Sanjay Ghodawat University Kolhapur



School of Engineering and Technology

Department of Artificial Intelligence and Machine Learning

B. Tech. Artificial Intelligence and Machine Learning

Curriculum Structure (R25)

Semester: III-IV

Programme Structure for B. Tech.: Second Year Artificial Intelligence and Machine Learning Semester III: Teaching Scheme

Course code	Course name	Teaching scheme(Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
UAIPC301	Computer Organization and Architecture	3	-	-	3	-	-	3
UAIPC302	Data Structure using Python	3	-	-	3	-	-	3
UAIPC303	Introduction to AI and Generative AI	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor I	2	-	-	2	-	-	2
UXXOE01X	Open Elective I	2	-	-	2	-	-	2
UCMEM001	Engineering Economics	2	-	-	2	-	-	2
UAIPC304	Data Structure using Python Lab	-	2	-	-	1	-	1
UAIPC305	Introduction to AI and Generative AI Lab	-	2	-	-	1	-	1
UAIPC306	Object Oriented Programming using JAVA	-	4	-	-	2	-	2
UAIVE003	Advanced Web Designing Lab	-	4	-	-	2	-	2
UHSAE008	Employability Skills	1	2	-	-	2	-	2
Total		**16**	**14**	**-**	**15**	**8**	**-**	**23**

Semester III: Evaluation Scheme

Course code	Course Name	Theory Marks					Practical Marks		Total
		Internal Assessment				SEE	Term work	Practical Oral/POE	
		T1	T2	FET	Total				
UAIPC301	Computer Organization and Architecture	10	10	5	25	50	-	-	75
UAIPC302	Data Structure using Python	10	10	5	25	50	-	-	75
UAIPC303	Introduction to AI and Generative AI	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor I	-	10	5	15	35	-	-	50
UXXOE01X	Open Elective I	-	10	5	15	35	-	-	50
UCMEM001	Engineering Economics	-	10	5	15	35	-	-	50
UAIPC304	Data Structure using Python Lab	-	-	-	-	-	25	-	25
UAIPC305	Introduction to AI and Generative AI Lab	-	-	-	-	-	25	-	25
UAIPC306	Object Oriented Programming using JAVA	-	-	-	-	-	25	25	50
UAIVE003	Advanced Web Designing Lab	-	-	-	-	-	25	25	50
UHSAE008	Employability Skills	-	-	-	-	-	25	25	50
								Total	**575**

Programme Structure for B. Tech.: Second Year Artificial Intelligence and Machine Learning (R25)

IV: Teaching Scheme

Course code	Course Name	Teaching scheme(Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
UAIPC401	Software Engineering	3	-	1	3	-	1	4
UAIPC402	Operating System	3	-	-	3	-	-	3
UAIPC403	Database Management System	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
UXXOE02X	Open Elective II	2	-	-	2	-	-	2
UCMEM003	Project Management	2	-	-	2	-	-	2
UAIVS401	Software Proficiency I (R Programming)	-	4	-	-	2	-	2
UAIPC405	Database Management System Lab	-	2	-	-	1	-	1
UAIFP401	Field Project	-	4	-	-	2	-	2
USTVE005	AI Ethics & Governance	1	-	1	1	-	1	2
Total		**16**	**10**	**2**	**16**	**5**	**2**	**23**

Semester IV: Evaluation Scheme

Course code	Course Name	Theory Marks					Practical Marks		Total
		Internal Assessment				SEE	Term work	Practical Oral/POE	
		T1	T2	FET	Total				
UAIPC401	Software Engineering	10	10	5	25	50	25	-	100
UAIPC402	Operating System	10	10	5	25	50	-	-	75
UAIPC403	Database Management System	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor II	-	10	5	15	35	-	-	50
UXXOE02X	Open Elective II	-	10	5	15	35	-	-	50
UCMEM003	Project Management	-	10	5	15	35	-	-	50
UAIVS401	Software Proficiency I (R Programming)	-	-	-	-	-	25	25	50
UAIPC405	Database Management System Lab	-	-	-	-	-	25	-	25
UAIFP401	Field Project	-	-	-	-	-	25	25	50
USTVE005	AI Ethics & Governance	10	10	5	25	-	25	-	50
								Total	**575**

Exit Option:

- Students will have the flexibility to enter a programme in odd semesters and exit a programme after the successful completion of even semesters as per their future career needs.
- Students exiting the Second Year Programme after securing minimum 80 credits will be awarded UG Diploma in the relevant Discipline/Subject provided they secure additional 8 credits in skill-based vocational courses(skill-based courses, internship, mini projects etc) offered during summer vacation after the second year.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UAIPC301	Computer Organization and Architecture	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		3	-	-	3	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		Total (Marks)
UAIPC301	Computer Organization and Architecture	T1	T2	FET	Total	Min pass	Marks	Min pass	75
		10	10	5	25	40%	50	40%	

Course Description: This course introduces the fundamental concepts of computer organization and architecture, focusing on the structure and operation of modern computer systems. It covers the arrangement and functioning of functional units, arithmetic and logic unit design, CPU architecture, instruction execution, and control unit design. The course also explores memory hierarchy, cache and virtual memory techniques, and input/output organization. Additionally, it provides an overview of parallel processing, pipelining, and multiprocessor configurations to understand performance enhancement in computing systems.

Course Objectives:

- To introduce the basic structure and organization of a computer system.
- To help students understand arithmetic operations and algorithms used in the Arithmetic Logic Unit (ALU).
- To develop awareness of CPU architecture, instruction execution, and control unit design.
- To equip students with knowledge of memory organization, including cache and virtual memory techniques.
- To promote understanding of input/output (I/O) organization and parallel processing concept

Course Outcomes: After the successful completion of the course students will able to:

CO1 **Explain**[2] the organization of a computer system, its functional units, and bus structures including the Von Neumann architecture.

CO2 **Apply**[3] arithmetic algorithms such as Booth's multiplication and division techniques for solving binary operations.

CO3 **Illustrate**[3] CPU architecture, instruction execution cycle, and control unit design

CO4 **Examine**[4] memory hierarchy, cache organization, and virtual memory techniques for Efficient data storage and access.

CO5 **Analyze**[4] various input/output techniques and parallel processing concepts

Course Contents

Module	Unit	Description	Hours
1.0		**Basic structure of computer**	
1	1.1	Introduction of computer system and its sub modules, Basic organization of computer and block level description of the functional Units. Von Neumann model	9
	1.2	Introduction to buses and connecting I/O devices to CPU and memory, Asynchronous and synchronous bus, PCI, SCSI. **Case Study:** Computer System in an Online Banking System	
2.0		**Arithmetic and Logic Unit**	
2	2.1	Arithmetic and logical unit hardware implementation, Booth's Recoding, Booth's algorithm for signed multiplication	9
	2.2	Restoring division and non-restoring division algorithm, IEEE floating point number representation and operations. **Case Study:** High-Speed Arithmetic Processing in Calculators, implementing fast multiplication and division using Booth's algorithm and floating-point operations.	
3.0		**Central processing unit.**	
3	3.1	CPU architecture, Register Organization, Instruction formats and addressing modes (Intel processor), Basic instruction cycle, Instruction interpretation and sequencing, Control Unit operation, Hardwired control unit design methods and design examples	9
	3.2	Multiplier control unit, Micro programmed control unit, basic concepts, Microinstruction sequencing and execution, Micro operations, concepts of nano programming, Introduction to RISC and CISC architectures, design issues and examples of RISC processors **Case Study:** Instruction Execution in Intel Processors, analyzing instruction cycle, addressing modes, and control unit design for efficient processing.	
4.0		**Memory Organization.**	
4	4.1	Characteristics of memory system and hierarchy, concepts of semiconductor memories, main memory, ROM, EPROM, RAM,SRAM, DRAM, SDRAM, RDRAM, Flash memory, Stack Organization	9
	4.2	High speed memories: Cache memory organization and mapping, replacement algorithms, cache coherence, Interleaved and associative memories, Virtual memory, main memory allocation, segmentation paging, Secondary storage, RAID, optical memory, CDROM, DVD. **Case Study :** Memory Management in ATMs	
5.0		**I/O Organization and Multiprocessor Configurations.**	
5	5.1	Input /Output systems, Programmed I/O, Interrupt driven I/O, I/O channels, DMA, Peripheral Devices, U.S.B.	9
	5.2	Flynn's classifications, parallel processing concepts, Introduction to pipeline processing and pipeline hazards, design issues of pipeline architecture, Instruction pipeline, Instruction level parallelism and advanced issues. **Case study:** Multi-core System	

