T2 should be based on first to fourth modules for 10 marks.
2. Fifth module will be assessed for 5 marks separately.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 7 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC103	Introduction to Generative AI	1	2	-	-	2	-

Course Code	Course name	Evaluation scheme (In semester)					Semester End Exam(SEE)		Total (Marks)
		T1	T2	FET	Term Work	Total	Marks	Min Pass	
UADPC103	Introduction to Generative AI	-	-	-	50	50	-	40%	50

INTRODUCTION TO GENERATIVE AI

Course Description

This course introduces the foundations of Generative AI and its applications across industries. Students explore large language models, AI tools, and frameworks used for text, image, and audio generation. The course covers prompt engineering techniques and AI workflow automation using tools such as n8n and Gemini. Students learn to build AI-powered applications and autonomous AI agents with tool integrations. Advanced topics include memory mechanisms, Model Context Protocol, and modern AI development workflows.

Course Objectives

1. To introduce the fundamental concepts and applications of Generative AI.
2. To develop skills in prompt engineering and AI workflow automation.
3. To explore AI-based content creation using image and audio generation tools.
4. To design and implement AI agents with external tool integration.
5. To understand advanced concepts such as memory mechanisms and Model Context Protocol (MCP) in AI systems.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Identify**[1] the core concepts of Generative AI, its role across industries, and the landscape of available LLMs, AI tools, and frameworks.

CO2 **Describe**[2] prompt engineering techniques, including zero-shot, one-shot, and few-shot learning, and explain AI workflow automation using tools such as n8n, Gemini, and SerpAPI.

CO3 **Apply**[3] AI-based content creation tools to generate multimedia content, including images using Stable Diffusion and audio using text-to-speech synthesis tools.

CO4 **Analyze**[4] the architecture and functioning of autonomous AI agents, including LLM, memory, and tool components, to evaluate their role in task automation using the ReAct framework.

CO5 **Evaluate**[5] and manage advanced AI agents using memory mechanisms, Model Context Protocol (MCP), and Composio for seamless tool integration and context retention.

Course Contents

Module	Unit	Description	Hours
1.0		**FOUNDATIONS OF GENERATIVE AI**	9
	1.1	Generative AI Roadmap: Introduction and Significance, Role of AI in Various Industries, Tools and Frameworks Overview, Gen AI in Action: Text Generation, Image Creation, Audio Synthesis, Video Transformation.	
	1.2	Exploring Gen AI Capabilities: Research, Content Creation, Communication, Decision-Making, Open-source vs Proprietary Models, Analyzing LLMs: GPT, Gemini, Claude, Gamma, Otter ai, Napkin ai, CodeToTutorial, Quizgeco, Perplexity, Hugging Face Ecosystem, AI Job Landscape and Career Paths.	
2.0		**PROMPT ENGINEERING & AI WORKFLOWS**	9
	2.1	Prompt Engineering Fundamentals: Crafting Effective Prompts, Zero-Shot, One-Shot, Few-Shot Learning Techniques, Advanced Prompt Engineering: Optimization Techniques, Complex Prompt Structuring, Creative and Technical Problem-Solving.	
	2.2	AI Workflows, n8n nodes and connections, Building Social Media Content Automation Workflow, Content Creation Automation, Multi-Platform Posting (LinkedIn, Instagram, Twitter), AI-Powered Workflows using Gemini, RSS Feeds and API Integration, Real-Time Summaries, External Data Sources (SerpAPI), LLM limitations: hallucination, passive.	
3.0		**AI-POWERED CONTENT CREATION AND BUILDING NO-CODE APPS**	9
	3.1	Image Generation Applications: Use Cases, Tools: GPT-4o, Imagen 3, Midjourney, Stable Diffusion 3.5, Kaggle Platform.	
	3.2	Audio Generation Applications: Use Cases, Audio Synthesis, Tools: ElevenLabs, Suno, Murf AI, PlayHT, Text-to-Speech, Speech-to-Text models, AI-Driven Podcasts, Lovable, n8n webhook.	
4.0		**AI AGENTS AND AUTOMATION**	9
	4.1	Introduction to AI Agents, LLM, Memory, Tool, Autonomous Decision-Making, Task Completion without Human Intervention, ReAct Framework.	
	4.2	Building a Learning Path Generator: Personalized Learning Path Creation, SerpAPI, Google Calendar Integration for Scheduling, Text-Based Shopping Assistant, Amazon Product Scraping, Telegram Bot Integration, Voice Input Enhancement, Speech-to-Text Conversion.	
5.0		**ADVANCED AI AGENT DEVELOPMENT & MEMORY**	9
	5.1	Importance of Memory in AI Agents, Short-Term and Long-Term Memory Implementation, Context Window, Short-term memory limitations, Personalized User Interactions, Context Retention.	
	5.2	Introduction to Model Context Protocol (MCP): Standardizing AI-Tool Interactions, MCP Server Setup, composio, n8n client node.	

Text Books

1. Bernard Marr, Generative AI in Practice, Wiley, 2024.
2. Tom Taulli, Generative AI: How ChatGPT and Other AI Tools Will Revolutionize Business, Apress, 2023.
3. Chip Huyen, AI Engineering: Building Applications with Foundation Models, O'Reilly Media, 2024.

Reference Books

1. Andriy Burkov, The Hundred-Page Machine Learning Book.
2. Ian Goodfellow, Yoshua Bengio & Aaron Courville, Deep Learning, MIT Press.
3. Russell & Norvig, Artificial Intelligence: A Modern Approach, Pearson.

Internal Assessment (T1, T2 and FET)

1. T1 (Test 1) should be based on first two modules and T2 (Test 2) should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 50 marks comprised of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADAE103	Quantitative Aptitude	3	-	-	3	-	-

Course Code	Course Name	Evaluation scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min pass	
UADAE103	Quantitative Aptitude	10	10	5	-	25	50	40%	75

QUANTITATIVE APTITUDE

Course Description

This course helps students build strong basic math and problem-solving skills by covering key topics like number systems, divisibility rules, factors, multiples, LCM, and GCD. It also introduces percentages and their real-life uses such as increases, decreases, and comparisons. Students will learn practical concepts like profit and loss, discounts, and simple and compound interest, which are useful in everyday financial situations. The course further explains ratios and proportions and their applications in problems invol ving mixtures, partnerships, and ages, helping students improve logical thinking, speed, and accuracy for exams and real-life problem-solving.

Course Objectives

1. To build a strong foundation in number systems, including divisibility rules, co-primes, and numerical patterns.
2. To develop problem-solving skills using factors, multiples, LCM, and GCD concepts.
3. To apply percentage concepts for solving real-life problems involving increase, decrease, and comparisons.
4. To understand and solve problems related to profit and loss, discounts, and simple and compound interest.
5. To strengthen analytical thinking using ratios, proportions, mixtures, partnerships, and age-based problems.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] the classification of numbers, divisibility rules, co-prime concepts, power cycles, and remainder cycle patterns.

CO2 **Describe**[2] the properties and applications of multiples, factors, LCM, and GCD in solving structured numerical problems.

CO3 **Apply**[3] percentage concepts, conversions, and financial mathematics to solve problems involving profit and loss, discounts, and interest.

CO4 **Analyze**[4] ratios, proportions, mixtures, allegations, and partnership concepts to solve practical quantitative problems.

CO5 **Evaluate**[5] solutions to problems involving time, speed, and distance for accuracy and efficiency.

