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 (R25)

Semester: III-IV

Programme Structure R25 for B.Tech. :Second Year Artificial Intelligence and Data Science

Semester III: Teaching Scheme

Course Code	Course Name	Teaching Scheme(Hrs/week)			Credits Assigned			Total Credits
		Th	Pr	Tu	Th	Pr	Tu	
UADPC301	Server Side Engineering	3	-	-	3	-	-	3
UADPC302	Design and Analysis of Algorithms	3	-	-	3	-	-	3
UADPC303	Probability and Statistics	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor I	2	-	-	2	-	-	2
UADAE301	Logical Reasoning	2	-	-	2	-	-	2
UXXOE01X	Open Elective I	1	2	-	-	2	-	2
UADPC304	Server Side Engineering Laboratory	-	2	-	-	1	-	1
UADPC305	Design and Analysis of Algorithms Laboratory	-	4	-	-	2	-	2
UADFP301	Field Project	-	4	-	-	2	-	2
UISCC0XX	Co-Curricular Courses I	-	4	-	-	2	-	2
	Total	**14**	**16**	**-**	**13**	**9**	**-**	**22**

Semester III: Evaluation Scheme

Course Code	Course Name	Theory Marks					Practical Marks		
		T1	T2	FET	Total	SEE	Term work / Tutorial	Practical Oral/POE	Total
UADPC301	Server Side Engineering	10	10	5	25	50	-	-	75
UADPC302	Design and Analysis of Algorithms	10	10	5	25	50	-	-	75
UADPC303	Probability and Statistics	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor I	-	10	5	15	35	-	-	50
UADAE301	Logical Reasoning	-	10	5	15	35	-	-	50
UXXOE01X	Open Elective I	-	-	-	-	-	50	-	50
UADPC304	Server Side Engineering Laboratory	-	-	-	-	-	25	-	25
UADPC305	Design and Analysis of Algorithms Laboratory	-	-	-	-	-	25	25	50
UADFP301	Field Project	-	-	-	-	-	25	25	50
UISCC0XX	Co-Curricular Courses I	-	-	-	-	-	50	-	50
								Total	**550**

Programme Structure R25 for B.Tech.: Second Year Artificial Intelligence and Data Science

Semester IV: Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			Total Credits
		Th	Pr	Tu	Th	Pr	Tu	
UADPC401	Operating Systems	2	-	-	2	-	-	2
UADPC402	Computer Networks	2	-	-	2	-	-	2
UADPC403	Advanced Algorithms & Optimization	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
UADAE401	Advanced Aptitude	3	-	-	3	-	-	3
UXXOE02X	Open Elective II	1	2	-	-	2	-	2
USTBS401	Linear Algebra and Calculus	3	-	-	3	-	-	3
UADPC404	Operating Systems Laboratory	-	2	-	-	1	-	1
UADPC405	Computer Networks Laboratory	-	2	-	-	1	-	1
UADPC406	Advanced Algorithms & Optimization Laboratory	-	2	-	-	1	-	1
UISCC0XX	Co-Curricular Courses II	-	4	-	-	2	-	2
	Total	**16**	**12**	**-**	**15**	**7**	**-**	**22**

Semester IV: Evaluation Scheme

Course Code	Course Name	Theory Marks					Practical Marks		
		T1	T2	FET	Total	SEE	Term work / Tutorial	Practical Oral/POE	Total
UADPC401	Operating Systems	-	10	5	15	35	-	-	50
UADPC402	Computer Networks	-	10	5	15	35	-	-	50
UADPC403	Advanced Algorithms & Optimization	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor II	-	10	5	15	35	-	-	50
UADAE401	Advanced Aptitude	10	10	5	25	50	-	-	75
UXXOE02X	Open Elective II	-	-	-	-	-	50	-	50
USTBS401	Linear Algebra and Calculus	10	10	5	25	50	-	-	75
UADPC404	Operating Systems Laboratory	-	-	-	-	-	25	-	25
UADPC405	Computer Networks Laboratory	-	-	-	-	-	25	-	25
UADPC406	Advanced Algorithms & Optimization Laboratory	-	-	-	-	-	25	-	25
UISCC0XX	Co-Curricular Courses II	-	-	-	-	-	50	-	50
								Total	**550**

Course Code	Course Name	Teaching Scheme(Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC301	Server Side Engineering	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	
UADPC301	Server Side Engineering	10	10	5	-	25	50	40%	75

SERVER SIDE ENGINEERING

Course Description

This course covers the fundamentals of server-side application development using Node.js and Express. It begins with setting up HTTP servers, handling requests, and managing errors, followed by database integration with SQLite and building RESTful APIs. Students will learn authentication using JWT tokens, middleware implementation, and security best practices. The course also covers file handling, streams, and MongoDB integration, equipping students to build real-world, scalable server-side applications.

Pre-requisites: Database Management Systems

Course Objectives

1. To introduce the fundamentals of server-side application development and backend architectures.
2. To develop skills in creating APIs, handling requests, and managing server-side logic.
3. To enable students to work with databases, authentication, and session management techniques.
4. To familiarize students with RESTful services, middleware, and backend deployment workflows.
5. To provide practical exposure in building scalable and secure server-side applications.

Course Outcomes:

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

CO1 **Identify**[1] the core concepts of Node.js, Express, HTTP server architecture, CommonJS/ES modules, and NPM package management.

CO2 **Describe**[2] the HTTP request-response cycle, routing mechanisms, error handling, and SQLite database integration in a server-side application.

CO3 **Implement**[3] RESTful APIs with request validation, JWT authentication, and middleware-based security controls.

CO4 **Analyze**[4] file handling, stream operations, and MongoDB integration to assess their role in building scalable backend solutions.

CO5 **Construct**[6] a complete data-driven backend application integrating authentication, file handling, and MongoDB with Node.js.

Course Contents

Module	Unit	Description	Hours
1.0		**NODE JS AND EXPRESS FUNDAMENTALS**	9
	1.1	Introduction to Backend Development, Node.js Basics, Express.js Introduction, HTTP Server Setup, Handling HTTP Requests	
	1.2	Development Environment Setup, NPM Package Management, Routing Basics, Request and Response Objects, Error Handling.	
2.0		**DATABASE INTEGRATION AND RESTFUL API DEVELOPMENT**	9
	2.1	SQLite Data Creation and Integration, Database Schema, CRUD Operations, Query Writing, REST Principles, RESTful API Development,	
	2.2	Query Handling, Request Validation, Response Formatting, Testing APIs, Debugging Techniques.	
3.0		**AUTHENTICATION AND MIDDLEWARE**	9
	3.1	JWT Token Authentication, Authentication Strategies, User Management, Middleware Functions and Implementation	
	3.2	Custom Middleware Development, Password Hashing, Security Best Practices, Error Logging	
4.0		**FILE HANDLING AND STREAMS**	9
	4.1	File Handling using the fs Module, File System Operations, Working with File Streams, Data Transformation Streams	
	4.2	Error Handling with Streams, Stream Piping, Buffer Management, Asynchronous File Operations	
5.0		**MONGODB AND NODE JS INTEGRATION**	9
	5.1	MongoDB Installation and Setup, MongoDB Atlas Integration, Native Driver vs Mongoose, Schema Design, Defining Models, Filtering and Sorting.	
	5.2	CRUD APIs with MongoDB, Role-Based Authorization, Testing MongoDB-Backed APIs, Debugging	

Text Books

1. Buna, S., Efficient Node.js: A Beyond-the-Basics Guide, 1st ed., 2025.
2. Rodrigues, D., Learn Express.js: Build Scalable Applications with Express.js, 1st ed., 2024.
3. Borucki, A., MongoDB 8.0 in Action: Building on the Atlas Data Platform, 3rd ed., Manning Publications, 2025.

Reference Books

1. Pasquali, S., Mastering Node.js: Advanced Techniques for Production-Ready Development, 1st ed., 2023.
2. Brown, E., Web Development with Node and Express: Leveraging the JavaScript Stack, 2nd ed., O'Reilly Media, 2022.
3. Lim, G., Beginning Node.js, Express and MongoDB Development, 1st ed., Apress, 2019.

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 (FET).
3. Term work assessment shall be based on the overall performance of the student with every tutorial graded from time to time.
4. The grades will be converted to marks as per the credit and grading system manual.

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 the 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
UADPC302	Design and Analysis of Algorithms	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	
UADPC302	Design and Analysis of Algorithms	10	10	5	-	25	50	40%	75

DESIGN AND ANALYSIS OF ALGORITHMS

Course Description

This course covers core data structures and algorithmic techniques used in software engineering and competitive programming. Students will learn hashing, array pattern techniques (prefix sum, sliding window, two-pointer), linked lists, stacks, queues, trees, binary search trees, and heaps. Emphasis is placed on problem-solving, implementation, and analysis of time and space complexity for real-world applications.

Pre- requisites : Foundation of Data Structures

Course Objectives

1. To introduce the fundamental concepts of algorithm design and computational complexity analysis.
2. To develop problem-solving skills using divide-and-conquer, greedy, and dynamic programming techniques.
3. To enable students to analyze algorithm efficiency using asymptotic notations and recurrence relations.
4. To familiarize students with graph, sorting, and searching algorithms for optimization problems.
5. To build competency in designing efficient algorithms for real-world computational challenges.

Course Outcomes:

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

CO1 **Define**[1] hash tables, hash functions, collision resolution strategies, and fundamental array pattern techniques including prefix sum and two-pointer approach.

CO2 **Describe**[2] the structure, operations, and time/space complexity of singly, doubly, and circular linked lists along with standard traversal and manipulation techniques.