Text Books:

1. Miles Murdocca, "Computer Architecture and Organization", Wiley India, Standard Edition,2007.

2. William Stallings, "Computer Organization and Architecture: Designing and performance": Prentice-Hall India,12th Edition,2022

References:

1. Andrew S. Tanenbaum," Structured Computer Organization", Prentice-Hall India, 6th Edition,2021

- **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. 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 end semester 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 question (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
UAIPC302	Data Structure using Python	3	-	-	3	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		Total (Marks)
		T1	T2	FET	Total	Min pass	Marks	Min pass	
UAIPC302	Data Structure using Python	10	10	5	25	40%	50	40%	75

Course Description: The Data Structures course provides students with a foundational understanding of organizing and managing data efficiently in computer systems. Students will explore various data structures and algorithms used to store, retrieve, and manipulate data, gaining insight into their strengths, weaknesses, and applications. Through a combination of theoretical concepts and practical implementations, students will develop the skills necessary to analyze problems and design efficient data structures and algorithms for solving them.

Pre-requisites: Python Programming

Course Objectives:

- To help students understand the fundamental concepts of data, data structures, abstract data types, and algorithm analysis.
- To develop skills in problem-solving using linear data structures such as arrays and linked lists.
- To familiarize students to stack and queue data structures along with their real-world applications.
- To impart basic knowledge of searching, sorting, and hashing techniques for efficient data organization and retrieval.
- To develop awareness of nonlinear data structures such as trees and graphs for solving complex computational problems.

Course Outcomes: After the successful completion of the course students will able to:

CO1 **Explain**[2] primitive and non-primitive data types, abstract data types, and operations on data structures.

CO2 **Explain**[2] the concepts and operations of linear data structures such as arrays and linked lists.

CO3 **Describe**[2]the working principles of stack and queue data structures and their applications.

CO4 **Apply**[3]searching, sorting, and hashing techniques to solve computational problems.

CO5 **Analyze**[4]nonlinear data structures such as trees and graphs and compare their traversal techniques and applications.

Course Contents

Module	Unit	Description	Hours
1.0		**Basics of Data Structures**	9
	1.1	Introduction: Data and Information, Data Structure, Classification of Data Structures, Primitive Data Types, Non-Primitive Data Types, Abstract Data Types, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm.	
	1.2	Array: Introduction, One-Dimensional Array, Memory Representation of One-Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two-Dimensional Arrays, Advantages and Limitations of Arrays, Application of Array. **Case Study**: 1. Student Records Management System, storing and sorting student data using arrays. 2. Employee Records Database Optimization, finding employees based on salary or sorting them by department using Array	
2.0		**Linked List**	
	2.1	Linked List, Implementation of Singly Linked List (SLL), Operation(Insertion and Deletion) in SLL, copying a List into Other List, Merging Two Linked Lists Reversing One-way Linked List, Circular Linked List (CLL) and its operations	9
	2.2	Implementation of Doubly Linked List (DLL), Operations (Insertion, Deletion, searching) of DLL. Application of Linked List (SLL, CLL,DLL) **Case Study:** 1.Music Playlist Manager, dynamically adding, deleting, and rearranging songs using linked lists. 2.Growing employee database dynamically using LL.	
3.0		**Stack and queue**	
	3.1	Stack: Introduction, Operations on the Stack, Memory Representation of Stack, Array Representation of Stack, Linked List representation of stack, Applications of Stack: Evaluation of Arithmetic Expression, Matching Parenthesis, Infix and postfix operations, Recursion.	9
	3.2	Queue: Introduction to Queue, Operations on the Queue, Memory Representation of Queue: Array representation of Queue, Linked List Representation of Queue, Circular Queue, and Some special kinds of Queues: Deque, Priority Queue, and Applications of Queues **Case study:** 1**.**Backtracking algorithmic strategy, Use of stack in backtracking**.** **2.** Job scheduling using priority queue.	
4.0		**Searching and Hashing**	
	4.1	Bubble, Selection, Insertion, Merge Sort. Heap, Memory representation of Heap, Operation on Heap, Heap Sort. Searching: Sequential, binary, Indexed Sequential Searches, Binary Search.	9
	4.2	Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and Rehashing. **Case Study:** Online Search Engine Indexing, fast data retrieval using searching and hashing techniques.	
5.0		**Nonlinear Data Structure**	

	5.1	Tree: Tree, Binary Tree, Properties of Binary Tree, Memory Representation of Binary Tree, Operations Performed on Binary Tree, Reconstruction of Binary Tree from its Traversals, Huffman Algorithm, Binary Search Tree, Operations on Binary Search Tree, Advanced Tree Structures: Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, Operations performed on AVL Tree, B-Tree, Applications of Tree.	9
	5.2	Graph: Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List/Linked Representation of Graph, Operations Performed on Graph, Graph Traversal (BFS, DFS), Applications of the Graph **Case Study:** 1.Route Navigation System, finding shortest paths and connections using trees and graphs. 2. Compare the complexity of BST and Linear search	

Text Books

1. Narasimha Karumanchi, "Data Structures and Algorithms Made Easy in Python", CareerMonk Publications, Latest Edition, 2023
2. Reema Thareja, "Data Structures Using Python", Oxford University Press India, Latest Edition, 2023.

References

1. A. M. Tenenbaum, Y. Langsam, M. J. Augenstein, "Data Structures Using C", Prentice-Hall of India Pvt. Limited, Latest Edition, 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.

Semester End Exam

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 question (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
UAIPC303	Introduction to AI and Generative AI	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam		Total (Marks)
		T1	T2	FET	Total	Min. Passing %	Marks	Min pass	
UAIPC303	Introduction to AI and Generative AI	10	10	5	25	40%	50	40%	75

Course Description: This course provides a comprehensive introduction to the fundamental concepts, techniques, and applications of Artificial Intelligence (AI) along with an overview of emerging Generative AI technologies. It enables students to understand how intelligent systems are designed to perceive, reason, learn, and make decisions in complex environments, covering core topics such as intelligent agents, problem-solving through search strategies, knowledge representation, reasoning under uncertainty, and basic optimization techniques. The course also introduces modern advancements in Generative AI, including models such as Variational Autoencoders (VAE), Generative Adversarial Networks (GANs), and diffusion models

Pre-requisites: Data Structures Using C++

Course Objectives

- To introduce the core concepts, principles, and methodologies of Artificial Intelligence.
- To promote understanding of real-world problems addressed using Artificial Intelligence techniques.
- To impart basic knowledge of Artificial Intelligence algorithms and problem-solving strategies for intelligent solutions.
- To help students understand Artificial Intelligence-based approaches used in different application domains.