Course Contents

Module	Unit	Description	Hours
1.0		**NUMBER SYSTEMS — FUNDAMENTALS**	9
	1.1	Number Systems: Classification of numbers, divisibility rules, concept of co-primes, power cycles.	
	1.2	Remainder cycles, trailing zeros, and complex numerical problems.	
2.0		**NUMBER SYSTEMS — APPLICATIONS**	9
	2.1	Applications: Understanding multiples, factors, least common multiple (LCM).	
	2.2	Greatest common divisors (GCD) along with their properties and applications.	
3.0		**PERCENTAGES AND ITS APPLICATIONS**	9
	3.1	Percentages: Percentage formula, conversion between decimals and fractions, percentage increase and decrease.	
	3.2	Applications of percentage increase and decrease in real-life scenarios.	
4.0		**PROFIT, LOSS & INTEREST APPLICATIONS**	9
	4.1	Solving problems related to Profit, Loss, Discounts.	
	4.2	Simple Interest and Compound Interest in business and financial scenarios.	
5.0		**RATIOS, PROPORTIONS & ITS APPLICATIONS**	9
	5.1	Ratios & Proportions: Understanding ratio concepts, direct and inverse proportion, and their mathematical applications.	
	5.2	Applications: Mixtures and allegations, partnership-based profit sharing, age-based problems.	

Text Books

1. A. Sharma, How to Prepare for Quantitative Aptitude for the CAT, McGraw-Hill Education, Latest Edition, 2025.
2. R. S. Aggarwal, Quantitative Aptitude for Competitive Examinations, S. Chand Publishing, Latest Edition, 2024.

Reference Books

1. R. S. Aggarwal, A Modern Approach to Logical Reasoning, S. Chand Publishing, Latest Edition.
2. Nishit K. Sinha, Logical Reasoning and Data Interpretation for the CAT, Pearson Education, Latest Edition.
3. Abhijit Guha, Quantitative Aptitude for All Competitive Examinations, McGraw-Hill Education, Latest Edition.

Internal Assessment (T1, T2 and FET)

1. T2 should be based on first to fourth modules for 10 marks.
2. Fifth module will be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 50 marks comprised of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UHSAE006	Communication English Foundation	1	2	-	-	2	-

Course Code	Course name	Evaluation scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
UHSAE006	Communication English Foundation	-	-	-	50	50	-	40%	50

COMMUNICATION ENGLISH FOUNDATION

Course Description

This course focuses on building foundational English communication skills required for academic and professional environments . Students learn sentence formation, grammar usage, and vocabulary development. The course includes practice in speaking, listening, reading, and writing through structured activities. Topics such as descriptions, comparisons, email writing, and paragraph writing are introduced to improve communication clarity. The course also aligns language skills with Cambridge B1 level proficiency standards.

Course Objectives

1. To develop the ability to construct grammatically correct sentences for effective communication.
2. To improve speaking, listening, reading, and writing skills for academic and professional contexts.
3. To build vocabulary and grammar skills required for clear and confident communication.
4. To enhance the ability to describe, interpret, and present information effectively.
5. To prepare students for language proficiency aligned with the Cambridge B1 level.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] foundational grammar rules including sentence formation, noun-verb agreement, countable and uncountable nouns, possessive forms, and pronoun usage.

CO2 **Describe**[2] information clearly using adjectives, degrees of comparison, articles, and descriptive sentence structures for spoken and written communication.

CO3 **Apply**[3] correct grammar and vocabulary, including prepositions, question forms, adverbs, and contextual grammar, to communicate accurately in conversations.

CO4 **Analyze**[4] language patterns using conjunctions, connected speech, pronunciation cues, and email writing conventions to evaluate their role in coherent expression.

CO5 **Evaluate**[5] language tasks aligned with the Cambridge B1 level to assess readiness for academic and professional English proficiency standards.

Course Contents

Module	Unit	Description	Hours
1.0		**BUILDING ACCURACY AND SENTENCE STRUCTURE**	9
	1.1	Sentence formation for clarity, noun-verb agreement for grammatical precision.	
	1.2	Countable and uncountable nouns for accurate expression, possessive forms for ownership and relationships, pronoun use for cohesion and fluency.	
2.0		**DEVELOPING DESCRIPTION AND EXPRESSION**	9
	2.1	Adjectives for detailed description, degrees of comparison for contrast, articles for general and specific reference.	
	2.2	Descriptive sentences for clear communication, real-time speaking practice for confident expression, abstract writing for concise expression, writing short professional messages for effective workplace communication.	
3.0		**EXPRESSING RELATIONSHIPS AND MEANING**	9
	3.1	Prepositions for time, place, and direction, question forms for interaction.	
	3.2	Adverbs for accuracy in action description, contextual grammar for natural communication, integrated tasks for applied language use, interpreting visual information for clear data presentation.	
4.0		**LINKING IDEAS AND EXPRESSING FEELINGS**	9
	4.1	Conjunctions for logical flow, contractions and interjections for natural tone.	
	4.2	Connected speech for fluency, pronunciation practice for clear delivery, discussions for expressive confidence, email writing.	
5.0		**APPLYING TENSES IN REAL COMMUNICATION**	9
	5.1	Present tense for facts and routines, past tense for completed actions, tense consistency for grammatical accuracy.	
	5.2	Scenario-based speaking and writing for real-life application, reflection and review for skill consolidation, paragraph writing.	

Text Books

1. Raymond Murphy, English Grammar in Use, 5th Edition, Cambridge University Press, 2019.
2. Michael Swan, Practical English Usage, 4th Edition, Oxford University Press, 2022.

Reference Books

1. Carter, R. and McCarthy, M., Cambridge Grammar of English: A Comprehensive Guide, Cambridge University Press, 2015.
2. Yule, G., Oxford Practice Grammar, Oxford University Press, 2017.
3. Hewings, M., Thaine, C. and McCarthy, M., Cambridge Academic English: An Integrated Skills Course for EAP, Cambridge University Press.

Internal Assessment (T1, T2 and FET)

1. T1 (Test 1) should be based on first two modules and T2 (Test 2) should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 50 marks comprised of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC108	Web Application Development Laboratory	-	4	-	-	2	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min Pass	Marks	Min Pass	Total (Marks)
UADPC108	Web Application Development I Laboratory	25	40%	25	40%	50

WEB APPLICATION DEVELOPMENT I LABORATORY

Course Outcomes:

After the end of this course, students will be able to:

CO1 **Apply**[3] HTML5 elements, CSS properties, selectors, and the CSS Box Model to build and style structured and visually formatted web pages.

CO2 **Design**[6] and develop responsive web layouts using CSS Flexbox, Grid, positioning techniques, media queries, and multimedia elements to create modern web pages and portfolio websites.

Suggested List of Experiments

1. Build a basic HTML webpage using headings, paragraphs, lists, and hyperlinks.
2. Style a webpage using CSS text properties, colors (Hex codes), and background images.
3. Design a card layout demonstrating the CSS Box Model with border, padding, and margin.
4. Build an HTML5 webpage using semantic elements, multimedia (video/audio), and new input types.
5. Create a styled page using CSS selectors (class, id, pseudo-classes, combinators) and specificity rules.
6. Build a responsive product listing page using CSS Flexbox properties.
7. Design a magazine-style layout using CSS Grid with template areas, fr unit, and minmax.
8. Build a webpage with a sticky header and fixed sidebar using CSS positioning techniques.
9. Develop a responsive webpage that adapts to desktop, tablet, and mobile using media queries.
10. Build a complete responsive portfolio website using HTML5, CSS Flexbox, Grid, and GenAI tools.
11. Create a webpage demonstrating HTML structure with header, navigation, section, and footer elements using semantic HTML.
12. Create a webpage demonstrating CSS font properties including font-family, font-size, font-style, and text-decoration.
13. Create a webpage where images act as hyperlinks and allow navigation within the same document using anchor links.
14. Design a webpage demonstrating Flexbox alignment properties such as justify-content, align-items, flex-wrap, and order.
15. Create a responsive webpage layout using CSS Grid and media queries that adapts to multiple screen sizes.

Reference Books

1. Eric A. Meyer & Estelle Weyl, CSS: The Definitive Guide, O'Reilly Media.
2. Keith Grant, CSS in Depth, Manning Publications.
3. Ben Frain, Responsive Web Design with HTML5 and CSS, 4th Edition, Packt, 2022.