CO3 **Implement**[3] stack and queue data structures using array and linked-list representations and apply monotonic structures and expression evaluation techniques to solve problems.

CO4 **Analyze**[4] binary trees and BSTs by applying traversal algorithms, computing structural properties (height, diameter, LCA), and evaluating BST operations for correctness.

CO5 **Evaluate**[5] heap-based data structures and priority queues by assessing their suitability for priority-based and frequency-based problems, justifying algorithmic choices.

Course Contents

Module	Unit	Description	Hours
1.0		**HASHING, SLIDING WINDOW & TWO-POINTER**	9
	1.1	Hash Tables, Hash Functions, Collision Handling, Collision Resolution: Separate Chaining, Open Addressing.	
	1.2	Prefix Sum, Sliding Window Technique, Two-Pointer Technique, Longest Subarray with Sum K, Maximum Subarray (Kadane's Algorithm), Two Sum.	
2.0		**LINKED LISTS**	9
	2.1	Singly Linked List: Traversal, Insertion, Deletion, Doubly Linked List: Traversal, Insertion, Deletion, Circular Linked List.	
	2.2	Reversals, Floyd's Cycle-Finding Algorithm, Cycle Detection and Length, Adding Two Numbers, Merge Two Sorted Lists, Intersection of Linked Lists, Flatten List.	
3.0		**STACKS & QUEUES**	9
	3.1	Stack Implementation (Array and Linked List), Expression Conversions and Evaluations (Infix, Prefix, Postfix), Monotonic Stack, Balanced Parenthesis, Min Stack, Next Greater/Smaller Element, Asteroid Collision, Histogram.	
	3.2	Queue Implementation (Array and Linked List), Stack Using Queue, Queue Using Stack.	
4.0		**TREES & BINARY SEARCH TREES**	9
	4.1	Binary Tree: Introduction, Traversals (Inorder, Preorder, Postorder, Level Order), Height, Diameter, Balanced Tree, Top View, Right View, Lowest Common Ancestor (LCA), Maximum Path Sum.	
	4.2	BST: Insertion, Deletion, Validation, Kth Smallest Element, Predecessor and Successor, Merge Two BSTs.	
5.0		**HEAPS**	9
	5.1	Min-Heap, Max-Heap, Heap Sort, Heap Conversions.	
	5.2	Priority Queue Applications, Kth Largest Element, Merge K Sorted Arrays, Top K Frequent Elements.	

Text Books

1. Narasimha Karumanchi, Data Structures and Algorithmic Thinking with Python, CareerMonk Publications, 2020.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.
3. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, 3rd Edition, 2017.

Reference Books

1. Adnan Aziz, Tsung-Hsien Lee and Amit Prakash, Elements of Programming Interviews, 2nd Edition, 2012.
2. Gayle Laakmann McDowell, Cracking the Coding Interview, 6th Edition, CareerCup, 2015.
3. Steven S. Skiena, The Algorithm Design Manual, 3rd Edition, Springer, 2020.

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 (FET).

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 the 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
UADPC303	Probability and Statistics	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	
UADPC303	Probability and Statistics	10	10	5	-	25	50	40%	75

PROBABILITY AND STATISTICS

Course Description

This course covers the fundamentals of descriptive statistics, probability theory, random variables, and statistical inference. Students explore data visualization techniques, probability rules, and behavior of random variables through different distribution functions. The course further covers sampling techniques, the central limit theorem, and hypothesis testing, preparing studen ts to handle data-driven decision-making and statistical analysis effectively.

Course Objectives

1. To introduce descriptive statistics and data visualization for summarizing and interpreting data.
2. To develop understanding of probability concepts including conditional probability and Bayes' theorem.
3. To enable students to work with discrete and continuous probability distributions for real-world applications.
4. To teach inferential statistics including sampling, hypothesis testing, and confidence intervals.
5. To apply correlation and regression techniques to analyze and model relationships between variables.

Course outcomes:

After successful completion of this course, students will able to

CO1 **Recall**[1] fundamental concepts of descriptive statistics, including measures of central tendency, dispersion, and data visualization techniques.

CO2 **Describe**[2] probability concepts, including conditional probability and Bayes' theorem, to model and interpret uncertain events.

CO3 **Apply**[3] discrete and continuous probability distributions, including Binomial, Poisson, Normal, and Exponential distributions, to solve real-world problems.

CO4 **Analyze**[4] relationships between variables using correlation, covariance, simple linear regression, and multiple linear regression techniques.

CO5 **Evaluate**[5] sampling methods, confidence intervals, and hypothesis testing techniques to draw meaningful conclusions and support data-driven decision-making.

Course Contents

Module	Unit	Description	Hours
1.0		**DESCRIPTIVE STATISTICS**	9
	1.1	Introduction to Statistics, Importance, Data Visualization Using tables and plots, Measures of Central Tendency (Mean, Median, Mode).	
	1.2	Measures of Dispersion (Range, IQR, Variance, Standard Deviation), Applications in Quality Control and Risk Assessment, Skewness, Kurtosis, Types of Skewness and Kurtosis.	
2.0		**FUNDAMENTALS OF PROBABILITY**	9
	2.1	Basics of Probability, Classical, Empirical, and Subjective Probability, Sample Space and Events, Mutually Exclusive, Exhaustive, and Independent Events.	
	2.2	Addition and Multiplication Rules, Conditional Probability, Law of Total Probability, Bayes' Theorem, Permutations and Combinations.	
3.0		**PROBABILITY DISTRIBUTIONS**	9
	3.1	Introduction to Random Variables, Discrete & Continuous Random Variables, Discrete Probability Distributions, Probability Mass Function (PMF) - Bernoulli, Binomial, Poisson distributions;	
	3.2	Continuous Probability Distributions - Probability Density Function (PDF), Cumulative Distribution Function (CDF), Normal Distribution & its types, Uniform, Beta, Gamma, & Exponential Distributions.	
4.0		**INFERENTIAL STATISTICS**	9
	4.1	Introduction to Sampling, Sampling Techniques, Sampling Distribution, Central Limit Theorem, T, Chi-Square, & F Distributions, Confidence intervals & levels, Level of significance.	
	4.2	Hypothesis Testing (Null and Alternative Hypotheses, Type I and Type II Errors, p-value, Significance Level), z-test, t-test (One-sample, Two-sample, Paired), Chi-square Tests (Independence, Goodness-of-fit), One-way ANOVA, Post-hoc Analysis.	
5.0		**CORRELATION AND REGRESSION**	9
	5.1	Correlation, Covariance, Types of Correlation, Coefficient of Correlation.	
	5.2	Simple Linear Regression, Multiple Linear Regression, Nonlinear Regression.	

Text Books

1. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 6th Edition, Academic Press, 2021.
2. Ani Adhikari and John DeNero, Computational and Inferential Thinking: The Foundations of Data Science, 2019.
3. Douglas C. Montgomery and George C. Runger, Applied Statistics and Probability for Engineers, 7th Edition, Wiley, 2018.

Reference Books

1. Roger D. Peng and Elizabeth Matsui, The Art of Data Science: A Guide for Anyone Who Works with Data, 2018.
2. Morris H. DeGroot and Mark J. Schervish, Probability and Statistics, 4th Edition, Pearson, 2012.
3. William Feller, An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition, Wiley, 2008.

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 (FET).
3. Term work assessment shall be based on the overall performance of the student.

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 the 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
UXXMM0XX	Multidisciplinary Minor I	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	
UXXMM0XX	Multidisciplinary Minor I	-	10	5	-	15	35	40%	50

MULTIDISCIPLINARY : MINOR I (AI APPLICATIONS IN BANKING AND FINANCE)

Course Description

This course introduces the application of Artificial Intelligence in financial services, covering key financial domains such as banking, insurance, and capital markets. It explores how Generative AI is transforming finance through applications like fraud detection, trading, and advisory systems. The course focuses on building AI-driven financial agents using techniques such as RAG and integrating real-world data sources. It also covers evaluation, monitoring, and fine-tuning of AI models for financial use cases. Additionally, the course emphasizes ethical considerations, data privacy, and responsible AI practices in financial systems.

Pre-requisites: Building LLM Applications

Course Objectives

1. To introduce fundamental concepts of finance and their integration with AI technologies in financial services.
2. To develop the ability to build AI-driven financial applications using modern techniques such as RAG and agent-based systems.
3. To explore evaluation, observability, and performance measurement techniques for financial AI systems.
4. To provide knowledge of advanced model optimization methods including fine-tuning and deployment strategies.
5. To emphasize ethical considerations, governance, and responsible AI practices in financial applications.

Course outcomes:

After successful completion of this course, students will able to

CO1 **Define**[1] finance fundamentals, key financial verticals, and AI-driven applications in financial services.

CO2 **Apply**[3] RAG-based techniques and tools to design and develop AI-powered financial agents for real-world workflows.

CO3 **Analyze**[4] evaluation metrics and implement observability techniques to assess the performance of financial AI systems.

CO4 **Evaluate**[5] fine-tuning strategies and deployment techniques for optimizing scalable financial AI models.

CO5 **Design**[6] responsible and ethical AI solutions by addressing bias, governance, and risk in financial systems.