Course Outcomes: After the successful completion of the course students will able to:

CO1	**Explain**[2]fundamental concepts, perspectives, evolution, applications, and ethical aspects of Artificial Intelligence.
CO2	**Explain**[2] the structure, types, and environmental characteristics of intelligent agents used in AI systems.
CO3	**Apply**[3]search strategies, optimization techniques, and reasoning methods for problem solving, planning, and decision making in AI..
CO4	**Analyze**[4]probabilistic reasoning models and constraint satisfaction techniques for solving AI problems under uncertainty
CO5	**Explain**[2] the fundamental concepts and types of generative AI models, including Variational Autoencoders (VAE), Generative Adversarial Networks (GANs), and diffusion models**.**

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction To AI:**	**9**
1	1.1	Artificial Intelligence (AI), AI Perspectives: Acting and Thinking humanly, Acting and Thinking rationally	
	1.2	History of AI, Applications of AI, The present state of AI, Ethics in AI **Case Study:** Kroger: How This U.S. Retail Giant Is Using AI And Robots To Prepare For The 4th Industrial Revolution	
2.0		**Intelligent Agents**	**9**
2	2.1	Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents , Types of Agents: Simple Reflex, Model Based, Goal Based, Utility Based Agents	
	2.2	Environment Types: Deterministic, Stochastic, Static, Dynamic, Observable, Semi-observable, Single Agent, Multi Agent **Case Study:** 4th Industrial Revolution Using AI	
3.0		**Search Optimization & Planning:**	**9**
3	3.1	Problem spaces and search, Knowledge and rationality, Heuristic search Informed (BFS, A*) & Uniformed Search (DFS, BFS) strategies, Search and optimization (gradient descent)	
	3.2	Adversarial search, Planning and scheduling, Manufacturing scheduling, Logic and inference, Bayesian reasoning, Temporal reasoning **Case Study:** Machine Learning At Google: The Amazing Use Case Of Becoming A Fully Sustainable Business	
4.0		**Introduction to Generative AI**	**9**
4		Definition and significance of generative models, Evolution and milestones in Generative AI, Discriminative & Generative models, Prompt-based generation overview, LLM	
		Types of Generative Models: Variation Auto encoders (VAE), Generative Adversarial Networks (GANs), Diffusion models. **Case Study:** AI Content Generation System, using generative models like GANs, VAEs, diffusion models, and LLMs to create text, images, and multimedia content based on user prompts.	
5.0		**Generative AI Applications:**	**9**
5	5.1	Applications: Art and Creativity, Image and Video Generation, Text Generation, Music Composition, Healthcare , Finance. Real-world use cases and challenges in deploying generative AI models	

Text Books

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson Education, 4th Edition, 2021.
2. Numa Dhamani, “Introduction to Generative AI”, Kindle Edition, 2024.

References

1. Kevin Knight and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, McGraw-Hill, 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.

Semester End Exam

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 question (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
UCMEM001	Engineering Economics	02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme(In Semester)					Semester End Exam(SEE)		Total (Marks)
		T1	T2	FET	Total	Min pass	Marks	Min pass	
UCMEM001	Engineering Economics	-	10	5	15	40%	35	40%	50

Course Description

The course focuses on the economic aspects of engineering projects, providing insights into techniques and methods used in feasibility studies related to design and implementation. Its primary purpose is to build a solid understanding of engineering economy concepts and principles, while developing proficiency in rational decision-making methods for problems encountered in professional practice.

Pre-requisites: - mathematics

Course Objectives:

- To introduce the role of economics in engineering practice by connecting financial principles with technical decision-making.
- To help students understand macroeconomic concepts such as GDP, GNP, NNP, per capita income, inflation, economic policies, and business cycles, and their relevance to engineering projects.
- To impart basic knowledge of investment evaluation through comparative methods for identifying the most viable alternatives.
- To enable students to understand and apply depreciation techniques in engineering projects.
- To enhance awareness of financial management by exploring working capital strategies

Course Outcomes: After the successful completion of the course students will able to:

CO1	**Apply**[3] economic principles to engineering decision-making and project feasibility.
CO2	**Analyze**[4] national income concepts, inflation, and business cycles in an engineering context.
CO3	**Apply**[3] economic evaluation methods to select optimal engineering investment alternatives.
CO4	**Evaluate**[5] financial viability using payback, benefit-cost, and break-even techniques.
CO5	**Analyze**[4] depreciation methods and working capital management for financial planning.

Course Contents

Module	Unit	Description	Hours
1.0		**Fundamentals of Engineering Economics**	**06**
	1.1	Meaning, Nature, Scope and importance of Engineering Economics, Role of economics in engineering decision-making, Demand & Supply- Concepts, Time value of money, Cash flow, cash flow diagrams, simple Interest, and compound Interest.	
2.0		**Macro economics in Engineering**	**06**
	2.1	National income concepts: GDP, GNP, NNP, per Capita income, Measurement of National Income, Inflation: types, causes, effects, and control measures, Monetary policy, and fiscal policy: tools and implications, Business cycles, and their relevance to engineering projects.	
3.0		**Evaluating Investment Choices**	**06**
	3.1	Present worth comparisons, Comparisons of assets with equal, unequallives, comparison of deferred investments, Annual worth method, Rate of return method, Future worth method, NPV Method.	
4.0		**Decision Tools for Financial Viability**	**06**
	4.1	Payback period method-conventional and discounted payback period method, Benefit cost ration methods, break even analysis method.	
5.0		**Depreciation Dynamics &Working Capital Management**	**06**
	5.1	Definition, methods for calculating depreciation, straight line method, constant percentage method, sinking fund method ,Meaning and need& components of working capital management.	

Textbooks:

1. N. Gregory Mankiw, "Microeconomics", South-Western Cengage Learning, 9th Edition, 2020.
2. R.R.Gupta,Engineering Economics and financial Management,New Delhi,S.Chand Pulishing,2022

Reference:

1.Panneer Selvam R., "Engineering Economics", Prentice Hall of India Ltd., New Delhi, 2001

Internal Assessment (T1, T2 and FET)

1. T2 should be based on Third to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 07 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UAIPC304	**Data Structure using Python Lab**	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		Total (Marks)
UAIPC304	**Data Structure using Python Lab**	Term work	Min pass	Marks	Min pass	**25**
		25	40%	-	-	

Course Description: The Data Structure using Python Lab provides practical exposure to the implementation of linear and non-linear data structures such as linked lists, stacks, queues, trees, and graphs using Python. The lab focuses on applying sorting and searching algorithms, as well as traversal techniques, to solve real-world problems. It enhances students' programming, analytical, and problem-solving skills through hands-on experiments aligned with the theoretical course**.**

Course Objectives:

- To help students understand the implementation and application of linear and non-linear data structures using Python, including linked lists, stacks, queues, trees, and graphs for problem solving.
- To impart basic knowledge of sorting, searching, and traversal techniques, and to enhance programming and analytical skills through data structure applications and real-world scenarios.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** I**mplement**[3] linear and non-linear data structures like linked list, stacks, queues, tree and graph
- **CO2** **Demonstrate** [3] different sorting and searching algorithm and traversal techniques on non-linear data structures.

List of Experiments:

All the experiments will be based on the course content of Data Structure using python with curse code UAIPC302

References :

1. John Canning, Alan Broder, Robert Lafore – Data Structures & Algorithms in Python – 2022 – 1st Edition
2. Shriram K. Vasudevan, Abhishek S. Nagarajan, Karthick Nanmaran – Data Structures Using Python – 2021 – 1st Edition

Evaluation Scheme:

1.TERM WORK assessment must be based on the overall performance of the student with every assignment and course work assigned by course coordinator graded from time to time.
2.The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UAIPC305	Introduction to AI and Generative AI LAB	-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)		Semester End Exam(SEE)		Total (Marks)
		Term work	Min. Passing %	Marks	Min pass	
UAIPC305	Introduction to AI and Generative AI - LAB	25	40%	-	-	25

Course Description: The Introduction to AI and Generative AI Lab provides practical exposure to fundamental AI concepts and their application in real-world problem solving. It enables students to implement and analyze AI techniques, including generative models such as VAE, GANs, and diffusion models, while developing analytical and problem-solving skills.