In Semester Evaluation Scheme

1. FEP assessment must be based on the overall performance of the student with every assignment/practical graded from time to time.
2. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC105	Computer Programming Laboratory	-	4	-	-	2	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min Pass	Marks	Min Pass	Total (Marks)
UADPC105	Computer Programming Laboratory	25	40%	25	40%	50

COMPUTER PROGRAMMING LABORATORY

Course Outcomes:

After the end of this course, students will be able to:

CO1 **Apply**[3] Python programming concepts using variables, data types, operators, conditional statements, loops, and control statements to solve basic computational problems.

CO2 **Develop**[6] Python programs using functions and data structures such as lists, tuples, sets, and dictionaries to manage and process structured data.

Suggested List of Experiments

1. Write a Python program to perform arithmetic operations and demonstrate operator precedence.
2. Create a program that accepts user input and performs type conversion between string and integer.
3. Implement a grade calculator using if-elif-else conditional branching.
4. Build a multiplication table generator using while and for loops with nested loop support.
5. Write a program using break, continue, and pass to control loop execution.
6. Create a program to perform list operations: creation, slicing, modification, and joining.
7. Implement a tuple-based student record system using tuple methods.
8. Define functions with keyword, positional, and default arguments to compute simple statistics.
9. Write a program to perform set operations: union, intersection, difference, and symmetric difference.
10. Build a dictionary-based contact book with add, search, and delete functionality.
11. Write a Python program to read from and write to text files using file handling methods.
12. Create a Python program to handle exceptions using try-except-finally blocks.
13. Implement a simple class with attributes and methods to demonstrate object-oriented programming.
14. Write a program using list comprehensions and lambda functions for data processing.
15. Build a Python program using standard library modules such as math, random, and datetime.

Reference Books

1. Mark Lutz, Learning Python, O'Reilly Media.
2. Al Sweigart, Automate the Boring Stuff with Python, No Starch Press.
3. Brett Slatkin, Effective Python, Addison-Wesley.

In Semester Evaluation Scheme

1. FEP assessment must be based on the overall performance of the student with every assignment/practical graded from time to time.
2. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Programme Structure R26 for B.Tech.: First Year Artificial Intelligence and Data Science

Semester II: Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total Credits
UADPC201	Web Application Development II	3	-	-	3	-	-	3
UADPC202	Database Management System	3	-	-	3	-	-	3
UADPC203	Data Structures	4	-	-	4	-	-	4
UADPC204	Building LLM Applications	1	2	-	-	2	-	2
UADAE201	Numerical Ability	2	-	-	2	-	-	2
UHSAE007	Communicative English Advanced	1	2	-	-	2	-	2
UADPC205	Database Management Systems Laboratory	-	2	-	-	1	-	1
UADPC206	Web Application Development II Laboratory	-	2	-	-	1	-	1
UADPC207	Data Structures Laboratory	-	4	-	-	2	-	2
UHSVE001 **OR** USTBS206	Constitution of India **OR** Modern Physics	2	-	-	2	-	-	2
Total		**16**	**12**	**-**	**14**	**8**	**-**	**22**

Semester II: Evaluation Scheme

Course Code	Course Name	Theory Marks					Practical Marks		
		T1	T2	FET	Total	SEE	TermWork/ tutorial	Practical Oral/POE	Total
UADPC201	Web Application Development II	10	10	5	25	50	-	-	75
UADPC208	Database Management System	10	10	5	25	50	-	-	75
UADPC202	Foundation of Data Structures	10	10	5	25	75	-	-	100
UADPC204	Building LLM Applications	-	-	-	-	-	50		50
UADAE201	Numerical Ability	-	10	5	15	35	-	-	50
UHSAE007	Communicative English Advanced	-	-	-	-	-	50	-	50
UADPC210	Database Management Systems Laboratory	-	-	-	-	-	25	-	25
UADPC209	Web Application Development II Laboratory	-	-	-	-	-	25	-	25
UADPC207	Foundation of Data Structures Laboratory	-	-	-	-	-	25	25	50
UHSVE001 **OR** USTBS206	Constitution of India **OR** Modern Physics	-	10	5	15	35	-	-	50
								Total	**550**

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC201	Web Application Development II	3	-	-	3	-	-

Course Code	Course name	Evaluation scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
UADPC201	Web Application Development II	10	10	5	-	25	50	40%	75

WEB APPLICATION DEVELOPMENT II

Course Description

This course focuses on developing interactive web applications using JavaScript and modern frontend programming techniques. Students learn core JavaScript concepts such as variables, arrays, functions, objects, and DOM manipulation to create dynamic web pages. The course also introduces event handling, HTTP requests, JSON data processing, and form-based web applications. Advanced topics include modern JavaScript features, asynchronous programming with Promises and async/await, and class-based object-oriented programming. Students further explore Node.js, ES6 modules, and npm packages to run and manage JavaScript applications and build full-stack solutions.

Pre-requisites: Web Application Development I

Course Objectives

1. To understand JavaScript basics, DOM manipulation, and event handling for building interactive web pages.
2. To use arrays, functions, objects, and schedulers to write structured and reusable programs.
3. To make HTTP requests, handle JSON data, and build dynamic web applications using forms and APIs.
4. To apply modern JavaScript concepts, classes, and asynchronous programming using Promises and async/await.
5. To work with Node.js, ES6 modules, and npm packages to run and manage JavaScript applications.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Identify**[1] core JavaScript concepts including variables, data types, DOM structure, conditional statements, and event listener mechanisms.

CO2 **Describe**[2] arrays, functions (arrow and callback functions), objects, object destructuring, and scheduler functions used to build structured web features.

CO3 **Apply**[3] HTTP requests using Fetch API, JSON data handling, and form elements to develop dynamic and interactive web applications.

CO4 **Analyze**[4] modern JavaScript concepts including spread and rest operators, template literals, constructor functions, and JavaScript scoping to evaluate program behavior.

CO5 **Create**[6] classes, inheritance, Promises, async/await, Node.js modules, and npm packages to develop structured full-stack JavaScript applications.

Course Contents

Module	Unit	Description	Hours
1.0		**INTRODUCTION TO JAVASCRIPT & DOM FUNDAMENTALS**	9
	1.1	Dynamic Web Applications & JavaScript Basics: Introduction to JavaScript, Writing First JavaScript Program, Console, JavaScript Variables, Declaring Variables, Assigning Values, Software Compatibility.	
	1.2	DOM: DOM & HTML DOM, DOM Tree, Manipulating HTML and CSS, DOM Methods, DOM Manipulations, Adding Event Listeners Dynamically, Events: onclick Event, Event Listeners, Event Object, Keyboard Events, Primitive Types: Data Types, typeof, Converting String into Number, Conditional Statements & Operators: Conditional Statements, Comparison Operators, Logical Operators, Ternary Operator.	
2.0		**ARRAYS, FUNCTIONS & OBJECTS**	9
	2.1	Math Functions & Input Elements: Random Number, Math Functions, HTML Input Element, Arrays: Creating Arrays, Accessing and Modifying Arrays, Array Length, Array Methods, for of loop, Functions: Function Declaration, Function Expression, Arrow Functions, Callback Function, Default Parameters.	
	2.2	Objects: Creating Objects, Identifiers, Accessing Object Properties, Object Destructuring, Modifying Objects, Property Values, Schedulers: null as value, Scheduler Functions.	
3.0		**HTTP, FORMS & DYNAMIC APPLICATIONS**	9
	3.1	HTTP Basics: Protocols, HTTP Request, HTTP Response, HTTP Requests using JavaScript: Making HTTP Requests, HTTP Request Methods, JSON Methods.	
	3.2	Web Applications: Wikipedia Search Application, Input Element Search, Static Spinner, classList, Forms: Form Element, Label Element, Event Methods, Form Events, Select Element, Radio Buttons, Leveraging GenAI: GenAI for Debugging & Building, Building a Dynamic Website using Gen AI.	
4.0		**MODERN JAVASCRIPT & ADVANCED CONCEPTS**	9
	4.1	Modern JS Concepts: Spread Operators, Rest Parameter, Template Literals.	
	4.2	Functions: Creating Multiple Objects, Factory Functions, Constructor Functions, Object Property Shorthand, Built-in Constructor Functions, this keyword, More JS Concepts: JS Mutable and Immutable Values, Variable Declaration.	
5.0		**CLASSES, PROMISES & NODE.JS**	9
	5.1	JS Classes: Creating Objects using Classes, Constructor, Inheritance, JS Promises: Asynchronous and Synchronous, Promise States, Consuming Promises, Creating Promises, Async/Await.	
	5.2	String Manipulations & Array Methods: String Methods, Advanced Array Methods, More JS Concepts: Scoping, Hoisting, JS Modules & Node Packages: MERN Stack, Node JS, Running JavaScript, Modules, Module Exports, Core Modules, Node Package Manager, Third-Party Packages.	