Course Contents

Module	Unit	Description	Hours
1.0		**FINANCE FUNDAMENTALS AND AI FOR FINANCE**	6
	1.1	Finance fundamentals - markets, instruments, institutions, Key verticals: banking, insurance, capital markets, wealth management	
	1.2	Why Gen AI is transforming financial services, Real-world examples.Applications: document intelligence, fraud detection, trading, advisory, compliance,	
2.0		**APPLICATIONS OF AI IN FINANCE**	6
	2.1	Building personal finance advisor agent using RAG, Tool integration with financial APIs, web search (market data, filings, news)	
	2.2	Designing simple financial agents for real-world workflows, Tools for enhancing productivity with AI in financial use cases.	
3.0		**EVALUATION AND OBSERVABILITY FOR FINANCIAL AI SYSTEMS**	6
	3.1	Implementing RAG evaluations for Financial AI AgentsRAGAS metrics: Faithfulness, Context Precision, Answer Relevancy, Context Recall for financial use cases (trading signals, fraud detection)	
	3.2	LLM-as-judge evaluation for financial decision-making quality,Detecting and reducing hallucinations in financial AI systems,Building automated evaluation pipelines and regression test suites for financial workflows Task success rate evaluation for financial processes (e.g., trading, fraud detection)	
4.0		**ADVANCED FINE-TUNING AND DEPLOYING AI MODELS**	6
	4.1	Production monitoring of financial LLM systems, LangSmith tracing and evaluation workflows for finance agents, Evaluation frameworks for RAG-based financial document systems, RAGAS evaluation framework, Core RAG metrics, Prompting vs RAG vs fine-tuning decision strategies for financial use cases	
	4.2	Fundamentals of LoRA and adapter-based fine-tuning on financial datasets, QLoRA and 4-bit quantization techniques for cost-efficient financial AI systems, Preparing domain-specific financial training datasets (transactions, contracts, reports), Instruction-tuning dataset creation for financial tasks, Training pipelines using HuggingFace PEFT libraries, Evaluating fine-tuned models against base models for fraud detection and risk scoring, Deployment of financial AI models with APIs, monitoring, and versioning.	
5.0		**AI ETHICS AND RESPONSIBLE AI IN FINANCIAL SYSTEMS**	6
	5.1	Ethical risks in financial AI systems (bias in credit scoring, algorithmic trading risks), Bias in financial training data and machine learning models, Data privacy and security considerations for sensitive financial data	
	5.2	Responsible AI frameworks in financial services, Fairness and transparency in AI-driven financial decision-making, AI governance frameworks for financial institutions, Risk management strategies for AI-powered financial products, Human-in-the-loop AI systems for fraud review and compliance, Building trustworthy, explainable, and accountable AI systems in finance.	

Text Books

1. AI for Finance – Edward P. K. Tsang, 2023
2. AI for Finance Professionals – Prabash Galagedara, 2024
3. Artificial Intelligence (AI) and Finance – Bahaaeddin A. M. Alareeni, 2023

Reference Books

1. Supremacy: AI, ChatGPT and the Race That Will Change the World – Parmy Olson, 2024
2. Resisting AI: An Anti-fascist Approach to Artificial Intelligence – Dan McQuillan, 2022
3. Finance & Development: Artificial Intelligence – What AI Means for Economics – International Monetary Fund, 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. FET shall 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 the 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
UADAE301	Logical Reasoning	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	
UADAE301	Logical Reasoning	-	10	5	-	15	35	40%	50

LOGICAL REASONING

Course Description

This course is designed to strengthen logical reasoning and data arrangement skills essential for aptitude assessments. Learners will master techniques for solving seating, sequencing, and blood relation problems along with data sufficiency and deduction - based reasoning. The course also enhances visual reasoning through pattern recognition and geometric analysis.

Pre-requisites: Numerical Ability and Quantitative Aptitude

Course Objectives

1. To introduce the fundamental concepts of logical reasoning and analytical thinking.
2. To develop skills in solving seating arrangements, grouping, and sequencing problems.
3. To enable students to analyze data sufficiency and blood relation problems using logical deduction.
4. To familiarize students with syllogisms, Venn diagrams, and reasoning puzzle techniques.
5. To build confidence in solving visual reasoning and pattern deduction problems for competitive examinations.

Course outcomes:

After successful completion of this course, students will able to

CO1 **Identify[1]** seating arrangement patterns and rules in linear, circular, and conditional arrangements.

CO2 **Identify[1]** seating arrangement patterns and rules in linear, circular, and conditional arrangements.

CO3 **Apply[3]** Venn diagrams and logical deduction techniques to solve set theory and syllogism problem

CO4 **Analyze[4]** syllogism conclusions and logical reasoning puzzles using structured reasoning frameworks.

CO5 **Evaluate[5]** visual reasoning problems involving mirror images, water images, paper folding and figure analogies by assessing spatial relationships and solution accuracy

Course Contents

Module	Unit	Description	Hours
1.0		**DATA ARRANGEMENTS**	6
	1.1	Linear and Circular Arrangements, Row and Column-Based Seating Arrangements.	
	1.2	Grouping and Sequencing, Conditional Arrangements.	
2.0		**DATA SUFFICIENCY AND BLOOD RELATIONS**	6
	2.1	Data Sufficiency: Assessing Sufficiency of Statements in Logical Reasoning involving Numbers, Arrangements, Ages.	
	2.2	Blood Relations: Family Tree-Based Problems, Coded Blood Relations, Generation-Based Questions.	
3.0		**SET THEORY AND BASIC SYLLOGISMS**	6
	3.1	Venn Diagrams: Set Representation, Union, Intersection, and Complements of Sets.	
	3.2	Syllogisms: Statements and Conclusions, Logical Deduction Using Venn Diagrams.	
4.0		**VALIDITY AND LOGICAL REASONING PUZZLES**	6
	4.1	Evaluating the Validity of Syllogism Conclusions and Categorical Syllogism Problems.	
	4.2	Logical Reasoning Puzzles: Floor-Based, Box Arrangement, Scheduling, Classification, Grid-Based Puzzles.	
5.0		**SHAPES AND PATTERN DEDUCTION**	6
	5.1	Counting Figures: Identifying Geometric Shapes such as Triangles, Squares, Rectangles within Complex Figures.	
	5.2	Visual Reasoning: Mirror and Water Images, Paper Folding and Cutting, Pattern Completion, Figure Analogy, Odd-One-Out Identification.	

Text Books

1. R. S. Aggarwal, A Modern Approach to Logical Reasoning, Revised Edition, S. Chand Publishing, 2023.
2. Nishit K. Sinha, Logical Reasoning and Data Interpretation for the CAT, Latest Edition, Pearson Education.
3. R. S. Aggarwal, A Modern Approach to Non-Verbal Reasoning, Revised Edition, S. Chand Publishing, 2022.

Reference Books

1. Arun Sharma, How to Prepare for Logical Reasoning for CAT, Latest Edition, McGraw-Hill Education, 2023.
2. Edgar Thorpe, The Complete Guide to Verbal Ability and Logical Reasoning, Latest Edition, Pearson Education, 2021.
3. P. Khatri, Logical and Analytical Reasoning, Latest Edition, Upkar Prakashan, 2022.

Internal Assessment (T1, T2 and FET)

3. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
4. FET shall be assessed for 5 marks separately.

Semester End Examination

6. Question paper will comprise of 5 questions, each carrying 7 marks.
7. The duration of semester end examination shall be 2 hours.
8. The students need to solve all 5 questions.
9. Question No.1 will be compulsory and based on the entire syllabus.
10. 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
UXXOE01X	Open Elective I	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	
UXXOE01X	Open Elective I	-	-	-	50	50	-	40%	50

OPEN ELECTIVE I: APPLIED COMMUNICATIVE ENGLISH

Course Description

This course focuses on strengthening communicative competence through the practical application of English in academic, social, and professional contexts. It integrates grammatical accuracy with interactive speaking, writing, and presentation skills, emphasizing clarity, fluency, and confidence. Through structured modules on discussion, questioning, persuasion, and public speaking, learners develop the ability to express ideas logically, engage collaboratively, and adapt language effectively across varied communicative settings. It also provides targeted skill practice modules aligned with the B2 Cambridge English qualification, helping students prepare effectively for international certification while strengthening their academic and workplace communication abilities.

Pre-requisites: Communicative English Advanced

Course Objectives

1. To develop grammatical accuracy and fluency for effective participation in group discussions.
2. To enhance the ability to organize and present ideas logically using appropriate linguistic structures.
3. To build skills in questioning, responding, and articulating ideas clearly in academic contexts.
4. To strengthen persuasive communication and argumentation techniques.
5. To develop confidence and competence in public speaking and presentation delivery.

Course outcomes:

After successful completion of this course, students will able to

CO1 **Apply**[3] grammatical accuracy and construct coherent sentences to enhance fluency in group discussions.

CO2 **Apply**[3] appropriate linguistic structures and linking devices to organize ideas effectively in communication.

CO3 **Analyze**[4] questioning techniques and formulate precise responses using appropriate language structures.

CO4 **Evaluate**[5] and construct persuasive arguments using appropriate language, style, and discourse techniques.

CO5 **Design**[6] and deliver structured presentations with effective verbal and non-verbal communication skills.

Course Contents

Module	Unit	Description	Hours
1.0		**COLLABORATIVE COMMUNICATION FOR GROUP DISCUSSIONS (PART - 1)**	9
	1.1	Grammatical Accuracy and Fluency in Group Interaction	
	1.2	Structuring Sentences for Coherence and Variety.	
2.0		**COLLABORATIVE COMMUNICATION FOR GROUP DISCUSSIONS (PART - 2)**	9
	2.1	Organizing Ideas through Clauses, Phrases, and Transitions	
	2.2	Using Linking Devices for Logical Flow, Maintaining Clarity, Tone, and Appropriateness in Discussions.	
3.0		**STRATEGIC QUESTIONING AND RESPONSE TECHNIQUES**	9
	3.1	Purpose and Types of Questions in Academic Settings, Framing Clear and Context-Based Questions	
	3.2	Using Modals for Permission, Obligation, and Possibility, Responding with Precision and Confidence, Paraphrasing and Reporting Responses Accurately.	
4.0		**PERSUASION AND INFLUENCING OTHERS**	9
	4.1	Structuring Persuasive Discourse Effectively, Using Idioms, Collocations, and Fixed Phrases	
	4.2	Ensuring Fluency and Contextual Suitability, Building Credibility and Stylistic Coherence, Crafting Arguments and Counterarguments.	
5.0		**PUBLIC SPEAKING AND PRESENTATION SKILLS**	9
	5.1	Stress, Rhythm, and Intonation for Effective Delivery, Organizing and Structuring Presentations	
	5.2	Ensuring Tense and Grammatical Consistency, Engaging Audiences through Verbal and Non-Verbal Cues, Developing Confidence and Stage Presence.	