Course Objectives:

- To introduce the basic concepts of Artificial Intelligence and Generative Artificial Intelligence, and help students understand AI techniques.
- To impart basic knowledge of generative models and to promote understanding of their applications.

Course Outcomes: After the successful completion of the course students will able to:

CO1	**Apply**[3]basic principles of AI in solutions that require problem solving, inference, Perception, knowledge representation, and learning.
CO2	**Analyze**[4] generative AI applications using models such as VAE, GANs, and diffusion models for real-world use cases.

List of Experiments

All the experiments will be based on the course content of **Introduction to AI & Generative AI** with **course code UAIPC303**

References:

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson Education, 4th Edition, 2021.
2. Numa Dhamani, “Introduction to Generative AI”, Kindle Edition, 2024.

Evaluation Scheme:

1. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UAIPC306	Object Oriented Programming Using JAVA	-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		Total (Marks)
		Term work	Min pass	Marks	Min pass	
UAIPC306	Object Oriented Programming Using JAVA	25	40%	25	40%	**50**

Course Description: This course aims to develop problem-solving skills by introducing learners to object-oriented programming concepts such as classes, objects, inheritance, polymorphism, abstraction, and encapsulation using Java. It enables students to design, implement, and test modular, reusable, and secure software solutions aligned with industry practices. The course also promotes computational thinking, and application development skills to support multidisciplinary learning and innovation.

Prerequisites: C++ Programming

Course Objectives:

- To help students understand programming concepts and object-oriented principles using Java.
- To impart basic knowledge of Java features and to enhance skills in application development and problem solving.

Course Outcomes: After the successful completion of the course students will able to:

CO1:Apply[3] core Java programming and object-oriented concepts to develop modular and reusable applications.

CO2: Analyze[4] and implement Java-based application and mini-project.

List of Experiments

Four hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 18-20 experiments. Students of different batches should implement different programs based on the following guidelines:

- Basics of Java Programming
  1. Java Setup and Basic Input/Output
  2. Working with Arrays
  3. Strings and String Buffer/String Builder
  4. Control Flow Statements
  5. Defining Methods and Method Overloading
  6. Recursion in Java

- Object-Oriented Programming Concepts
  1. Creating Classes and Objects
  2. Constructor Overloading
  3. Inheritance (Single and Multilevel)
  4. Method Overriding and Dynamic Binding
  5. Abstraction Using Abstract Classes
  6. Interfaces and Multiple Inheritance

- Advanced OOP Features
  1. Packages and Access Modifiers
  2. Exception Handling
  3. Java Collections Framework

- File Handling and Advanced Concepts
  1. File Handling in Java
  2. Regular Expressions
  3. Multithreading Using Thread Class
  4. Multithreading Using Runnable Interface

- Application Development
  1. Java Mini Project (OOP-Based Application)

References:

1. Herbert Schildt, “Java: The Complete Reference”, McGraw-Hill, 13th Edition, 2024.
2. E. Balagurusamy, “Object-Oriented Programming with Java”, McGraw-Hill, 7th Edition, 2024.

Evaluation Scheme:

1. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UAIVE003	Advanced Web Designing Lab	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		Total (Marks)
UAIVE003	Advanced Web Designing Lab	Term work	Min pass	Marks	Min pass	50
		25	40%	2 5	40%	

Course Description: This course aims to develop practical skills in modern web application development by enabling students to design responsive user interfaces using HTML, CSS, and JavaScript, build dynamic applications with modern frameworks, and develop server-side solutions using Node.js with database integration. It also focuses on RESTful API development, basic web security practices, and deploying full-stack web applications using modern tools and cloud platforms.

Course Objectives:

- To introduce the fundamentals of modern web application development and to help students understand webpage design using HTML, CSS, and JavaScript.
- To impart basic knowledge of responsive web design, client-side scripting, and server-side web application development with database integration.

Course Outcomes: After the successful completion of the course students will able to:

CO1 **Apply**[3]HTML, CSS and PHP concepts for web application development.

CO2 **Examine**[4]JavaScript frameworks such as jQuery and Backbone**.**

List of Experiments

Four hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 18-20 experiments. Students of different batches should implement different programs based on the following guidelines

- **Front-End Web Development Fundamentals**

1. Design and create a structured webpage using HTML with semantic elements.
2. Apply CSS styling techniques to enhance webpage presentation.
3. Develop responsive layouts using Flexbox and Grid systems.
4. Implement basic client-side scripting using JavaScript.
5. Perform DOM manipulation to dynamically update webpage content.

- **Advanced Client-Side Development**

1. Implement event handling to manage user interactions.
2. Explore and compare modern front-end frameworks (e.g., React, Vue.js).
3. Develop a basic single-page application using a chosen front-end framework.
4. Utilize external libraries and packages for enhanced functionality (e.g., visualization, utilities).
5. Optimize front-end performance and ensure cross-browser compatibility.

- **Server-Side Development with Node.js**

1. Set up a basic Node.js server to handle HTTP requests and responses.
2. Implement routing to serve static and dynamic content.
3. Integrate middleware for request processing and logging.
4. Connect a database (e.g., MongoDB) to the server application.
5. Implement CRUD operations for database interaction.

- **API Development and Web Security**

1. Design and implement RESTful APIs for client-server communication.
2. Test APIs using tools like Postman or similar platforms.
3. Identify and mitigate common web security vulnerabilities.
4. Implement authentication and basic authorization mechanisms.

- **Deployment and Project Implementation**

1. Develop and deploy a full-stack web application (e.g., portfolio website) with cloud hosting and CI/CD integration

References:

1. Uttam K. Roy, “Web Technologies”, Oxford University Press, 1st Edition, 2010.

2. Steven Holzner, “The Complete Reference PHP”, Tata McGraw-Hill, 1st Edition, 2008

3. Robin Nixon, “Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5”, O’Reilly Publications, 4th Edition, 2015.

Evaluation Scheme

1.TERM WORK assessment shall be based on the overall performance of the student with every assignment graded from time to time.

2.The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.

3.Based on above scheme grading and TERM WORK assessment should be done

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UHSAE008	Employability Skills	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		01	02	-	-	02	-

Evaluation Scheme

Course Code	**Course Name**	**In Semester Evaluation**		**Semester End Exam (OE/POE)**		**Total (Marks)**
UHSAE008	Employability Skills	Term work	Min pass	Marks	Min pass	**50**
		25	40%	2 5	40%	

Course Description: Engineering education can no longer remain confined to technical knowledge alone. Employers today expect graduates not only to possess disciplinary understanding, but also to demonstrate communication ability, professional conduct, problem-solving capacity, workplace awareness, and increasing comfort with digital tools, including emerging AI-enabled environments. The shift is visible across sectors. Whether a student enters core engineering, IT services, analytics, manufacturing, design, operations, entrepreneurship, or higher studies, employability now depends as much on readiness for work as on academic performance.

A common employability course, therefore, is not an add-on. It is a foundational academic intervention. It prepares students to understand themselves better, communicate more effectively, work with others, think with structure, and present themselves with greater clarity in professional settings. It also introduces them to the changing nature of work, where digital literacy, responsible use of AI tools, and professional judgment are becoming increasingly relevant.

This course has been designed as a common offering for all engineering streams, keeping the content broad-based, practical, interactive, and immediately relevant. The intention is not to overload students with theory, but to build awareness, confidence, and observable workplace behaviors that will support both campus readiness and long-term career development.

Course Objectives:

1. To develop greater self-awareness regarding their strengths, interests, learning gaps, and professional aspirations.
2. To communicate more clearly and confidently in academic and workplace-like settings.
3. To demonstrate basic professional etiquette, ethical awareness, and responsible conduct.
4. To use structured thinking for solving simple problems and to participate effectively in team-based activities.