Text Books

1. Jon Duckett, JavaScript and jQuery: Interactive Front-End Web Development (Updated Edition), Wiley, 2023.
2. Marijn Haverbeke, Eloquent JavaScript, 4th Edition, No Starch Press, 2024.
3. Kyle Simpson, You Don't Know JS Yet: Get Started, 2nd Edition, O'Reilly, 2023.

Reference Books

1. Ethan Brown, Learning JavaScript, 3rd Edition, O'Reilly, 2023.
2. Addy Osmani, Learning JavaScript Design Patterns, 2nd Edition, O'Reilly, 2023.
3. Azat Mardan, Node.js in Action, 2nd Edition Updated, Manning, 2023.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. Fifth module will be assessed for 5 marks separately.
3. Tutorial should be assessed based on accuracy, completeness, conceptual understanding, problem-solving approach, clarity of explanation, and presentation quality.
4. Term work assessment shall be based on the overall performance of the student with every tutorial graded from time to time.
5. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC202	Database Management System	3	-	-	3	-	-

Course Code	Course name	Evaluation Scheme (In semester)					Semester End Exam (SEE)		Total (Mark)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
UADPC202	Database Management System	10	10	5	-	25	50	40%	75

DATABASE MANAGEMENT SYSTEM

Course Description

This course introduces the fundamental concepts of database systems and data management using SQL and NoSQL technologies. Students learn how to design and manage relational databases using SQL queries, constraints, and data manipulation operations. The cou rse covers advanced database concepts such as joins, subqueries, aggregation, and database normalization for efficient data organization. Students also explore transaction management, ACID properties, and concurrency control to ensure data integrity and reliability. Additionally, the course introduces NoSQL databases and MongoDB, enabling students to work with flexible, document-based data models for modern applications.

Pre-requisites: Computer Programming

Course Objectives

1. To introduce students to basic database concepts and SQL so they can create tables and manage data using simple SQL commands.
2. To help students write SQL queries to filter, sort, group, and process data using operators, functions, and set operations.
3. To enable students to model relational databases and perform advanced querying using joins, views, and subqueries.
4. To teach database normalization, transactions, and ACID properties to ensure data accuracy and reliability.
5. To provide an introduction to NoSQL concepts and MongoDB for storing and querying flexible, non-relational data formats.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] fundamental database concepts including DBMS types, relational and non-relational databases, SQL data types, constraints, and basic DDL and DML commands.

CO2 **Describe**[2] SQL querying mechanisms including comparison operators, pattern matching, ordering, aggregation functions, set operations, and CASE expressions.

CO3 **Apply**[3] ER modeling concepts to design database schemas and construct SQL queries using joins, views, aliases, and subqueries for efficient data retrieval.

CO4 **Analyze**[4] normalization techniques (1NF, 2NF, 3NF, and BCNF), functional dependencies, and transaction management concepts including ACID properties and concurrency control.

CO5 **Evaluate**[5] NoSQL database concepts and perform CRUD operations in MongoDB to assess when document-based models are preferable to traditional SQL databases.

Course Contents

Module	Unit	Description	Hours
1.0		**INTRODUCTION TO DATABASE SYSTEMS AND SQL FUNDAMENTALS**	9
	1.1	Database Fundamentals: DBMS Types, Relational vs Non-Relational Databases, Introduction to Database covering Data and Database Concepts, DBMS, Advantages of DBMS, and Types of Databases, SQL Basics including Data Types, Constraints and Schemas, Table Creation, Modification and Deletion.	
	1.2	Data Manipulation covering INSERT, UPDATE, DELETE operations, and WHERE Clause Instances.	
2.0		**QUERYING WITH SQL**	9
	2.1	Comparison and Logical Operators, Pattern Matching with LIKE, Operator Precedence, IN and BETWEEN Operation, Result Ordering using ORDER BY, LIMIT and OFFSET clauses.	
	2.2	Aggregation Functions, GROUP BY Implementation, HAVING Usage, Set Operations including UNION, INTERSECT, and EXCEPT, SQL Functions, CASE Expressions, and Expression Usage in Queries.	
3.0		**MODELLING DATABASE AND QUERY OPERATIONS**	9
	3.1	Database Modelling and ER Modelling Concepts.	
	3.2	SQL Joins covering Join Types including Natural Join, Inner Join, Left Join, Right Join, Self Join, and Cross Join, Alias Usage, View Creation and Management, and Subqueries.	
4.0		**DATABASE NORMALIZATION AND TRANSACTION PROCESSING**	9
	4.1	Schema Refinement and Functional Dependencies, Normalization covering 1NF, 2NF, 3NF, and Boyce-Codd Normal Form, Denormalization concepts.	
	4.2	Transaction Management, ACID Properties, and Concurrency Control.	
5.0		**INTRODUCTION TO NOSQL DATABASES AND MONGODB**	9
	5.1	NoSQL Database Concepts, Collections vs Tables, Documents vs Records, Fields, Schema Flexibility, Setting up MongoDB and Creating a Database.	
	5.2	CRUD Operations in MongoDB, Querying: Projection, Conditional Operators, Ordering Results, Pagination, Logical Operators, Nested Documents, Array Fields, Basic Array Queries, Array Operators, $elemMatch, Grouping & Aggregate functions, Comparison of SQL vs NoSQL.	

Text Books

1. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, 4th Edition, McGraw-Hill, 2022.
2. Abraham Silberschatz, Henry Korth, S. Sudarshan, Database System Concepts, 8th Edition, McGraw-Hill, 2023.
3. Carlos Coronel & Steven Morris, Database Systems: Design, Implementation & Management, 14th Edition, Cengage, 2022.

Reference Books

1. Peter Rob & Carlos Coronel, Database Principles: Fundamentals of Design, Implementation and Management, Cengage, 2022.
2. Pramod J. Sadalage & Martin Fowler, NoSQL Distilled (Updated Edition), Addison-Wesley, 2022.
3. Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2016.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. Fifth module will be assessed for 5 marks separately.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC208	Foundation of Data Structures	4	-	-	4	-	-

Course Code	Course name	Evaluation scheme (In semester)					Semester end exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min pass	
UADPC208	Foundation of Data Structures	10	10	5	-	25	75	40%	100

FOUNDATION OF DATA STRUCTURES

Course Description

This course introduces the fundamentals of programming and problem-solving using C++ and basic data structures. Students learn programming constructs such as operators, control structures, functions, and STL containers. The course develops algorithmic thinking through complexity analysis, arrays, recursion, and string manipulation techniques. It also covers classical sorting and searching algorithms us ed for efficient data processing. In addition, students explore bit manipulation and number system concepts for low-level computational problem solving.

Course Objectives

1. To help students understand the basics of C++ programming and introduce algorithm analysis for writing efficient code.
2. To develop problem-solving skills using mathematical logic and array-based techniques for real-world computational tasks.
3. To enable students to work with recursion, backtracking, and string manipulation to solve complex programming challenges.
4. To teach students various sorting and searching techniques and apply them to optimize data access and processing.
5. To build the ability to understand number systems and apply bit manipulation techniques for efficient low-level problem-solving.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] fundamental C++ programming constructs including operators, type conversion, conditional statements, loops, functions, and key STL containers.