Text Books

1. Lucas, S. E., The Art of Public Speaking, 13th Edition, McGraw-Hill Education, 2019.
2. Raman, M. and Sharma, S., Technical Communication: Principles and Practice, 3rd Edition, Oxford University Press, 2022.
3. Bovee, C. L. and Thill, J. V., Business Communication Today, 15th Edition, Pearson Education, 2021.

Reference Books

1. Cialdini, R. B., Influence: The Psychology of Persuasion, New and Expanded Edition, Harper Business, 2021.
2. Carnegie, D., How to Win Friends and Influence People, Revised Edition, Simon and Schuster, 2022.
3. Leary, M. R., Introduction to Behavioral Research Methods, 7th Edition, Pearson Education, 2021.

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. FET shall be assessed for 5 marks separately.
3. Term work assessment shall be based on the overall performance of the student.

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 the 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
UADPC304	Server Side Engineering Laboratory	-	2	-	-	1	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam(OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
UADPC304	Server Side Engineering Laboratory	25	40%	-	40%	25

SERVER SIDE ENGINEERING LABORATORY

Course outcomes:

After successful completion of this course, students will able to

CO1 **Identify**[1] the structure of a Node.js project, Express server setup, and the flow of HTTP request-response cycles.

CO2 **Implement**[3] RESTful API development with SQLite database integration to perform CRUD data management operations.

CO3 **Apply**[3] JWT-based authentication, authorization middleware, and security best practices in server-side applications.

CO4 **Analyze**[4] file handling and stream operations by implementing upload, download, and data transformation scenarios.

CO5 **Construct**[6] complete data-driven backend APIs integrating MongoDB with Node.js and validate functionality using appropriate testing tools.

List of Experiments

1. Set up a Node.js project, initialize NPM, and create an Express server to handle HTTP GET and POST requests.
2. Implement modular routing with Express.js to create route handlers for multiple API endpoints with request and response handling.
3. Configure middleware for request logging and error handling; test routes using Postman or Thunder Client.
4. Integrate SQLite with a Node.js application and perform CRUD operations using parameterized SQL queries.
5. Build a complete RESTful API with request validation, structured JSON response formatting, and error handling.
6. Implement query handling, filtering, and debugging techniques for a SQLite-backed REST API.
7. Implement JWT-based user authentication including registration, login, and token verification in an Express application.
8. Create custom middleware functions for authorization controls, error logging, and security enforcement.
9. Implement role-based access control using JWT claims and test protected routes with different user roles.
10. Implement file upload and download functionality using the fs module and multipart form handling.
11. Use Node.js Readable and Writable streams to handle large file operations and data transformation.
12. Implement stream piping and asynchronous file operations with error handling and buffer management.
13. Connect a Node.js application to MongoDB Atlas and perform CRUD operations using Mongoose models.
14. Implement password hashing, JWT authentication, and role-based authorization for a MongoDB-backed API.
15. Build and test a complete backend application combining authentication, MongoDB, and file operations.

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 average

Course Code	Course Name	Teaching Scheme(Hrs/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC305	Design and Analysis of Algorithms Laboratory	-	4	-	-	2	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam(OE/POE)		
		Term Work	Min Pass	Marks	Min pass	Total (Marks)
UADPC305	Design and Analysis of Algorithms Laboratory	25	40%	25	40%	50

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

Course outcomes:

After successful completion of this course, students will able to

CO1 **Identify**[1] the appropriate search, sorting, and traversal algorithms for a given problem and recall their time and space complexity.

CO2 **Implement**[3] dynamic programming and greedy algorithms to solve classic optimization problems and verify their correctness.

CO3 **Analyze**[4] graph algorithms for shortest path and traversal problems, comparing their efficiency and applicability to real-world scenarios..

List of Experiments

1. Implement a hash table with separate chaining for collision resolution and demonstrate insert, search, and delete operations.
2. Implement open addressing (linear probing) for collision resolution and compare with separate chaining.
3. Solve the 'Two Sum' problem using hashing and compare time complexity with brute force.
4. Find the longest subarray with a given sum K using the prefix sum and sliding window technique.
5. Apply the two-pointer technique to solve problems: pair with given sum in a sorted array and container with most water.
6. Implement singly linked list with traversal, insertion (head, tail, position), and deletion operations.
7. Implement doubly linked list with forward and backward traversal, insertion, and deletion.
8. Detect cycle in a linked list using Floyd's Cycle-Finding Algorithm and find cycle length.
9. Merge two sorted linked lists and find the intersection point of two linked lists.
10. Implement stack using array and linked list; evaluate postfix expressions.
11. Solve Next Greater Element and Min Stack problems using monotonic stack.
12. Implement queue using array and linked list; implement queue using two stacks.
13. Implement Binary Tree with all traversals: inorder, preorder, postorder, and level order (BFS).
14. Find height, diameter, and check if a binary tree is balanced; print top view and right view.
15. Implement BST with insertion, deletion, and validation; find Kth smallest element.
16. Implement Min-Heap and Max-Heap with insert, delete, and heap sort operations.

17. Solve Kth Largest Element in an array and Top K Frequent Elements using a priority queue.
18. Merge K sorted arrays using a min-heap and analyze time complexity.
19. Implement LCA (Lowest Common Ancestor) in a Binary Tree and find Maximum Path Sum.
20. End-to-End: Solve a combined problem using hashing, linked list, stack, tree, and heap concepts.

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
UADFP301	Field Project	-	4	-	-	2	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam(OE/POE)		
		Term Work	Min pass	Marks	Min Pass	Total (Marks)
UADFP301	Field Project	25	40%	25	40%	50

FIELD PROJECT

Course Description

This course provides students with an opportunity to apply the knowledge and skills acquired throughout their program by designing, developing, and deploying a real-world software application. Students will select a project from a set of application domains such as entertainment, healthcare, e-commerce, social networking, productivity, education, travel, and analytics platforms. The course emphasizes problem-solving, software design, full-stack development, testing, deployment, and project documentation while following industry-standard development practices.

Prerequisite

Students should have prior knowledge of:

1. Programming fundamentals and object-oriented programming concepts.
2. Web development technologies including HTML, CSS, JavaScript, and frontend frameworks.
3. Database management concepts and SQL.
4. API integration, version control, and software development lifecycle fundamentals.

Course Objectives

- To provide students with hands-on experience in developing industry-relevant software applications.
- To enable students to apply software engineering principles in designing scalable and maintainable solutions.
- To strengthen problem-solving, analytical thinking, and decision-making abilities through project implementation.
- To develop skills in frontend development, backend development, database integration, testing, and deployment.
- To foster teamwork, communication, project management, and professional documentation skills.
- To prepare students for real-world software development environments and industry expectations.

Course outcomes:

After successful completion of this course, students will able to

CO1 **Apply**[3] software development concepts, tools, and methodologies to address the requirements of the selected application.

CO2 **Analyze**[4] user requirements, system constraints, and functional specifications to identify appropriate design and implementation strategies.

CO3 **Design**[5] scalable system architectures, user interfaces, and database models for real-world applications.

CO4 **Evaluate**[5] alternative technologies, software designs, and application performance using appropriate testing and quality assurance techniques

CO5 **Create**[6] scalable and user-centric software applications by integrating frontend, backend, databases, and third-party services.

List of Projects

1. **Movies App (Netflix/Amazon Prime Clone)** – A platform to browse, search, and watch movie details similar to streaming services.
2. **Covid Dashboard** – A dashboard to display real-time COVID-19 statistics and trends.
3. **Tasty Kitchens (Swiggy/Zomato Clone)** – A food ordering app to explore restaurants and place orders.
4. **Spotify Remix (Spotify Clone)** – A music streaming app to play songs and manage playlists.
5. **Mini Games (Collection of Memory Games**) – A set of simple interactive games to test memory and logic skills.
6. **Task Manager (Trello Clone)** – A productivity tool to create, organize, and track tasks using boards and cards.
7. **Github Profile Visualizer** – An app to fetch and display GitHub user profile details and repositories.
8. **Book Hub** – A platform to browse, search, and manage a collection of books.
9. **Insta Share (Instagram Clone)** – A social media app to share posts, like, and view feeds.
10. **Nxt Assess App** – An application to take and manage online assessments or quizzes.
11. **Quiz Game** – An interactive quiz application with questions, answers, and scoring.
12. **Nxt Mart** – An e-commerce app to browse products and simulate online shopping.
13. **Travel Trip** – A travel planning app to explore destinations and manage trips.
14. **Daily Mood Tracker** – An app to track and visualize daily moods over time.

Note: Students may choose any one of the above-mentioned projects based on their interest and will be required to work on it as part of their project implementation and learning process.