5. To develop placement-oriented aptitude, workplace expectations, and early career preparation.
6. To acquaint learners with the growing relevance of digital tools and AI in professional environments and use such tools with responsibility and awareness.

Course Outcomes: At the end of this course students will be able to:

- explain the importance of employability skills in engineering careers;
- present themselves with greater confidence through self-introduction, written communication, and presentations;
- participate constructively in group discussions and team tasks;
- demonstrate basic aptitude readiness and structured problem-solving ability;
- prepare a basic résumé and articulate career goals;
- Identify responsible and productive uses of digital and AI tools in learning and work contexts.

Syllabus

Module	Unit	Description	Hours
1.0		**Employability Awareness and Self-Discovery**	
1	1.1	Faculty input on employability, workplace expectations, and factors beyond academic performance (attitude, communication, reliability);	04
	1.2	Structured self-assessment covering strengths, habits, communication gaps, and first impressions (worksheet-based activity);	
	1.3	Guided reflection with peer discussion to identify improvement areas and employability gaps;	
	1.4	Classroom discussion and preparation of a personal employability improvement statement (visible output);	
2.0		**Professional Mindset, Ethics and Responsibility**	
2	2.1	Input on professional mindset: discipline, accountability, ownership, and workplace ethics;	04
	2.2	Case-based learning on missed deadlines, communication gaps, responsibility, and ethical dilemmas;	
	2.3	Role-play exercises demonstrating weak vs effective workplace behavior;	
	2.4	Reflection on “habits that influence professional credibility” and behavior	

		correction strategies;	
3.0		**Analytical Thinking, Aptitude, Emotional Intelligence and Decision-Making**	
3	3.1	Introduction to critical thinking: distinguishing facts vs assumptions, identifying problems, and structured analysis;	04
	3.2	Emotional intelligence in academic and workplace settings: empathy, listening, self-control, and response vs reaction;	
	3.3	Aptitude orientation: speed, accuracy, logical reasoning, and problem-solving strategies;	
	3.4	Practice exercises on reasoning and aptitude with focus on error analysis and improvement;	
	3.5	Decision-making in workplace scenarios: prioritization, judgment, and handling ambiguity;	
	3.6	Case-based activities integrating thinking, behavior, and decision-making with group discussion and reflection;	
4.0		**Career Readiness, Resume Building and Digital Presence**	
4	4.1	Overview of career pathways: core engineering, IT, analytics, higher studies, and entrepreneurship; preparation of personal career planning sheet;	06
	4.2	Résumé structure workshop covering sections, clarity, and recruiter expectations;	
	4.3	Resume Writing I: **ATS-friendly resume**, keyword optimization, achievement-based content, and project representation;	
	4.4	Resume Writing II: **visual resume, video resume, and e-profile development** (formats, scripting, and presentation guidelines);	
5.0		**Group Discussion and Team Interaction**	
5	5.1	Introduction to group discussion norms and evaluation parameters (content, clarity, listening, structure, and collaboration);	06
	5.2	Practice group discussions on familiar topics (e.g., technology in student life, workplace expectations);	

	5.3	Techniques for entering, sustaining, and concluding discussions; building arguments and responding to others;	
	5.4	Observation of speaking balance, idea contribution, listening behavior, and team dynamics during GD;	
	5.5	Team reflection on communication effectiveness and group behavior; identification of strengths and improvement areas;	
	5.6	Feedback using GD rubric and preparation of individual improvement note;	
6.0		**Presentation Skills and Personal Interview**	06
	6.1	Introduction to interview process: types of interviews, employer expectations, and common mistakes;	
	6.2	Body language, confidence, listening skills, and professional conduct during interviews;	
	6.3	Common HR questions, behavioral responses, and answer structuring techniques (clarity, relevance, examples);	
	6.4	Mock interviews (triad format) with peer and teacher evaluation using structured rubric;	
	6.5	Presentation skills: structure (opening, flow, conclusion), clarity, voice modulation, and audience engagement;	
	6.6	Student presentations (2–3 minutes) followed by peer observation, faculty feedback, and improvement note;	
7.0		**Digital Readiness, AI Awareness and Responsible Use (Only for Non-CSE and AIML Branches)**	
	7.1	Faculty-led discussion on AI in workplace tasks and engineering contexts	02
	7.2	Examples of AI-assisted writing, coding, research support	

Internal Assessment (T1, T2 and FET)

1. T1 or T2 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

Note: This course will be taught in PBL mode; hence, the focus will be on teaching applications of content in the syllabus in real life situations.

Programme Structure for B. Tech.: Second Year Artificial Intelligence and Machine Learning (R25)

IV: Teaching Scheme

Course code	Course Name	Teaching scheme(Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
UAIPC401	Software Engineering	3	-	1	3	-	1	4
UAIPC402	Operating System	3	-	-	3	-	-	3
UAIPC403	Database Management System	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
UXXOE02X	Open Elective II	2	-	-	2	-	-	2
UCMEM003	Project Management	2	-	-	2	-	-	2
UAIVS401	Software Proficiency I (R Programming)	-	4	-	-	2	-	2
UAIPC405	Database Management System Lab	-	2	-	-	1	-	1
UAIFP401	Field Project	-	4	-	-	2	-	2
USTVE005	AI Ethics & Governance	1	-	1	1	-	1	2
Total		**16**	**10**	**2**	**16**	**5**	**2**	**23**

Semester IV: Evaluation Scheme

Course code	Course Name	Theory Marks					Practical Marks		Total
		Internal Assessment				SEE	Term work	Practical Oral/POE	
		T1	T2	FET	Total				
UAIPC401	Software Engineering	10	10	5	25	50	25	-	100
UAIPC402	Operating System	10	10	5	25	50	-	-	75
UAIPC403	Database Management System	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor II	-	10	5	15	35	-	-	50
UXXOE02X	Open Elective II	-	10	5	15	35	-	-	50
UCMEM003	Project Management	-	10	5	15	35	-	-	50
UAIVS401	Software Proficiency I (R Programming)	-	-	-	-	-	25	25	50
UAIPC405	Database Management System Lab	-	-	-	-	-	25	-	25
UAIFP401	Field Project	-	-	-	-	-	25	25	50
USTVE005	AI Ethics & Governance	10	10	5	25	-	25	-	50
								Total	**575**

Exit Option:

- Students will have the flexibility to enter a programme in odd semesters and exit a programme after the successful completion of even semesters as per their future career needs.
- Students exiting the Second Year Programme after securing minimum 80 credits will be awarded UG Diploma in the relevant Discipline/Subject provided they secure additional 8 credits in skill-based vocational courses(skill-based courses, internship, mini projects etc) offered during summer vacation after the second year.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UAIPC401	Software Engineering	3	-	1	3	-	1

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme(In Semester)					Semester End Exam(SEE)			Total (Marks)
		T1	T2	FET	Total	Min pass	Term work	Marks	Min pass	
UAIPC401	Software Engineering	**10**	**10**	**5**	**25**	**40%**	**25**	**50**	**40%**	**100**

Course Description: This Software Engineering course is designed to equip students with a comprehensive understanding of fundamental principles, techniques, and best practices required for successful software development. It emphasizes a structured and systematic approach to managing complex software projects by covering key areas such as project estimation, requirements analysis, system design, testing methodologies, and quality management. By integrating theoretical knowledge with practical programming skills, the course enables students to develop high-quality, reliable software solutions that meet industry standards and client requirements, preparing them to contribute effectively to professional software development environments

Course Objectives:

- To introduce the principles of Software Engineering.
- To impart basic knowledge of methods for capturing, specifying, visualizing, and analyzing software requirements.
- To help students understand design and testing principles in software project development.
- To promote understanding of project management through the software development life cycle.