CO2 **Describe**[2] algorithm analysis using asymptotic notations, time and space complexity, and mathematical foundations including digit operations, divisors, prime numbers, GCD, and LCM.

CO3 **Apply**[3] array representations and traversal techniques to solve problems involving largest elements, duplicate removal, and zero movement.

CO4 **Analyze**[4] recursive and backtracking solutions for problems such as factorial computation, Fibonacci series generation, subset generation, and string manipulation tasks including anagram and isomorphic string detection.

CO5 **Evaluate**[5] sorting, searching, and bit manipulation techniques to determine the most efficient approach for organizing and retrieving data in real-world scenarios.

Course Contents

Module	Unit	Description	Hours
1.0		**PROGRAMMING & COMPUTATIONAL FOUNDATIONS**	9
	1.1	C++ Basics covering Operators, Type Conversion, Conditionals, Loops, and Functions.	
	1.2	C++ STL including Pair, Vector, Deque, Stack, Queue, Priority Queue, Sets, Maps, and Utility Functions, Algorithms and Data Structures Introduction, Asymptotic Notations, Time and Space Complexity, and Pseudo Code.	
2.0		**MATHEMATICAL FOUNDATIONS & ARRAYS**	9
	2.1	Mathematical Foundations covering Digit Operations, Divisors, Prime Numbers, GCD and LCM.	
	2.2	Linear Arrays including Representation, Traversal, and Linear Search, Array Problems covering Largest/Second Largest, Maximum Consecutive Ones, Removing Duplicates, and Moving Zeros to End.	
3.0		**RECURSION & STRINGS**	9
	3.1	Recursion covering Factorial, Fibonacci Series, and Array Problems including Reverse Array and Sum of N Numbers, Printing 1 to N and N to 1.	
	3.2	Backtracking including Subsets Generation, Combinations, and All Subsequences, Strings covering Basics and Built-in Methods, Anagram, Reverse Words, and Isomorphic Strings.	
4.0		**SORTING & SEARCHING**	9
	4.1	Sorting including Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, and Quick Sort.	
	4.2	Searching covering Linear Search, Binary Search, and Lower/Upper Bound.	
5.0		**BIT MANIPULATION**	9
	5.1	Number Systems and Conversions, Bitwise Operators including OR, AND, NOT, XOR, and Shifts.	
	5.2	Bit Manipulation Techniques covering Check, Set, Clear, Toggle, Count Set Bits, Swap Numbers, and Remove Rightmost Set Bit.	

Text Books

1. Thomas H. Cormen et al., Introduction to Algorithms, 4th Edition, MIT Press, 2022.
2. Bjarne Stroustrup, Programming: Principles and Practice Using C++, 3rd Edition, Addison-Wesley, 2022.
3. Michael T. Goodrich, Data Structures and Algorithms in C++, 2nd Edition Updated, Wiley, 2023.

Reference Books

1. Aditya Bhargava, Grokking Algorithms, 2nd Edition, Manning, 2023.
2. Steven Skiena, The Algorithm Design Manual, 3rd Edition, Springer, 2022.
3. Scott Meyers, Effective Modern C++ (Updated Edition), O'Reilly, 2022.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. Fifth module will be assessed for 5 marks separately.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 7 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC204	Building LLM Applications	1	2	-	-	2	-

Course Code	Course name	Evaluation scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
UADPC204	Building LLM Applications	-	-	-	50	50	-	40%	50

BUILDING LLM APPLICATIONS

Course Description

This course introduces the design and development of applications using Large Language Models (LLMs). It covers LLM fundamentals, API integration, and advanced techniques such as Retrieval-Augmented Generation (RAG) using vector databases and embeddings. Learners will build AI agents and multi-agent systems using frameworks like LangChain and CrewAI, and explore context engineering for intelligent workflows. The course also focuses on evaluating and deploying LLM applications in local and cloud environments with considerations for performance, cost, and safety.

Course Objectives

1. Explain LLM architecture and integrate models using APIs for real-world applications.
2. Build RAG-based systems using embeddings, vector databases, and retrieval techniques.
3. Design and implement AI agents using LangChain/LangGraph with tools and workflows.
4. Develop multi-agent systems for collaborative task execution..
5. Evaluate and deploy LLM applications considering performance, cost, and safety.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] the fundamentals of LLM architecture, including transformer models, tokenization, embeddings, attention mechanisms, and context window concepts.

CO2 **Describe**[2] RAG architecture, retrieval methodologies (dense, sparse, and hybrid), vector databases, and embedding models used to enhance LLM responses with external knowledge.

CO3 **Apply**[3] LangChain and LangGraph frameworks to build AI agents that utilize tools, prompt templates, chains, and autonomous task workflows.

CO4 **Analyze**[4] multi-agent system architectures, including MCP tools, agent roles, task delegation, and CrewAI or n8n workflows, to evaluate collaborative AI problem-solving approaches.

CO5 **Evaluate**[5] LLM systems using metrics such as correctness, relevance, and safety, and assess deployment strategies including local, cloud, and cost-performance trade-offs.

Course Contents

Module	Unit	Description	Hours
1.0		**LLM FUNDAMENTALS AND INTEGRATION**	9
	1.1	LLM Architecture, Evolution of LLMs, Transformer architecture, Tokenization and embeddings, Attention mechanisms, Context window, LLM Integration using OpenAI/Claude/Gemini APIs, Temperature and other generation settings, Context Management and conversation state management.	
	1.2	Techniques for handling long inputs, Rate limits and costs, Building UI for LLM Applications, Introduction to Gradio, Understanding deployment: local vs cloud, Step-by-step deployment process on Hugging Face Spaces, Google Anti-gravity, Comet AI Browser, etc.	
2.0		**TOOL USE AND RETRIEVAL AUGMENTED GENERATION**	9
	2.1	Intro to Tool/Function Calling, Creating tool functions, Defining tool schemas, Sending tool definitions to LLMs, Handling tool calls, Sending tool results back to LLM, Multi-step reasoning with tools, LangChain setup and workflows.	
	2.2	Tool integration, Introduction to Retrieval Augmented Generation (RAG), RAG Architecture and Components, Retrieval Methodologies - Dense, Sparse, Hybrid, Indexing Essentials, Vector Databases: Introduction and Use Cases, Embedding Models, Creating and Managing Embeddings, Integration with LLM Workflows, Managing Context with RAG.	
3.0		**AI AGENTS AND CONTEXT ENGINEERING**	9
	3.1	Building AI Agents using Langchain/Langgraph, Agent Architecture and Tool Usage, Autonomous Agent Workflows, Task Planning and Execution, Setting Up Prompt Templates, Short-term and Long-term memory, LangGraph fundamentals.	
	3.2	Checkpointer-based persistence, thread_id and context management, Building Conversational AI systems, multi-step workflows, speech integration (STT/TTS), dynamic response generation, RAG agents, tool-based retrieval, hybrid search, Vector DBs (Pinecone, Chroma), Context Engineering principles and optimization techniques.	
4.0		**MODEL CONTEXT PROTOCOL AND MULTI-AGENT SYSTEMS**	9
	4.1	MCP integration with LangChain, MCP adapters and server-based tool access, Integrating MCP in LangChain, Building Custom MCP Tools, Introduction to Multi-Agent Systems, Agent Roles and Task Delegation.	
	4.2	Defining Agent Goals and Collaboration, Collaborative AI and Multi-Agent Frameworks, Building Multi-Agent Solutions Using CrewAI, Key Features of Multi-Agent Systems: Memory, Role-based Reasoning, Process Management, Orchestrating Agent Collaboration and Workflows.	
5.0		**LLM EVALUATION AND LOCAL DEPLOYMENT MODEL**	9
	5.1	Model vs System evaluation, evaluation criteria (Correctness, Relevance, Faithfulness, Safety), challenges in testing LLM-based applications, Evaluation Techniques — Human evaluation, automated metrics (BLEU, BERTScore), LLM-as-a-Judge approaches, evaluation workflows and scoring strategies.	
	5.2	Evaluation Frameworks: DeepEval, test case structuring, bias and consistency considerations, best practices for reliable evaluation, Local Model Deployment: Running models locally (Ollama, LM Studio), model sizes and hardware requirements, integration with applications, trade-offs vs cloud models.	