Programme Structure R25 for B.Tech.: Second Year Artificial Intelligence and Data Science

Semester IV: Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			Total Credits
		Th	Pr	Tu	Th	Pr	Tu	
UADPC401	Operating Systems	2	-	-	2	-	-	2
UADPC402	Computer Networks	2	-	-	2	-	-	2
UADPC403	Advanced Algorithms & Optimization	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
UADAE401	Advanced Aptitude	3	-	-	3	-	-	3
UXXOE02X	Open Elective II	1	2	-	-	2	-	2
USTBS401	Linear Algebra and Calculus	3	-	-	3	-	-	3
UADPC404	Operating Systems Laboratory	-	2	-	-	1	-	1
UADPC405	Computer Networks Laboratory	-	2	-	-	1	-	1
UADPC406	Advanced Algorithms & Optimization Laboratory	-	2	-	-	1	-	1
UISCC0XX	Co-Curricular Courses II	-	4	-	-	2	-	2
	Total	**16**	**12**	**-**	**15**	**7**	**-**	**22**

Semester IV: Evaluation Scheme

Course Code	Course Name	Theory Marks					Practical Marks		
		T1	T2	FET	Total	SEE	Term work / Tutorial	Practical Oral/POE	Total
UADPC401	Operating Systems	-	10	5	15	35	-	-	50
UADPC402	Computer Networks	-	10	5	15	35	-	-	50
UADPC403	Advanced Algorithms & Optimization	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor II	-	10	5	15	35	-	-	50
UADAE401	Advanced Aptitude	10	10	5	25	50	-	-	75
UXXOE02X	Open Elective II	-	-	-	-	-	50	-	50
USTBS401	Linear Algebra and Calculus	10	10	5	25	50	-	-	75
UADPC404	Operating Systems Laboratory	-	-	-	-	-	25	-	25
UADPC405	Computer Networks Laboratory	-	-	-	-	-	25	-	25
UADPC406	Advanced Algorithms & Optimization Laboratory	-	-	-	-	-	25	-	25
UISCC0XX	Co-Curricular Courses II	-	-	-	-	-	50	-	50
								Total	**550**

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC401	Operating Systems	2	-	-	2	-	-

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)	Min. Passing%	Total (Marks)
		T1	T2	FET	Term Work	Total			
UADPC401	Operating Systems	-	10	5	-	15	35	40%	50

OPERATING SYSTEMS

Course Description

This course provides a comprehensive understanding of operating system principles and internals. Students learn core OS concepts including process management, CPU scheduling, inter-process communication, memory management, and file systems. Practical exposure to Linux environments enables students to work with the command line, manage processes and files, and apply system programming concepts. The course also covers deadlock handling, virtual memory, disk scheduling, and modern OS architectures.

Course Objectives

1. To introduce the fundamental concepts of operating systems including process, memory, and file management.
2. To develop skills in working with Linux environments and command-line operations.
3. To enable students to apply process scheduling, synchronization, and deadlock handling techniques.
4. To familiarize students with memory management, virtual memory, and disk scheduling algorithms.
5. To provide practical understanding of file systems and operating system architectures.

Course Outcomes:

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

CO1 **Recall**[1] core operating system concepts, including processes, memory management, file systems, and system architecture.

CO2 **Describe**[2] Linux commands, file system navigation, and process monitoring in a Linux environment.

CO3 **Apply**[3] CPU scheduling and process management techniques to optimize system performance.

CO4 **Analyze**[4] concurrency issues, synchronization mechanisms, and deadlock-handling strategies in operating systems.

CO5 **Evaluate**[5] memory management techniques, file system designs, and disk scheduling algorithms for system efficiency.

Course Contents

Module	Unit	Description	Hours
1.0		**OPERATING SYSTEM BASICS**	9
	1.1	Introduction to OS and Internet, Overview of Computer Hardware, Operating Systems Objectives and Functions, Evolution of Operating Systems, Operating System Services.	
	1.2	System Calls, Operating System Structures, Types of Operating Systems, Case Studies: Windows Operating Systems, Linux Operating Systems, Operating Systems for Handheld Devices.	
2.0		**LINUX BASICS AND COMMAND LINE INTERFACE**	9
	2.1	Introduction to Linux Shell, Day-to-Day Use of Commands, Google Cloud Shell, Getting Started with Command Line, Working with File System, Working with Files.	
	2.2	Logging into a Linux System, Using Manual Help Pages, Linux Basic Commands, Miscellaneous Commands.	
3.0		**PROCESS MANAGEMENT AND CPU SCHEDULING**	9
	3.1	Overview of Process Management, Process Life Cycle and States, Process Control Block (PCB), Context Switching, Threads and Multithreading Models.	
	3.2	Scheduling Criteria, Types of Scheduling (Non-preemptive and Preemptive), CPU Scheduling Algorithms	
4.0		**PROCESS SYNCHRONIZATION AND DEADLOCK HANDLING**	9
	4.1	Concurrent Processes, Principles of Concurrency, Inter-Process Communication (Shared Memory, Message Passing), The Critical-Section Problem, Synchronization Hardware, Semaphores.	
	4.2	Classic Problems of Synchronization (Producer-Consumer, Dining Philosophers, Readers-Writers), Critical Regions, Monitors, Deadlock: Characterization, Prevention, Avoidance (Banker's Algorithm), Detection, and Recovery.	
5.0		**MEMORY MANAGEMENT, FILE SYSTEMS AND DISK SCHEDULING**	9
	5.1	Introduction to Memory Management, Memory Partitioning, Paging, Segmentation, Combined Paging and Segmentation, Virtual Memory Concepts, Demand Paging, Page Faults, Page Replacement Policies, File and Directory Navigation, Create and Delete Files and Directories, Copy and Move operations.	
	5.2	Linux File System Navigation, File and Directory File system and Implementation I/O Subsystems,, Disk I/O, Disk Scheduling Algorithms, File Management Principles, File Directories, File System Architecture, File Allocation, Inode Structure, Network Basics	

Text Books

1. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, Operating System Concepts, 10th Edition, Wiley Publications, 2021.
2. Andrew S. Tanenbaum and Herbert Bos, Modern Operating Systems, 4th Edition, Pearson Education, 2015.
3. Ramez Elmasri and A. Gil Carrick, Operating Systems: A Spiral Approach, 1st Edition, McGraw-Hill, 2009.

Reference Books

1. William Stallings, Operating Systems: Internals and Design Principles, 9th Edition, Pearson Education, 2018.
2. Gary J. Nutt, Operating Systems, 3rd Edition, Addison-Wesley, 2004.
3. Maurice J. Bach, The Design of the UNIX Operating System, Prentice Hall, 1986.

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. The fifth module will be assessed for 5 marks separately (FET).
3. Term work assessment shall be based on the overall performance of the student with every tutorial 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 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 the entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC402	Computer Networks	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	
UADPC402	Computer Networks	-	10	5	-	15	35	40%	50

COMPUTER NETWORKS

Course Description

This course introduces core concepts of computer networks across five modules. Students study networking fundamentals, physical and data link layer mechanisms, network layer routing and addressing, transport layer protocols for reliable communication, a nd application layer services. The course covers the OSI and TCP/IP models, transmission methods, switching, error handling, routing protocols, congestion control, QoS, DNS, HTTP, SMTP, and multimedia streaming, with practical exposure to protocol operations across wired and wireless networks.

Course Objectives

1. To introduce the structure and functions of computer networks and layered communication models.
2. To develop understanding of transmission techniques, switching methods, and networking media.
3. To enable students to apply routing, addressing, and forwarding mechanisms in network communication.
4. To familiarize students with transport layer and application layer protocols for reliable communication.
5. To comprehend modern networking concepts including QoS, multimedia communication, and wireless networks.

Course Outcomes:

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

CO1 **Recall**[1] key elements of computer networks, OSI and TCP/IP models, and addressing schemes.

CO2 **Describe**[2] the physical and data link layers, including transmission media, switching techniques, error control, and LAN technologies.

CO3 **Implement**[3] network layer functionalities, including routing, forwarding, and IP protocol mechanisms.

CO4 **Analyze**[4] transport layer protocols and compare their suitability for reliable communication.

CO5 **Evaluate**[5] application layer protocols for web, email, DNS, and file transfer services.

Course Contents

Module	Unit	Description	Hours
1.0		**NETWORKING FUNDAMENTALS**	6
	1.1	Introduction to Data Communication, Networks, Internet, Protocols and Standards, Network Topologies.	
	1.2	Network Models, Layered Tasks, Layers in OSI Model and their Functions, TCP/IP Protocol Suite, IPv4 and IPv6 Addressing.	
2.0		**PHYSICAL LAYER AND DATA LINK LAYER**	6
	2.1	Physical Layer, Digital and Analog Transmission, Bandwidth Utilization, Multiplexing and Spreading Techniques.	
	2.2	Switching, Transmission Media, Data Link Layer, Error Detection and Correction Techniques, Data Link Control, Multiple Access Protocols, Wired LANs (Ethernet), Wireless LANs (Wi-Fi), Virtual Circuit Networks.	
3.0		**NETWORK LAYER**	6
	3.1	Network Layer Functions, Logical Addressing, Internet Protocol (IPv4 and IPv6), Address Mapping (ARP/RARP), Error Reporting (ICMP).	
	3.2	Multicasting, Delivery and Forwarding, Routing Algorithms, Shortest Path Routing, Link State Routing, Hierarchical Routing, Distance Vector Routing, Multicast and Broadcast Routing.	
4.0		**TRANSPORT LAYER**	6
	4.1	Transport Layer Services, Process-to-Process Delivery, Client/Server Paradigm, Multiplexing and Demultiplexing, UDP Protocol, TCP Protocol and Connection Management.	
	4.2	Flow Control, Error Control, Congestion Control, SCTP, Quality of Service (QoS), Techniques to Improve QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.	
5.0		**APPLICATION LAYER**	6
	5.1	Application Layer Overview, Domain Name System (DNS): Name Space, DNS in Internet, Resolution, DNS Message Formats.	
	5.2	Remote Logging, E-Mail (SMTP, POP, IMAP), File Transfer (FTP), World Wide Web, HTTP, Network Management (SNMP), Multimedia: Audio and Video Digitizing, Streaming Stored Audio and Video, RTCP, Voice Over IP.	