Course Outcomes: After the successful completion of the course students will able to:

CO1	**Explain**[2]fundamental concepts, characteristics, principles, and practices of software engineering.
CO2	**Apply**[3]prescriptive and agile process models to select suitable software development approaches
CO3	**Analyze**[4]software requirements, SRS, UML, DFDs, and design models for structured system development.
CO4	**Apply**[3] different Risks and Configuration Management for software product
CO5	**Identify**[3] defects to improve the overall quality of the software using automated testing

Course Contents

Module	Unit	Description	Hours
1.0		**Basics of Software Engineering**	
	1.1	Software engineering, characteristics of software, types of software	9
	1.2	Software engineering core principles, communication practices, planning practices, modelling practices, construction practices, software deployment practices **Case Study:** Online Shopping System Development, applying core software engineering principles for efficient design, development, and deployment.	
2.0		**Software Engineering Process Models**	
	2.1	Prescriptive process models: Waterfall model, Incremental model, RAD model, prototyping model, Spiral model, V model	9
	2.2	Agile software development: Agile process, and its importance, extreme programming, Scrum, scrum Selection criteria for software process model, Kanban. **Case Study:** E-commerce Website Development using Agile Scrum, selecting suitable process models for iterative and flexible development.	
3.0		**Software Requirement, Modeling and Design**	
	3.1	Requirement engineering: Requirement engineering task, types of requirements, developing use-case	9
	3.2	Need of SRS, format and it's characteristics, Data flow diagram (DFD), structured, flowcharts, API Design Basics, UML **Case Study:** Library Management System Design, capturing requirements and creating UML, DFDs, and SRS for system development.	
4.0		**Risks and Configuration Management**	
	4.1	Risk Management: Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan	9
	4.2	Software Configuration Management: Software Configuration Management, The SCM Repository, SCM Process, Configuration Management for any suitable software system Suggested Free Open Source Tools: CF Engine Configuration Tool, Puppet Configuration Tool. **Case Study:** Banking Software Risk Management, identifying risks and using configuration tools to maintain system stability and version control.	
5.0		**Software Testing and Defect Management**	
	5.1	Software testing, objective of testing, software testing life cycle (STLC), The box approaches: Compare white box testing, black box testing, grey testing, types of testing	9
	5.2	Test planning: Preparing a test plan, Test cases, Test management: Test infrastructure management, Comparison of manual testing and automation testing **Case Study:** Mobile Application Testing, ensuring quality through test planning, execution, and defect tracking using manual and automated testing.	

Text Books:

1.Roger Roger S. Pressman and Bruce R. Maxim, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 9th Edition, 2020.
2. Ian Somerville, "Software Engineering", Pearson Education, 10th Edition, 2015.

References:

1. David Farley, "Modern Software Engineering: Doing What Works to Build Better Software Faster", Addison-Wesley, 1st Edition, 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. Fifth module will be assessed for 5 marks separately.

Semester End Exam

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 question (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
UAIPC402	Operating System	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme(In Semester)					Semester End Exam(ESE)		Total (Marks)
		T1	T2	FET	Total	Min pass	Marks	Min pass	
UAIPC402	Operating System	10	10	5	25	40%	50	40%	75

Course Description: The Operating Systems course introduces to the fundamental concepts, principles, and Functionalities of operating systems. Operating systems serve as the bridge between computer hardware and software applications, managing system resources and providing a user-friendly interface for efficient interaction with the computer. This course covers topics such as process management, memory management, file systems, device management, and security.

Course Objectives

- To help students understand the basic concepts of operating systems and their services.
- To impart basic knowledge of processes, process states, Process Control Blocks (PCB), and CPU scheduling algorithms.
- To enable students to understand process synchronization, deadlock, detection, and prevention methods.
- To promote understanding of memory management strategies and file system design.

Course Outcomes: After the successful completion of the course students will able to:

CO1 **Explain**[2]basic concepts of operating system and their structures.

CO2 **Analyze**[4]process management and scheduling techniques.

CO3 **Apply**[3]synchronization mechanisms and deadlock handling methods**.**

CO4 **Analyze**[4]memory management and virtual memory techniques**.**

CO5 **Analyze**[4]file systems and I/O subsystem operations**.**

Course Contents

Module	Unit	Description	Hours
1.0		**Overview**	
	1.1	Introduction to Operating Systems, Operating System structure, Types of Operating Systems, Operating System Services, and Views of Operating System.	9
	1.2	System calls, Types of system Calls, System programs, Kernel and its types. Comparison of various OS. (Windows, Linux, Mac and Android) Introduction to Linux OS: Components of Linux system, Basic Shell commands **Case studies:** 1) Automating User and File Management in Linux using Shell Script 2) Demonstrating Different OS Structures (Monolithic, Layered, Microkernel OS)	
2.0		**Process Management**	
	2.1	Process concept: Basic concepts, Process States, Process Control Block, Context switch, Operations on processes, Inter-process communication, Threads.	9
	2.2	Process Scheduling: Scheduling criteria, Types of schedulers, scheduling algorithms, Multiple-Processor scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling. **Cast Studies -** 1) Prepare case study on challenges for Real-time Scheduling 2)Performance Comparison of Scheduling Algorithms (FCFS vs SJF, RR vs Priority Scheduling) under different workload conditions	
3.0		**Process Synchronization and Deadlock**	
	3.1	Background, the critical section problem, Peterson's solution, synchronization hardware, Semaphores, classic problems of Synchronization, Monitor	9
	3.2	System model, deadlock characterization, methods for handling deadlocks. Deadlock preventions, deadlock avoidance, deadlock detection, deadlock recovery **Case Studies -** 1) Interprocess Communication (IPC) in a Banking System 2) Deadlock in Railway Scheduling Systems	
4.0		**Memory Management**	
	4.1	Memory Management Strategies: Background, swapping, contiguous memory allocation, paging, structure of the page table, Segmentation	9
	4.2	Virtual Memory Management: Background, demand paging, copy-on-write, page replacement, Thrashing. **Case study :** Intel Pentium	
5.0		**File Management& I/O Subsystem**	
	5.1	File System: File concept, access methods, Directory and disk structure, Disk Scheduling, file-system mounting, file sharing, and protection.	9
	5.2	Overview of I/O system, I/O hardware, Application I/O interface, Kernel I/O subsystem **Case Studies -** 1) Study of Linux File System. 2) Study of Android File System.	

Text Books

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, "Operating System Concepts", John Wiley & Sons, 10th Edition, 2018
2. William Stallings, "Operating Systems: Internals and Design Principles", Pearson Education, 9th Edition, 2018.

References

1. David Farley, "Modern Software Engineering: Doing What Works to Build Better Software Faster", Addison-Wesley Professional, 1st Edition, 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. Fifth module will be assessed for 5 marks separately.

Semester End Exam

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 question (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
UAIPC403	Database Management System	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam(SEE)		Total (Marks)
		T1	T2	FET	Total	Min. Passing %	Marks	Min pass	
UAIPC403	Database Management System	10	10	5	25	40%	50	40%	75

Course Description:

This course aims to develop a strong foundation in database system design and implementation. Students explore conceptual and logical database modeling techniques, including E-R diagrams and relational design. The course covers SQL, database security, transaction management, and concurrency control mechanisms. By the end of the course, students will be able to analyze requirements, design database schemas, and implement efficient database systems.

Course Objectives:

- To help students understand fundamental concepts of database systems, architecture, data models, and database environments.
- To impart basic knowledge of relational model concepts, SQL commands, constraints, and file organization techniques for data storage and retrieval.
- To enable students to understand conceptual database design using ER diagrams, joins, queries, and aggregation functions.
- To promote understanding of database design theory including functional dependencies, normalization forms, and triggers.