Text Books

1. Chip Huyen, AI Engineering: Building Applications with Foundation Models, O'Reilly, 2024.
2. Nir Diamant, Prompt Engineering for Generative AI, O'Reilly, 2024.
3. Ben Auffarth, Generative AI with Python and Transformers, Packt, 2023.

Reference Books

1. LangChain Team, LangChain Documentation, 2024.
2. OpenAI, OpenAI API Developer Guide, 2024.
3. Hugging Face, Transformers Documentation, 2024.

Internal Assessment (T1, T2 and FET)

1. T1 (Test 1) should be based on first two modules and T2 (Test 2) should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 50 marks comprised of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADAE201	Numerical Ability	2	-	-	2	-	-

Course Code	Course name	Evaluation scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
UADAE201	Numerical Ability	-	10	5	-	15	35	40%	50

NUMERICAL ABILITY

Course Description

This course develops students' quantitative reasoning and problem-solving skills for real-world applications. It covers topics such as time, speed and distance, time and work, permutations and combinations, probability, and clock and calendar problems. Students will also learn to interpret data from tables and graphs and apply percentages, ratios, and averages for analysis. The course further strengthens logical thinking through ranking-based problems, enabling students to solve numerical problems efficiently.

Course Objectives

1. To understand and solve problems related to time, speed, and distance including trains, races, and average speed.
2. To develop the ability to solve problems related to time and work, efficiency, pipes and cisterns, and wage distribution.
3. To apply counting principles, permutations, combinations, and probability concepts to solve real-life problems.
4. To analyze and solve clock and calendar problems using angle concepts, odd days, and leap year calculations.
5. To build skills in data interpretation and ranking by analyzing tables, graphs, percentages, ratios, and averages.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] work efficiency concepts, pipes and cisterns, and wage distribution principles used in time and work problems.

CO2 **Describe**[2] fundamental counting principles, permutation and combination techniques, and probability rules for solving problems involving coins, dice, and cards.

CO3 **Apply**[3] time, speed, and distance concepts, including average speed, relative speed, and problems related to trains and races.

CO4 **Analyze**[4] data presented in tables, bar graphs, pie charts, and line graphs to calculate percentages, ratios, and averages for comparison and interpretation.

CO5 **Evaluate**[5] ranking-based problems involving positions, order, and arrangements in rows or queues to determine accurate outcomes.

Course Contents

Module	Unit	Description	Hours
1.0		**TIME, SPEED & DISTANCE**	6
	1.1	Time speed & distance: understanding the relation between time speed & distance, unit conversion.	
	1.2	Average speed, relative speed, effective speed, solving the problems related to trains and races and average based problems.	
2.0		**TIME & WORK**	6
	2.1	Time & Work: work efficiency concepts, pipes and cisterns.	
	2.2	Wage distributions, and the main chain rule for determining work completion timelines.	
3.0		**PnC & PROBABILITY**	6
	3.1	Permutations & Combinations: Fundamental counting principle, permutation and combination formulas, ranking-based questions, and circular permutations.	
	3.2	Probability: Solving problems with coins, dice, playing cards, and miscellaneous probability-based scenarios.	
4.0		**CLOCKS AND CALENDARS**	6
	4.1	Clocks: Angle between hands, gaining and losing time, problems on incorrect clocks.	
	4.2	Calendars: Odd days concept, leap year calculations, and calendar repetition.	
5.0		**DATA INTERPRETATION & RANKING**	6
	5.1	Tables and Graphs: Interpretation of data presented in tables, bar graphs, pie charts, and line graphs. Applications: Analyzing trends, calculating percentages, ratios, and averages, and comparing data across different categories.	
	5.2	Ranking: problem with relative and absolute ranking, finding a position in row or queue, highest lowest ranking scenarios.	

Text Books

1. Arun Sharma, Quantitative Aptitude for CAT, Latest Edition, McGraw-Hill, 2023.
2. R. S. Aggarwal, Quantitative Aptitude for Competitive Examinations (Updated Edition), S. Chand, 2022.
3. Sarvesh K. Verma, Quantum CAT Quantitative Aptitude, Arihant, 2022.

Reference Books

1. Abhijit Guha, Quantitative Aptitude for Competitive Exams, McGraw-Hill, 2023.
2. Nishit K. Sinha, Quantitative Aptitude for Competitive Examinations, Pearson, 2023.
3. R. S. Aggarwal, A Modern Approach to Logical Reasoning, S. Chand Publishing, Latest Edition.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. Fifth module will be assessed for 5 marks separately.
3. Term work assessment shall be based on the overall performance of the student with every assignment graded from time to time.
4. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Semester End Examination

1. Question paper will be of 50 marks comprised of 5 questions, each carrying 10 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UHSAE007	Communicative English Advanced	1	2	-	-	2	-

Course Code	Course name	Evaluation Scheme (In semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
UHSAE007	Communicative English Advanced	-	-	-	50	50	-	40%	50

COMMUNICATIVE ENGLISH ADVANCED

Course Description

This course focuses on enhancing advanced English communication skills required for academic, technical, and professional env ironments. Students practice using different tense structures to express future plans, ongoing actions, and past experiences effectively. The course also develops the ability to form questions, participate in discussions, and present ideas clearly in spoken and written communica tion. Emphasis is placed on hypothetical situations, conditional expressions, and storytelling using perfect tenses. Activities such as interviews, presentations, and writing exercises help improve fluency, pronunciation, and confidence in communication.

Pre-requisites: Communication English Foundation

Course Objectives

1. To help students express future plans, predictions, and decisions clearly using appropriate future tense forms in daily communication.
2. To enable students to describe ongoing actions and express ability, permission, obligation, and possibility in real-life speaking and writing situations.
3. To develop the ability to connect past and present experiences using simple, continuous, and perfect tenses for effective storytelling and conversations.
4. To strengthen advanced communication skills by forming clear questions, participating confidently in discussions, and improving spoken and written accuracy.
5. To help students express real and hypothetical situations, regrets, and imagined outcomes while improving overall fluency and pronunciation.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Recall**[1] tense structures used for expressing future plans, predictions, scheduled actions, and spontaneous decisions in spoken and written communication.

CO2 **Describe**[2] ongoing actions using continuous tense structures and explain modal verbs expressing ability, obligation, permission, and possibility in real-life situations.

CO3 **Apply**[3] simple and perfect tense structures to connect past experiences with present relevance, narrate events, and sequence past actions in storytelling and reports.

CO4 **Analyze**[4] question formation strategies, conditional structures, and advanced communication patterns to evaluate clarity and accuracy in discussions, interviews, and presentations.