Text Books

1. Andrew S. Tanenbaum, Nick Feamster, and David J. Wetherall, Computer Networks, 6th Edition, Pearson Education, 2022.
2. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, 8th Edition, Pearson Education, 2022.

Reference Books

1. Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach, 6th Edition, Morgan Kaufmann, 2022.
2. William Stallings, Data and Computer Communications, 10th Edition, Pearson Education, 2014.

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. The fifth module will be assessed for 5 marks separately (FET).

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 the entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC403	Advanced Algorithms & Optimization	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	
UADPC403	Advanced Algorithms & Optimization	10	10	5	-	25	50	40%	75

ADVANCED ALGORITHMS & OPTIMIZATION

Course Description

This course builds on foundational data structures to develop strong problem-solving skills using greedy algorithms, dynamic programming, and graph algorithms. Students will learn to formulate optimal solutions for classical and applied problems, understand trade-offs between approaches, and apply advanced graph techniques such as shortest paths, minimum spanning trees, and strongly connected components to real-world scenarios.

Pre-requisites: Design Analysis and Algorithm

Course Objectives

1. To develop problem-solving skills using greedy algorithms and optimization strategies.
2. To enable students to design dynamic programming solutions for computational problems.
3. To train the students in applying graph traversal and connectivity algorithms.
4. To enhance understanding of shortest path and minimum spanning tree algorithms for optimization.
5. To develop competency in solving advanced graph problems using SCC and critical connection algorithms.

Course Outcomes:

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

CO1 **Identify**[1] greedy strategy principles and classical optimization problems such as Knapsack and Activity Selection.

CO2 **Describe**[2] dynamic programming principles and their application to one-dimensional, two-dimensional, and subsequence-based problems.

CO3 **Apply**[3] BFS, DFS, cycle detection, and topological sorting techniques to solve graph connectivity problems.

CO4 **Analyze**[4] shortest-path algorithms (Dijkstra's and Bellman-Ford) and minimum spanning tree algorithms (Prim's and Kruskal's) for network optimization.

CO5 **Evaluate**[5] advanced graph algorithms, including strongly connected components, Kosaraju's algorithm, Tarjan's algorithm, and critical connections.

Course Contents

Module	Unit	Description	Hours
1.0		**GREEDY ALGORITHMS**	9
	1.1	Principles of Greedy Strategy, Fractional Knapsack, Activity Selection, Minimum Coins.	
	1.2	Job Scheduling, Platform Scheduling, Badge Distribution.	
2.0		**DYNAMIC PROGRAMMING – PART I**	9
	2.1	DP Principles, Memoization vs Tabulation, 1-D DP: Climbing Stairs, Frog Jump, House Robber.	
	2.2	2-D DP: Unique Paths, Maximum Path Sum.	
3.0		**DYNAMIC PROGRAMMING – PART II**	9
	3.1	Subsequence Problems: Subset Sum, Partition Equal Subset Sum, Count Subsets, Unbounded Knapsack.	
	3.2	Longest Common Subsequence (LCS), Longest Increasing Subsequence (LIS), Stock Trading Problems, Matrix Chain Multiplication.	
4.0		**GRAPH ALGORITHMS – PART I**	9
	4.1	Graph Representations: Adjacency Matrix, Adjacency List, Edge List, Traversals: BFS, DFS Applications, Number of Provinces, Number of Islands.	
	4.2	Cycle Detection (Directed and Undirected), Bipartite Graph, Topological Sort, Kahn's Algorithm, Alien Dictionary.	
5.0		**GRAPH ALGORITHMS – PART II**	9
	5.1	Shortest Paths: Dijkstra's Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm, Minimum Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Disjoint Sets (with Implementation).	
	5.2	Strongly Connected Components: Kosaraju's Algorithm, Tarjan's Algorithm, Articulation Points, Critical Connections.	

Text Books

1. Narasimha Karumanchi, Data Structures and Algorithmic Thinking with Python, CareerMonk Publications, 2020.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.
3. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, 3rd Edition, 2017.

Reference Books

1. Adnan Aziz, Tsung-Hsien Lee and Amit Prakash, Elements of Programming Interviews, 2nd Edition, 2012.
2. Gayle Laakmann McDowell, Cracking the Coding Interview, 6th Edition, CareerCup, 2015.
3. Steven S. Skiena, The Algorithm Design Manual, 3rd Edition, Springer, 2020.

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. The fifth module will be assessed for 5 marks separately (FET).

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 the 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
UXXMM0XX	Multidisciplinary Minor II	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	
UXXMM0XX	Multidisciplinary Minor II	-	10	5	-	15	35	40%	50

MULTIDISCIPLINARY : MINOR II (PRODUCTION OF AI SYSTEMS USING LLM)

Course Description

This course focuses on building real-world AI systems using Large Language Models (LLMs), covering end-to-end application development from prompt engineering to deployment. It introduces LLM-powered applications, conversational voice AI, and Text-to-SQL systems, along with time series forecasting for data-driven decision making. The course also explores optimization techniques such as quantization and model parallelism to improve performance and scalability. Students will learn how to design, deploy, and optimize production-ready AI systems with a focus on efficiency, scalability, and real-world applications .

Pre-requisites: Multidisciplinary I

Course Objectives

1. To introduce the fundamentals of building AI applications using Large Language Models and modern AI architectures.
2. To develop skills in conversational AI, speech systems, and Text-to-SQL applications.
3. To enable students to apply time-series forecasting techniques for predictive analytics.
4. To familiarize students with optimization techniques such as quantization and model parallelism in LLM systems.
5. To provide hands-on experience in designing and deploying scalable production-ready AI systems.

Course Outcomes:

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

CO1 **Explain**[1] the fundamentals of LLM-powered applications, architectures, and prompting techniques.

CO2 **Apply**[3] concepts of conversational AI, speech systems, and Text-to-SQL for building intelligent applications.

CO3 **Analyze**[4] time series data and develop forecasting models using modern AutoML techniques.

CO4 **Evaluate**[5] optimization techniques such as quantization and model parallelism for improving performance and scalability.

CO5 **Design**[6] and deploy production-ready LLM systems with optimized performance, scalability, and cost efficiency.

Course Content

Module	Unit	Description	Hours
1.0		**BUILDING LLM-POWERED AI APPLICATIONS & CONVERSATIONAL VOICE AI**	6
	1.1	Foundations of LLM-powered AI applications; AI application architecture; Prompting fundamentals; Integrating LLM APIs and frameworks; Designing end-to-end AI workflows.	
	1.2	Speech AI fundamentals; Speech-to-Text pipelines; Text-to-Speech systems; Conversational AI architecture; Integrating LLMs with speech systems; **Project:** AI Voice Assistant.	
2.0		**TEXT-TO-SQL SYSTEMS & TIME SERIES FORECASTING**	6
	2.1	Fundamentals of Text-to-SQL; Schema understanding for LLMs; Prompt engineering for SQL generation; Query validation and error handling; Evaluating SQL outputs.	
	2.2	Time series fundamentals; Data preprocessing and feature engineering; AutoML forecasting with AutoGluon; Forecast evaluation metrics; Designing production forecasting pipelines; **Projects:** Natural Language Data Analyst (Text-to-SQL System), Demand Forecasting System.	
3.0		**QUANTISATION FUNDAMENTALS: OPTIMIZING LLMS FOR SPEED & COST**	6
	3.1	Model optimization fundamentals; Quantization techniques; Post-training quantization; Quantization-aware training; Benchmarking performance improvements.	
	3.2	Implementation of Quantized LLM Deployment.	
4.0		**SCALING LLMS WITH MODEL PARALLELISM**	6
	4.1	Challenges in scaling large models; Data parallelism vs Model parallelism; Tensor parallelism; Pipeline parallelism; Distributed inference architectures.	
	4.2	**Project:** Scalable LLM Inference System.	
5.0		**OPTIMIZING LLM SYSTEMS FOR REAL-WORLD DEPLOYMENT**	6
	5.1	Inference optimization; Caching and batching strategies; Deployment architectures; Monitoring and performance tuning; Cost optimization strategies.	
	5.2	**Project:** Production-Ready LLM Application.	

Text Books

1. Alammar, J. & Grootendorst, M., Hands-On Large Language Models, O'Reilly Media, **2024**.
2. Tunstall, L., von Werra, L. & Wolf, T., Natural Language Processing with Transformers, O'Reilly Media, **2022**.
3. Moroney, L., AI and Machine Learning for Coders, O'Reilly Media, **2021**.

Reference Books

1. Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D., Machine Learning with PyTorch and Scikit-Learn, Packt, **2022**.
2. Chollet, F., Deep Learning with Python, **2nd ed.**, Manning Publications, **2021**.
3. Géron, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, **3rd ed.**, O'Reilly Media, **2022**.

Internal Assessment (T1, T2 and FET)

4. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
5. FET shall be assessed for 5 marks separately.