Course Outcomes: After the successful completion of the course students will able to:

CO1 **Explain**[2]database system concepts, data models, and ER/EER design techniques for structured database development

CO2 **Apply**[3]SQL queries and data storage techniques including indexing and hashing for efficient data retrieval and management.

CO3 **Analyze**[4]relational database design principles and normalization techniques to develop well-structured database schemas

CO4 **Analyze**[4]transaction management concepts and concurrency control protocols to ensure database consistency and reliability.

CO5 **Explain**[2]NoSQL database concepts and apply MongoDB operations for modern data storage and processing needs.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction to Databases:**	
1	1.1	Introduction to Database Management Systems, Purpose of Database Systems, Database-System Applications, View of Data, Database Languages, Database System Structure, Enterprise Constraints Data Models.	9
	1.2	Database Design and ER Model: Entity, Attributes, Relationships, Constraints, Keys, Design Process, Entity Relationship Model, ER Diagram, Design Issues, Extended E-R Features, Converting E-R & EER diagram into tables. **Case Study:** Study of Architecture of any DBMS like Oracle or MySQL. Design a database schema for any problem	
2.0		**SQL& Data storage:**	
2	2.1	SQL: DDL, DML, Select Queries, String, Date and Numerical Functions, Aggregate Functions ,View, Indexes, Group by and Having Clause, Join Queries, Set, Set operation, Set membership, Nested queries, DCL, TCL	9
	2.2	File Organization- Organization of records in File, Basic Concepts indexing, Static Hashing- Dynamic Hashing **Case Study:** Design and implement a Student Course Management System using SQL to manage students, courses, and faculty members efficiently. The system should store and retrieve relevant data, ensuring integrity, security, and performance optimization.	
3.0		**Relational Database**	
3	3.1	Relational Model: Basic concepts, Attributes and Domains, Codd's Rules, Relational Integrity, Referential Integrities,	9
	3.2	Database Design: Features of Good Relational Designs, Normalization, Atomic Domains and First Normal Form, Decomposition using Functional Dependencies, 2NF, 3NF, BCNF ,5NF **Case study:** Design and Optimization of a Relational Database for a University Management System	
4.0		**Transaction Management and Concurrency Control**	
4	4.1	Basic concept of a Transaction, Transaction Management, Properties of Transactions, ACID, Concept of Schedule, Serial Schedule, Serializability: Conflict and View, Cascaded Aborts, Recoverable and Non-recoverable Schedules	9
	4.2	Concurrency Control: Need, lock-based protocols, time-stamp based protocols, validation-based protocols **Case study:** Design Online Shopping Cart Transaction Management In an e-commerce platform, multiple users simultaneously add, update, and	

		purchase products. To ensure data consistency and reliability, the system must handle concurrent transactions effectively.	
5.0		**NoSQL Database**	
5	5.1	Introduction to NoSQL Database, NoSQL data models, CAP theorem and BASE Properties, Comparative study of SQL and NoSQL	9
	5.2	MongoDB: CRUD Operations, Indexing and Aggregation **Case study:** Study NoSQL Database Selection for a Social Media Platform.	

Text Books:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, 7th Edition, 2019.
2. Thomas Connolly and Carolyn Begg, "Database Systems: A Practical Approach to Design, Implementation, and Management", Pearson Education, 6th Edition, 2014.

References:

1. Adam Fowler, "NoSQL For Dummies", John Wiley & Sons, 1st Edition, 2025.

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 Exam:

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

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UCMEM003	**Project Management**	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		02	**-**	**-**	**02**	**-**	**-**

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme(In Semester)					Semester End Exam(SEE)		Total (Marks)
UCMEM003	**Project Management**	T1	T2	FET	Total	Min pass	Marks	Min pass	**50**
		-	**10**	**5**	**15**	**40%**	**35**	**40%**	

Course Description: This course presents the principles and techniques of managing engineering projects from the initiation phase, through planning, execution, control and closeout. Students will develop the analytical skills and awareness necessary for the management side of engineering projects.

Course Objectives

- To help students understand the principles of management and their application in engineering contexts.
- To impart basic knowledge of organizational structures, leadership styles, and decision-making processes.
- To enable students to understand project management techniques used in engineering problems.
- To promote understanding of financial and economic aspects of engineering projects.

Course Outcomes: After the successful completion of the course students will able to:

CO1	**Explain**[2] the fundamentals of management and organizational behavior.
CO2	**Apply**[3] project planning and scheduling techniques in engineering projects
CO3	**Analyze**[4] financial statements and cost estimation for engineering decisions.
CO4	**Evaluate**[5] leadership, motivation, and communication strategies in Organizations.
CO5	**Analyze**[4]strategic plans for innovation and technology management

Course Contents

Module	Unit	Description	Hours
1.0		**Principles of Management and Organizational Behavior**	**6**
	1.1	Evolution of management thought. Functions of management: Planning, Organizing, Staffing, Directing, Controlling. Organizational structures and design. Motivation theories, leadership styles, communication in organizations	
2.0		**Project Management**	**6**
	2.1	Project life cycle, Project planning and scheduling (PERT, CPM, Gantt Charts).Resource allocation and leveling .Risk management in engineering projects.	
3.0		**Financial and Cost Management**	**6**
	3.1	Basics of engineering economics. Cost estimation and budgeting. Breakeven analysis .Financial statements and ratio analysis.Capital budgeting techniques.	
4.0		**Decision-Making and Quality Management**	**6**
	.1	Decision-making models and tools. Total Quality Management (TQM) and Six Sigma. Productivity measurement and improvement. ISO standards and quality audits.	
5.0		**Strategic Management and Technology Innovation**	**6**
	5.1	Strategic planning and competitive advantage .Technology management andinnovationstrategies.Changemanagementinengineeringorganizations .Case studies in engineering management	

Textbook

1. Pravin Kumar, “Engineering Economy and Management”, Wiley India, 1st Edition, 2019.
2. K. Anil Kumar, “Engineering Management”, 2nd Edition, 2021

References

1. Harold Kerzner, “Project Management: A Systems Approach to Planning, Scheduling, and Controlling”, 13th Edition, 2022.

Internal Assessment (T1, T2 and FET)

1.T2 should be based on Third to Fourth modules, for 10 marks.
2.FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 07 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (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
UAIVS401	**Software Proficiency Programming I (R Programming)**	-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		Total (Marks)
		Term work	Min pass	Marks	Min pass	
UAIVS401	**Software Proficiency I (R Programming)**	25	40%	25	40%	**50**

Course Description: This laboratory course provides hands-on experience in R programming for data analysis and statistical computing. Students learn to perform data handling, preprocessing, visualization, and statistical analysis using R, along with debugging and working with packages and libraries. The course also emphasizes applying R for real-world tasks such as exploratory data analysis and basic web application development.

Prerequisite: - Data Structures using python, Database Management System.

Course Objectives:

- To introduce the fundamentals of R programming for data analysis and statistical computing.
- To help students understand data handling, preprocessing, visualization, and statistical analysis using R

Course Outcomes: After the end of this course students will able to

CO1: Apply[3] R programming concepts to perform data handling, statistical analysis, visualization, debugging, and package-based development tasks.

CO2: Analyze[4] datasets using regression, hypothesis testing, exploratory data analysis, and web application development using R tools and libraries

Experiment List

- **Basics of R Programming**

1. Usage of R – declaring variables, expressions, control structures, functions, and executing R scripts
2. Working with datasets and R tables – create, read, and write operations
3. Demonstration of arrays and matrices in R
4. Managing data using data frames
5. Operations on Date and Time in R

- **Data Manipulation and Preprocessing**

1. Manipulating and processing data – merging datasets and reshaping data
2. Data sorting, filtering, and merging techniques
3. Data cleaning and handling missing values
4. Text processing in R
5. Accessing and working with R packages and libraries

- **Statistical Analysis in R**

1. Descriptive statistical analysis
2. Correlation and covariance analysis
3. Performing parametric t-test in R
4. Performing linear and multiple regression in R

- **Visualization and Debugging**

1. Data visualization techniques in R
2. Working with advanced visualization libraries
3. Demonstration of debugging tools in R (Swirl)
4. Debugging R programs using Swirl (practical implementation)

- **Applications and Case Study**

1. Developing and building a web application in R
2. Case Study: Exploratory Data Analysis (EDA) using R

References:

1. Sandeep Rakshit, “R for Beginners”, McGraw-Hill Publications, 1st Edition, 2017.
2. Roger D. Peng, “R Programming for Data Science”, Leanpub Publications, 2022. .