Course Contents

Module	Unit	Description	Hours
1.0		**EXPRESSING FUTURE INTENTIONS AND PLANS**	6
	1.1	Expressing future plans, predictions, and promises, Describing spontaneous decisions and scheduled actions.	
	1.2	Applying future tense in real-life speaking contexts such as planning and forecasting, Integrating understanding of tense usage to improve grammatical accuracy in communication.	
2.0		**DESCRIBING ONGOING ACTIONS AND EXPRESSING POSSIBILITY**	6
	2.1	Performing live communication tasks using continuous tense structures, Describing ongoing actions and events happening at or around the present, past, or future time.	
	2.2	Differentiating between completed and ongoing actions in real contexts, Expressing ability, obligation, permission, and possibility in communication.	
3.0		**CONNECTING PAST AND PRESENT EXPERIENCES**	6
	3.1	Connecting past experiences with present relevance, Applying both simple and perfect tenses in dialogues, storytelling, and reporting experiences.	
	3.2	Indicating actions completed before another past event, Describing duration of actions before a specific past moment, describing narratives, reflections, and sequencing of past events.	
4.0		**MASTERING ADVANCED COMMUNICATION AND HYPOTHETICAL SCENARIOS**	6
	4.1	Describing actions that will be completed before a specified future moment, Emphasizing the duration of an ongoing future action.	
	4.2	Forming information questions to gather details in conversations, Constructing yes/no and auxiliary-based questions effectively, Participating in live interviews, discussions, and oral presentations, Improving clarity and accuracy in written communication, Enhancing pronunciation and clarity in spoken English, Expressing real and unreal situations, possibilities, regrets, and imagined outcomes in speech and writing.	

Text Books

1. Raymond Murphy, English Grammar in Use, 5th Edition, Cambridge University Press, 2023.
2. Mark Foley & Diane Hall, MyGrammarLab Intermediate (Updated Edition), Pearson, 2023.
3. Martin Hewings, Advanced Grammar in Use, 4th Edition, Cambridge University Press, 2022.

Reference Books

1. Michael McCarthy & Felicity O'Dell, English Vocabulary in Use (Advanced Edition), Cambridge, 2022.
2. Joan Saslow & Allen Ascher, Top Notch Communication Series, Pearson, 2022.
3. Carter, R. and McCarthy, M., Cambridge Grammar of English: A Comprehensive Guide, Cambridge University Press, 2015.

Internal Assessment (T1, T2 and FET)

1. T2 should be based on first to fourth modules for 10 marks.
2. Fifth module will be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 50 marks comprised of 5 questions, each carrying 7 marks.
2. The duration of semester end examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC209	Database Management Systems Laboratory	-	2	-	-	1	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		Total (Marks)
		Term Work	Min Pass	Marks	Min Pass	
UADPC209	Database Management Systems Laboratory	25	40%	-	40%	25

DATABASE MANAGEMENT SYSTEMS LABORATORY

Course Outcomes:

After the end of this course, students will be able to:

CO1 **Apply**[3] DDL commands to create and manage database tables with appropriate integrity constraints for consistent and accurate data storage.

CO2 **Apply**[3] DML statements, including insert, update, delete, and retrieval operations, along with filtering and sorting clauses for effective data management.

CO3 **Apply**[3] scalar and aggregate functions to perform calculations and generate meaningful summaries from stored data.

CO4 **Construct**[4] effective SQL queries using joins, grouping, and subqueries to extract and analyze data from multiple related tables.

CO5 **Implement**[6] triggers for automated data management and design NoSQL databases in MongoDB to store and query non-relational data.

Suggested List of Experiments

1. Create, alter, and drop tables with integrity constraints using DDL commands.
2. Retrieve and modify data from a database using DML statements.
3. Retrieve data from the database using IN, BETWEEN, LIKE, ORDER BY, GROUP BY, and HAVING clauses.
4. Use of scalar and aggregate functions.
5. Retrieve data from a database using Join operations (Inner and Outer Joins).
6. Use sub queries for retrieving data.
7. Use of trigger in data manipulation.
8. Design and develop a NoSQL Database.
9. Create tables with different constraints (Primary key, Foreign key, Unique, Check).
10. SQL queries using logical operators (AND, OR, NOT).
11. Queries using ORDER BY, LIMIT, and OFFSET.
12. Create and use SQL views.
13. ER diagram design and converting to relational schema.
14. Normalization demonstration (1NF, 2NF, 3NF).
15. MongoDB queries on nested documents and arrays.

Text Books

1. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, 4th Edition, McGraw-Hill, 2022.
2. Abraham Silberschatz, Henry Korth, S. Sudarshan, Database System Concepts, 8th Edition, McGraw-Hill, 2023.
3. Carlos Coronel & Steven Morris, Database Systems: Design, Implementation & Management, 14th Edition, Cengage, 2022.

Reference Books

1. Peter Rob & Carlos Coronel, Database Principles: Fundamentals of Design, Implementation and Management, Cengage, 2022.
2. Pramod J. Sadalage & Martin Fowler, NoSQL Distilled (Updated Edition), Addison-Wesley, 2022.
3. Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2016.

In Semester Evaluation Scheme

1. FEP assessment must be based on the overall performance of the student with every assignment/practical graded from time to time.
2. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC210	Web Application Development II Laboratory	-	2	-	-	1	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min Pass	Marks	Min Pass	Total (Marks)
UADPC210	Web Application Development II Laboratory	25	40%	-	40%	25

WEB APPLICATION DEVELOPMENT II LABORATORY

Course Outcomes:

After the end of this course, students will be able to:

CO1 **Apply**[3] JavaScript concepts such as variables, conditional statements, functions, and event handling to develop interactive web pages.

CO2 **Implement**[3] dynamic user interfaces by manipulating the DOM and handling browser events using JavaScript.

CO3 **Design**[6] web applications that process user input through forms and dynamically update content using JavaScript and HTML elements.

CO4 **Build**[6] web applications that retrieve and display data from external APIs using HTTP requests and JSON processing.

CO5 **Apply**[5] modern JavaScript features, including classes, promises, async/await, and Node.js modules, to develop scalable web applications.

Suggested List of Experiments

1. Build a webpage that dynamically changes content and styles using DOM manipulation and event listeners.
2. Create an interactive quiz app using conditional statements, comparison operators, and DOM updates.
3. Implement an array-based shopping list with add, remove, and display features using array methods.
4. Build a random quote generator using functions, arrow functions, and Math.random().
5. Create a dynamic form with validation using form events, select elements, and radio buttons.
6. Fetch data from a public API and display results dynamically using HTTP requests and JSON methods.
7. Build a Wikipedia search application using fetch, input elements, and dynamic DOM rendering.
8. Implement a countdown timer using scheduler functions (setTimeout/setInterval).
9. Create a class-based student management system using JS classes, constructors, and inheritance.
10. Build a Node.js script that uses core modules and a third-party npm package to process data.
11. DOM manipulation using querySelector and classList.
12. Form data storage and display dynamically.
13. JavaScript array operations using map(), filter(), reduce().
14. Async programming using Promises and async/await.
15. Build a REST API client using Node.js that fetches and displays data from a public API.

Text Books

1. Jon Duckett, JavaScript and jQuery: Interactive Front-End Web Development (Updated Edition), Wiley, 2023.
2. Marijn Haverbeke, Eloquent JavaScript, 4th Edition, No Starch Press, 2024.
3. Kyle Simpson, You Don't Know JS Yet: Get Started, 2nd Edition, O'Reilly, 2023.

Reference Books

1. Ethan Brown, Learning JavaScript, 3rd Edition, O'Reilly, 2023.
2. Addy Osmani, Learning JavaScript Design Patterns, 2nd Edition, O'Reilly, 2023.
3. Azat Mardan, Node.js in Action, 2nd Edition Updated, Manning, 2023.

In Semester Evaluation Scheme

1. FEP assessment must be based on the overall performance of the student with every assignment/practical graded from time to time.
2. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC207	Foundation of Data Structures Laboratory	-	4	-	-	2	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min Pass	Marks	Min Pass	Total (Marks)
UADPC207	Foundation of Data Structures Laboratory	25	40%	25	40%	50

FOUNDATION OF DATA STRUCTURES LABORATORY

Course Outcomes:

After the end of this course, students will be able to:

CO1 **Apply**[3] arithmetic operators, user input handling, and data type validation to perform basic calculations while following operator precedence rules.

CO2 **Implement**[3] loops, conditional statements, and user-defined functions to generate number patterns and compute mathematical series such as summations.