Semester End Examination

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

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADAE401	Advanced 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	
UADAE401	Advanced Aptitude	10	10	5	-	25	50	40%	75

ADVANCED APTITUDE

Course Description

This course builds a strong foundation in logical reasoning and quantitative aptitude. Students develop analytical thinking a nd decision-making skills through structured approaches to numerical, logical, and visual challenges. Topics include decision-making, logical sequences, input-output patterns, cryptarithmetics, game-based aptitude, functions, progressions, base arithmetic, algebraic structures, logarithms, geometry, mensuration, statistics, and trigonometry. The course prepares learners for competitive exams and technical recruitment assessments.

Pre-requisites: Quantitative Aptitude

Course Objectives

1. To build strong foundations in logical reasoning and quantitative aptitude for competitive examinations.
2. To enhance analytical thinking and structured problem-solving abilities.
3. To develop proficiency in numerical patterns, cryptarithmetics, and game-based aptitude challenges.
4. To enable students to apply algebraic, logarithmic, and quantitative concepts in problem-solving.
5. To build confidence in solving geometry, statistics, and trigonometric aptitude problems efficiently.

Course Outcomes:

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

CO1 **Identify**[1] decision-making scenarios, logical sequences, and argument structures in analytical reasoning.

CO2 **Describe**[2] input-output patterns, cryptarithmetic puzzles, and game-based reasoning problems.

CO3 **Apply**[3] quantitative techniques involving functions, progressions, simplification, and number system conversions.

CO4 **Analyze**[4] algebraic concepts including equations, polynomials, logarithms, and inequalities.

CO5 **Evaluate**[5] problems related to geometry, mensuration, statistics, and trigonometric heights and distances.

Course Contents

Module	Unit	Description	Hours
1.0		**DECISION-MAKING**	9
	1.1	Decision-Making: Assessing situations, evaluating multiple choices, and selecting the most appropriate course of action based on reasoning and judgment.	
	1.2	Logical Sequence: Arranging words, numbers, or events in a meaningful logical order. Statement and Arguments: Analyzing statements and determining whether the given arguments are strong or weak based on logic, relevance, and validity.	
2.0		**LOGICAL SIMULATION AND APPLIED REASONING**	9
	2.1	Input-Output: Pattern-based machine arrangements, identifying logic in stepwise transformations, word and number-based sequencing.	
	2.2	Cryptarithmetics: Solving arithmetic problems where digits are replaced by letters, finding numerical values based on logical deductions. Game-Based Aptitude: Logical and analytical reasoning through interactive puzzles, grid-based challenges, pattern recognition, and strategic problem-solving in simulated environments.	
3.0		**FUNCTIONS, SERIES, AND BASE ARITHMETIC**	9
	3.1	Functions: Definition and evaluation of functions, functional equations, composite functions, and inverse functions. Progressions: Arithmetic Progression (AP), Geometric Progression (GP), Harmonic Progression (HP), sum of series, nth term calculation, and applications.	
	3.2	Simplifications: Solving mathematical expressions using BODMAS/PEMDAS rule, surds and indices, approximation techniques, and number system-based calculations. Base System: Number system conversions (binary, octal, decimal, hexadecimal), arithmetic operations in different bases, and related problems.	
4.0		**ALGEBRAIC STRUCTURES AND LOGARITHMIC CONCEPTS**	9
	4.1	Equations: Solving linear and quadratic equations, simultaneous equations, factorization, and word problems involving equations. Algebra: Polynomials, algebraic identities, factorization, inequalities, and algebraic expression simplification.	
	4.2	Logarithms: Logarithmic identities, solving logarithmic equations, properties of logarithms, and applications in calculations. Surds and Indices: Simplification of surds, laws of indices, exponent rules, rationalization, and solving exponent-based problems.	
5.0		**QUANTITATIVE GEOMETRY AND TRIGONOMETRIC APPLICATIONS**	9
	5.1	Basic Geometry: Properties of lines, angles, triangles, circles, polygons, perimeter, area, volume, and coordinate geometry fundamentals. Mensuration: Area, perimeter, surface area, and volume for 2D and 3D geometric shapes.	
	5.2	Statistics: Measures of central tendency (mean, median, mode), measures of dispersion (variance, standard deviation), data representation, and probability basics. Heights and Distances: Applications of trigonometry, angle of elevation and depression, solving problems using sine, cosine, and tangent ratios.	

Text Books

1. R. S. Aggarwal, Quantitative Aptitude for Competitive Examinations, S. Chand Publishing, Revised Edition, 2024.
2. Arun Sharma, How to Prepare for Quantitative Aptitude for the CAT, McGraw-Hill Education, 8th Edition, 2021.
3. Nishit K. Sinha, Logical Reasoning and Data Interpretation for the CAT, Pearson Education, 5th Edition, 2021.

Reference Books

1. Arihant Experts, Fast Track Objective Arithmetic, Arihant Publications, 2022.
2. Abhijit Guha, Quantitative Aptitude for Competitive Examinations, McGraw-Hill Education, 7th Edition, 2018.
3. Dinesh Khattar, The Pearson Guide to Quantitative Aptitude for Competitive Examinations, Pearson Education, 3rd Edition, 2020.

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. Tutorial assessment shall be based on accuracy, structured reasoning, and problem-solving approach, for 25 marks.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 5 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 the entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UAROE0XX	Open Elective II	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	
UAROE0XX	Open Elective II	-	-	-	50	50	-	40%	50

OPEN ELECTIVE II: LANGUAGE ANALYTICS

Course Description

This course focuses on improving students' command of English through practical skill-based units in vocabulary, comprehension, writing, and communication. The course includes structured language practice and reasoning tasks designed to enhance accuracy, fluency, and confidence in professional use. It also provides targeted skill practice modules aligned with the B2 Cambridge English qualification, helping students prepare effectively for international certification while strengthening their academic and workplace communication abilities. The course also equips learners with the linguistic accuracy, analytical reasoning, and communication competence required to perform effectively in verbal ability assessments conducted by companies for technical recruitment.

Pre-requisites: Communicative English Advanced

Course Objectives

1. To strengthen students' vocabulary and word usage for clear and accurate communication.
2. To develop reading and comprehension skills for academic and professional purposes.
3. To train students in logical sentence construction and error correction.
4. To build effective writing skills for academic and workplace communication.
5. To prepare students for professional interactions, interviews, and real-world communication contexts.

Course Outcomes:

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

CO1 **Recall**[1] appropriate vocabulary and apply precise word usage in academic and professional contexts.

CO2 **Describe**[2] key ideas, tone, and inference strategies for accurate reading and interpretation of texts.

CO3 **Apply**[3] grammar rules and sentence logic to identify and correct errors in written communication.

CO4 **Analyze**[4] writing requirements to produce well-structured documents for formal and workplace contexts.

CO5 **Evaluate**[5] professional communication practices in interviews, presentations, and group discussions.

Course Contents

Module	Unit	Description	Hours
1.0		**VOCABULARY AND WORD PRACTICE**	6
	1.1	Synonyms and Antonyms, One-word Substitutions, Phrasal Verbs, Word Roots and Affixes, Homonyms and Confusing Word Pairs.	
	1.2	Idioms and Expressions used in the Workplace, Vocabulary Games and Contextual Usage Exercises, Word Association and Collocations.	
2.0		**READING AND COMPREHENSION PRACTICE**	6
	2.1	Short Passage Reading, Identifying Main and Supporting Ideas, Tone and Attitude Analysis, Inference and Conclusion Questions.	
	2.2	Paragraph Organization Tasks, Reading for Implied Meaning, Identifying the Author's Purpose, Timed Comprehension Practice.	
3.0		**SENTENCE LOGIC AND ERROR CORRECTION**	6
	3.1	Sentence Rearrangement and Sequencing, Identifying Grammatical and Logical Errors, Spotting Sentence Improvements, Practice with Connectors and Transitions.	
	3.2	Rewriting Faulty Sentences, Exercises on Clause Combination, Cohesion and Coherence, Correcting Sentence Structure in Emails and Reports.	
4.0		**WRITING SKILLS FOR STUDY AND WORK**	6
	4.1	Paragraph and Essay Writing, Report and Summary Writing, Paraphrasing Practice, Email and Letter Writing.	
	4.2	Editing and Proofreading Drills, Formatting Academic and Workplace Documents, Writing for Clarity and Tone, Active and Passive Voice in Professional Contexts.	
5.0		**COMMUNICATION AND CAREER READINESS**	6
	5.1	Resume and Cover-Letter Writing Practice, Mock Interviews and Group Discussions, Oral Presentations and Elevator Pitches.	
	5.2	Role-plays for Professional Communication, Non-verbal Cues and Body-language Practice, Workplace Conversation Drills.	

Text Books

1. Hewings, M., Thaine, C., and McCarthy, M., Cambridge Academic English: An Integrated Skills Course for EAP (Intermediate-Advanced), Cambridge University Press, 2012.
2. Zemach, D. E., Skillful: Reading and Writing; Listening and Speaking (Levels 1-4), Macmillan Education, 2013.
3. Arun Sharma and Meenakshi Upadhyay, Verbal Ability and Reading Comprehension for the CAT, McGraw-Hill Education, 4th Edition, 2020.

Reference Books

1. Slaght, J., English for Academic Study: Reading, Writing, Listening, Speaking, Vocabulary and Grammar (Series), Garnet Education, 2012.
2. Nishit K. Sinha, The Pearson Guide to Verbal Ability for the CAT, Pearson Education, 2021.
3. Raymond Murphy, English Grammar in Use, Cambridge University Press, 5th Edition, 2019.

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.