CO3 **Implement**[3] arrays, vectors, and queues to store and manage data, performing operations such as finding maximum, minimum, sum, average, and processing tasks in sequence.

CO4 **Analyze**[4] recursive functions to solve problems such as factorial and Fibonacci computation, and manipulate strings for reversing text, checking anagrams, and validating patterns.

CO5 **Evaluate**[5] sorting and searching algorithms to organize and retrieve data efficiently, and apply bit manipulation techniques to manage flags and settings in programs.

Suggested List of Experiments

1. Develop a program using C++ operators, type conversions, conditionals, loops, and functions to perform a sequence of basic computational tasks.
2. Build a menu-driven application using STL structures such as vectors, queues, and stacks to simulate simple task or ticket management operations.
3. Create a program that performs digit-based operations, finds divisors, checks for prime numbers, and computes the GCD and LCM of user inputs.
4. Design an array-based system to manage items like daily expenses or student scores, performing operations such as finding the maximum, minimum, sum, and average.
5. Write recursive programs to compute factorials, generate Fibonacci numbers, reverse arrays, and print sequences from 1 to N and N to 1.
6. Implement string-processing applications that reverse user messages, detect anagrams, and validate simple username or text patterns.
7. Apply sorting algorithms (Selection, Bubble, Insertion, Merge, and Quick Sort) to arrange real-world lists such as marks, product prices, or names.
8. Develop a searching module that demonstrates linear and binary search, along with lower and upper bound operations for structured datasets.
9. Create programs to convert numbers between decimal, binary, octal, and hexadecimal systems and perform bitwise operations including AND, OR, XOR, NOT, and shift operations.
10. Implement bit-manipulation utilities to check, set, clear, toggle, or count bits in integers, simulating real-world flag control or setting toggles.
11. Implement a program to remove duplicates from an array and move zeros to the end.
12. Array problem solving (second largest, consecutive ones)

13. Backtracking subsets/combinations.
14. Sorting algorithm comparison (performance).
15. Write a program to find lower bound and upper bound in a sorted array using binary search logic.

Text Books

1. Thomas H. Cormen et al., Introduction to Algorithms, 4th Edition, MIT Press, 2022.
2. Bjarne Stroustrup, Programming: Principles and Practice Using C++, 3rd Edition, Addison-Wesley, 2022.
3. Michael T. Goodrich, Data Structures and Algorithms in C++, 2nd Edition Updated, Wiley, 2023.

Reference Books

1. Aditya Bhargava, Grokking Algorithms, 2nd Edition, Manning, 2023.
2. Steven Skiena, The Algorithm Design Manual, 3rd Edition, Springer, 2022.
3. Scott Meyers, Effective Modern C++ (Updated Edition), O'Reilly, 2022.

In Semester Evaluation Scheme

1. FEP assessment must be based on the overall performance of the student with every assignment/practical graded from time to time.
2. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
USTBS206	Modern Physics	2	-	-	2	-	-

Course Code	Course name	Evaluation scheme (In semester)					Semester end exam (SEE)		Total (Marks)
		T1	T2	FET	Term work	Total	Marks	Min Pass	
USTBS206	Modern Physics	-	10	5	-	15	35	40%	50

MODERN PHYSICS

Course Description

This course introduces the basic concepts of quantum physics and how they are used to understand very small particles and modern technologies. It covers topics like wave-particle duality, Schrödinger's equation, and quantum tunnelling, along with an introduction to quantum computing concepts such as qubits, superposition, and entanglement. The course also explains semiconductor devices, optoelectronic components, and memory technologies used in computers. In addition, it explores electromagnetic waves, light interference, and their role in imaging systems, and finally discusses photonics, fiber optics, and sensors with applications in communication, IoT, and automation.

Course Objectives

1. To explain the fundamental concepts of quantum physics including wave-particle duality, Schrödinger's equation, and quantum tunnelling.
2. To build understanding of quantum computing concepts such as qubits, superposition, entanglement, and quantum circuits.
3. To evaluate semiconductor, optoelectronic, magnetic, and ferroelectric materials in computing technologies.
4. To analyze electromagnetic wave behavior, interference, and matrix-based modelling for imaging systems.
5. To apply principles of photonics, fiber optics, and sensor technologies in real-world applications.

Course Outcomes:

After the successful completion of the course, students will be able to:

CO1 **Explain**[1] fundamental quantum mechanical principles such as wave-particle duality, Schrödinger's equation, and quantum tunnelling.

CO2 **Apply**[3] quantum computing concepts including qubits, superposition, entanglement, and quantum circuits to solve basic computational problems.

CO3 **Analyze**[4] semiconductor, optoelectronic, magnetic, and ferroelectric materials and evaluate their role in modern computing technologies.

CO4 **Evaluate**[5] electromagnetic waves, interference phenomena, and matrix-based models for optical imaging and signal processing applications.

CO5 **Design**[6] solutions using photonics, fiber optics, and sensor technologies for communication, IoT, and automation systems.

Course Contents

Module	Unit	Description	Hours
1.0		**FUNDAMENTALS OF QUANTUM PHYSICS & QUANTUM COMPUTING**	
	1.1	Fundamentals of Quantum Physics: Planck's theory; Photoelectric effect; Wave-particle duality; Schrödinger's equation; Particle in 1D potential box; Quantum tunnelling (Transmission and Reflection coefficients).	6
	1.2	Introduction to Quantum Computing: Foundation of quantum computing; Qubits; Superposition; Quantum entanglement; Quantum gates; Quantum circuits.	
2.0		**SEMICONDUCTOR PHYSICS & OPTOELECTRONICS**	
	2.1	Semiconductor Physics: Energy bands in solids; Intrinsic and extrinsic semiconductors; PN junction diode; Transistors (qualitative); Applications in CPUs, GPUs, SSDs.	6
	2.2	Optoelectronics: Light-matter interaction; LEDs; Photodiodes; Laser diodes; Display technologies (LCD, LED, OLED); Applications in sensors and communication.	
3.0		**MAGNETISM & FERRO ELECTRIC MEMORY TECHNOLOGIES**	
	3.1	Magnetism: Classification of magnetic materials; Domain theory; Hysteresis; Soft and hard magnetic materials; Magnetic storage (HDDs, MRAM).	6
	3.2	Ferroelectricity: Dielectric materials; Ferroelectric capacitors and transistors; Memristors; Dielectric tunnelling; FeRAM and emerging memory technologies.	
4.0		**ELECTROMAGNETIC WAVES, INTERFERENCE & MATRIX REPRESENTATION**	
	4.1	Electromagnetic Waves: Light as EM wave; Superposition principle; Interference; Wave intensity; Digital sensors and image capture.	6
	4.2	Matrix Representation: Mathematical modelling of interference; Intensity computation using matrices; Image as matrix; Applications in imaging systems.	
5.0		**PHOTONICS, FIBER OPTICS & SENSORS**	
	5.1	Photonics & Fiber Optics: Total internal reflection; Optical fibers; Acceptance angle; Numerical aperture (qualitative); Signal losses (attenuation, dispersion); Applications.	6
	5.2	Sensors: Sensor characteristics (sensitivity, accuracy, precision); Mechanical, optical, and thermal sensors; Applications in robotics, IoT, and automation.	

Text Books

1. Neamen, D. A., Semiconductor Physics and Devices, **4th ed.**, McGraw-Hill, **2021**.
2. Kasap, S. O., Principles of Electronic Materials and Devices, **4th ed.**, McGraw-Hill, **2021**

Reference Books

1. Nielsen, M. A. & Chuang, I. L., Quantum Computation and Quantum Information, Cambridge University Press, **2020**
2. Saleh, B. E. A. & Teich, M. C., Fundamentals of Photonics, **3rd ed.**, Wiley, **2021**.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on the first two modules and T2 should be based on the next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.
3. Term work assessment shall be based on the overall performance of the student with every assignment graded from time to time.
4. The grades will be converted to marks as per the credit and grading system manual and should be added and averaged.