Semester End Examination

1. Question paper will comprise of 5 questions, each carrying 5 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 the entire syllabus.
5. Remaining questions (Q.2 to Q.5) will be selected from all the modules.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
USTBS401	Linear Algebra and Calculus	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	
USTBS401	Linear Algebra and Calculus	10	10	5	-	25	50	40%	75

LINEAR ALGEBRA AND CALCULUS

Course Description

This course covers foundational concepts in linear algebra and calculus with applications in data science and machine learnin g. Students explore vectors, matrices, vector spaces, eigenvalues, eigenvectors, and matrix decomposition techniques including SVD. The course also covers theory of equations, differential calculus, and multivariable calculus, including partial derivatives, gradients, Jacobians, Hessians, and Lagrange multipliers for optimization problems.

Course Objectives

1. To introduce the concepts of matrices, vectors, and linear transformations.
2. To develop problem-solving skills using differentiation and integration techniques.
3. To enable students to apply linear algebra concepts in computational and engineering problems.
4. To familiarize students with multivariable calculus and optimization methods.
5. To lay the foundation of understanding mathematical techniques used in AI, machine learning, and data science.

Course Outcomes:

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

CO1 **Recall**[1] core linear algebra concepts, including vectors, matrices, eigenvalues, and eigenvectors, and their applications in Machine Learning and Data Science.

CO2 **Explain**[2] polynomial equations, root-coefficient relationships, and root transformation techniques.

CO3 **Apply**[3] determinants, matrix inverses, diagonalization, and singular value decomposition (SVD) to solve data analysis problems.

CO4 **Analyze**[4] differential calculus concepts, including limits, derivatives, and partial derivatives, to analyze mathematical functions.

CO5 **Evaluate**[5] optimization problems using Newton-Raphson method, Lagrange multipliers, Jacobians, and Hessians.

Course Contents

Module	Unit	Description	Hours
1.0		**INTRODUCTION TO LINEAR ALGEBRA AND FUNDAMENTALS**	9
	1.1	Importance of Linear Algebra in Machine Learning and Data Science, Vectors and Scalars, Dot Product, Cross Product.Vector Spaces, Basis and Dimension.	
	1.2	Vector Spaces, Basis and Dimension, Matrix Operations, Matrix Properties, Systems of Linear Equations, Row Reduction Techniques,matrix subspace & rank theorem. ,Solving system of Linear equations using Numerical Methods, rank & linear independence, Eigenvalues and Eigenvectors of a Real Matrix.	
2.0		**THEORY OF EQUATIONS**	9
	2.1	Introduction to Algebraic Equations, Surds and Irrational Roots, Equations whose Roots are in AP, GP, and HP.	
	2.2	Relations between Roots and Coefficients, Symmetric Functions, Formation of Equations, Decreasing and Increasing the Roots, Transformation of Equations, Reciprocal Equations.	
3.0		**ADVANCED LINEAR ALGEBRA AND MATRIX DECOMPOSITIONS**	9
	3.1	Determinants and Matrix Inverses, Properties and Computation of Eigenvalues, Diagonalization.	
	3.2	Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Applications of Matrix Decompositions in Machine Learning and Data Science.	
4.0		**DIFFERENTIAL CALCULUS**	9
	4.1	Limits of Functions, One-Sided Limits, continuity & discontinuity, Cartesian and Polar Coordinates.	
	4.2	Functions of Two Variables, Implicit Functions,Derivatives and Tangents, properties of derivatives, increasing, decreasing functions & Maxima Minima, Sketching of graph	
5.0		**MULTIVARIABLE DIFFERENTIAL CALCULUS**	9
	5.1	Partial Derivatives, Gradients, Chain Rule, Gradient Vectors, Jacobian Matrix.	
	5.2	Hessian Matrix, Newton's method for optimization, Multi-variable Optimization, Total Differential, Lagrange's Multiplier Method.	

Text Books

1. Gilbert Strang, Introduction to Linear Algebra, Wellesley-Cambridge Press, 6th Edition, 2023.
2. Joel R. Hass, Christopher E. Heil, Maurice D. Weir, and Przemyslaw Bogacki, Thomas' Calculus: Early Transcendentals, Pearson Education, 15th Edition, 2023.
3. M. P. Deisenroth, A. A. Faisal, and C. S. Ong, Mathematics for Machine Learning, Cambridge University Press, 2020.

Reference Books

1. David C. Lay, Steven R. Lay, and Judi J. McDonald, Linear Algebra and Its Applications, Pearson Education, 6th Edition, 2021.
2. A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, Dive into Deep Learning, 1st Edition, 2019.

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. Tutorial assessment shall be based on the overall performance of the student.

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 the 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 (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC404	Operating Systems Laboratory	-	2	-	-	1	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
UADPC404	Operating Systems Laboratory	25	40%	-	40%	25

OPERATING SYSTEMS LABORATORY

Course Outcomes:

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

CO1 **Apply**[3] Linux system calls and file I/O operations for process-level operating system interaction.

CO2 **Implement**[3] process creation, synchronization, and inter-process communication (IPC) using system programming APIs.

CO3 **Analyze**[4] CPU scheduling algorithms by simulating and comparing their performance.

CO4 **Evaluate**[5] deadlock avoidance, page replacement, and disk scheduling algorithms for efficient resource management.

List of Experiments

1. Implement I/O system calls using open(), read(), write(), and close().
2. Implement process creation using fork(), exec(), and wait() system calls.
3. Demonstrate parent and child process communication and synchronization using pipes.
4. Implement process synchronization using semaphores.
5. Demonstrate the use of POSIX threads using pthread_create and pthread_join.
6. Implement the Producer-Consumer problem using semaphores.
7. Implement the Dining Philosophers problem using mutexes or semaphores.
8. Implement CPU scheduling algorithms: FCFS and Shortest Job First (SJF).
9. Implement CPU scheduling algorithms: Round Robin and Priority Scheduling.
10. Simulate the Banker's Algorithm for deadlock avoidance.
11. Implement page replacement algorithms: FIFO and Least Recently Used (LRU).
12. Implement disk scheduling algorithms: FCFS and SCAN.

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 (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC405	Computer Networks Laboratory	-	2	-	-	1	-

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
UADPC405	Computer Networks Laboratory	25	40%	-	40%	25

COMPUTER NETWORKS LABORATORY

Course Outcomes:

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

CO1 **Apply**[3] network topologies and protocols using simulation tools to design and test network configurations.

CO2 **Implement**[3] socket programming for TCP and UDP-based client-server communication.

CO3 **Analyze**[4] network traffic and protocol behavior using packet capture and analysis tools such as Wireshark.

CO4 **Evaluate**[5] network configuration, subnetting, and routing concepts in practical networking environments.

List of Experiments

1. Configure and test a basic network topology using a network simulator (Cisco Packet Tracer or GNS3).
2. Implement a TCP client-server program using socket programming in Python or C.
3. Implement a UDP client-server program and compare it with TCP communication.
4. Use Wireshark to capture and analyze HTTP and DNS network traffic.
5. Implement IP subnetting and configure static routes between network segments.
6. Simulate and analyze the OSI model layer-by-layer using a network simulator.
7. Implement a simple file transfer application using FTP concepts.
8. Configure and test DHCP and DNS services in a simulated environment.
9. Implement error detection using CRC and checksum algorithms.
10. Simulate routing protocols (RIP or OSPF) in a multi-router network environment.
11. Demonstrate ARP and ICMP operations using ping and traceroute commands.
12. Implement a basic HTTP client using socket programming to fetch web page content.

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 (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UADPC406	Advanced Algorithms & Optimization Laboratory	-	2	-	-	1	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
UADPC406	Advanced Algorithms & Optimization Laboratory	25	40%	-	40%	25

ADVANCED ALGORITHMS & OPTIMIZATION LABORATORY

Course Outcomes:

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

CO1 **Apply**[3] algorithmic patterns including brute force, two pointers, sliding window, and binary search techniques.

CO2 **Implement**[3] linked list, stack, queue, and string manipulation algorithms to solve data structure problems.

CO3 **Analyze**[4] recursive and backtracking solutions and compare their efficiency with iterative approaches.

CO4 **Evaluate**[5] tree, graph, dynamic programming, and greedy algorithms for solving diverse computational problems.

List of Experiments

1. Implement Fractional Knapsack using a greedy approach and analyze time complexity.
2. Solve the Activity Selection problem and Job Scheduling problem using greedy strategy.
3. Implement Minimum Coins problem and Platform Scheduling using greedy techniques.
4. Solve Climbing Stairs and Frog Jump using both memoization and tabulation.
5. Implement House Robber (1-D DP) and compare recursive vs DP approach.
6. Solve Unique Paths and Maximum Path Sum using 2-D DP.
7. Implement Subset Sum and Partition Equal Subset Sum using DP.
8. Solve Unbounded Knapsack and Count Subsets with Sum using tabulation.
9. Implement Longest Common Subsequence (LCS) and Longest Increasing Subsequence (LIS).
10. Solve Stock Trading problems (Buy and Sell Stock - multiple variants) using DP.
11. Implement BFS and DFS on a graph; solve Number of Provinces and Number of Islands.
12. Implement Cycle Detection in Directed and Undirected graphs.
13. Check if a graph is Bipartite; implement Topological Sort using DFS and Kahn's Algorithm.
14. Implement Dijkstra's Algorithm and Bellman-Ford Algorithm for shortest paths.
15. Implement Prim's and Kruskal's Algorithm for Minimum Spanning Tree with Disjoint Set Union (DSU).
16. Implement Kosaraju's or Tarjan's Algorithm to find Strongly Connected Components.
17. Find Articulation Points and Critical Connections in a graph.
18. Solve an end-to-end problem combining DP and graph techniques.
19. Compare greedy and DP approaches for the Knapsack problem variants.
20. Solve a real-world problem modelled as a graph (e.g., shortest delivery route, course scheduling).

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.