Evaluation Scheme:

1. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UAIPC404	**Database Management System Laboratory**	**-**	**02**	**-**	**-**	**01**	**-**

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		Total (Marks)
		Term work	Min pass	Marks	Min pass	
UAIPC404	**Database Management System Laboratory**	**25**	**40%**	**-**	**-**	**25**

Course Description: The Database Management System Laboratory provides hands-on experience in designing and implementing database systems using SQL and modern database tools. The lab focuses on applying database design concepts, executing queries for data manipulation, and managing transactions. Students also explore advanced SQL operations and NoSQL concepts such as MongoDB CRUD techniques for efficient data handling. Through practical experiments aligned with the theory course, the lab enhances skills in database design, analysis, and real-world application development.

Course Objectives:

- To introduce database design concepts and their implementation using SQL.
- To help students understand SQL queries for data manipulation and transaction management.

Course Outcomes: After the successful completion of the course students will able to:

CO1 **Apply**[3]database design concepts and execute SQL queries for data manipulation and transaction management

CO2 **Analyze**[4]and implement advanced SQL operations and MongoDB CRUD techniques for efficient data handling.

List of Experiments

All the experiments will be based on the course content of **Database Management System** with course code **UAIPC403**

References

1. Thomas Connolly and Carolyn Begg, “Database Systems: A Practical Approach to Design, Implementation and Management”, Pearson Education, 3rd Edition, 2003.
2. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database System Concepts”, McGraw-Hill Education, 6th Edition, 2011.

Evaluation Scheme:

1. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.

2. The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UAIFP401	**Field Project**	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		-	04	-	-	02	-

Evaluation Scheme

Course Description:

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		Total (Marks)
UAIFP401	**Field Project**	Term work	Min pass	Marks	Min pass	**50**
		25	40%	25	40%	

This course focuses on building a strong connection between the institution and the local community through active student engagement. It provides students with opportunities to interact with stakeholders, understand real-world challenges, and develop empathy, teamwork, and problem-solving abilities. Emphasis is placed on identifying community needs and applying engineering knowledge to design and implement practical, sustainable solutions. The course also encourages critical evaluation of the outcomes of these interventions to ensure meaningful and long-term impact on society.

Course Objectives

- To establish a mutually beneficial relationship between the institution and the local community.
- To provide opportunities for students to engage with the community, fostering empathy, teamwork, and problem-solving skills.

Course Outcomes: After the successful completion of the course students will able to:

CO1: **Analyze**[4]local community needs and challenges by engaging with stake-holders and evaluating real-world problems.

CO2: **Design**[5] and Implement practical, creative, and context-specific solutions using engineering principles to address community issues

Course Contents

Implementation

- A group of 3 to 4 students could be assigned for a particular habitation or village or municipal ward, as far as possible, in the near vicinity of their place of stay/college premise.
- Each group /practical batch is allotted to a faculty member of the department as a mentor.
- A division of 60 students can have 3 batches of minimum 20 students. Practical load of 4 hours to be allocated to each batch.
- The group of students will be associated with a government official / village authorities /NGOsetc. concerned, allotted by the district administration, during the duration of the

project.

- The Community Engagement Project should be different from the regular programmes of NSS/NCC/Green Club/Hobby Clubs, Special Interests Groups etc
- An activity book has to be maintained by each of the students to record the activities undertaken/involved and will be countersigned by the concerned mentor/HoD.
- Project report shall be submitted by each student/group of students.
- An internal evaluation shall also be conducted by a committee constituted by the HoD. Evaluation to be done based on the active participation of the student and marks could be awardedby the mentor/HoD.
- Students groups can conduct an awareness programme on Health and Hygiene or in Organic Farming or in Fisheries or in advocating prohibition of liquor or about renewable energy, e-waste management or any other activity in an area of their studies and as per his/her aptitude.
- Oral Examination shall consist of presentation and demonstration of the project work carriedout by the project groups.

Suggestive list of topics under Community Engagement Project:

The below lists are not exhaustive and open for HoD's or mentors to add, delete or Modify. It is expected that the focus should be on specific local issues in their nearby areas. The students are expected to carry out these projects with involvement, commitment, responsibility and accountability.

The mentors of a student/group of students shall

• Use/ miss-use of cell phones
• Career orientation of youth
• Water facilities and drinking water availability
• Health and hygiene of the school going students, home makers and old personals
• Health intervention and awareness programmes
• Horticulture
• Herbal and Nutrition
• Traditional and Modern health care methods
• Food habits
• Air /Sound /Water pollution
• Plantation and Soil protection
• Renewable energy and Solar Systems
• Yoga awareness and practice
• Health care awareness programmes and their impact
• Organic farming
• Food adulteration
• Incidence of Diabetes and other chronic diseases
• Blood groups and blood levels

• Chemicals in daily life
• Music and dance
• Women education and empowerment

AI based topics:

- AI for early detection of Diseases
- AI-based health monitoring systems
- AI for water quality analysis
- AI-based drinking water prediction systems
- AI in crop yield prediction
- AI for plant disease detection
- AI for air pollution monitoring
- AI for waste management and segregation
- AI-based career guidance for youth
- AI in personalized education systems
- AI to analyze mobile phone usage patterns
- AI chatbots for health awareness
- AI for women education and empowerment
- AI for food adulteration detection
- AI-based nutrition recommendation systems
- AI for solar energy optimization
- AI for energy consumption prediction
- AI for accident prediction and road safety
- AI for smart surveillance systems
- AI for mental health and stress detection

Project Scope:

- Conduct workshops or awareness drives on topics like digital literacy, environmental sustain-ability, mental health, or career planning for local stakeholders Develop a simple prototype or solution that addresses a real-world problem (e.g., a water-saving device, simple mobile apps, or tools for community use).
- Organize clean-up drives, tree plantations, recycling campaigns, or energy conservation initiatives.
- Promote health through awareness programs on hygiene, nutrition, and exercise.
- Teach basic computer or technical skills to students, staff, or the community

Proposal Submission:

Group should Submit a two-page project proposal, preferably prior to the term commencement outlining the following:-

• Title of the project
• Aim, Objective and expected outcome
• Plan of execution (timeline and activities).

• Place of the CEP and involvement of any local authority, NGP
• Required resources (if any).
• Get approval from the designated faculty mentor.

Reference :

1. Alan S. Waterman, "Service-Learning: A Guide to Planning, Implementing, and Assessing Student Projects", Routledge, Latest Reprint Edition, 2018.
2. Surendra Reddy Bandi, Ashok Shettar, Gopalkrishna Joshi, *"Service-Learning in Engineering Education: Building Empathy, Design Thinking and Community Engagement"*, Latest Edition, 2025.

.

Evaluation Scheme

1. TERM WORK assessment must be based on the overall performance of the student with every assignment done on computer graded from time to time.
2. The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UHSVE004	AI Ethics and Governance	01	-	01	02	--	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Term Work		Total (Marks)
		T1	T2	FET	Total	Min pass	Marks	Min pass	
UHSVE004	AI Ethics and Governance	10	10	5	25	40%	25	40%	50

Course Description

This course introduces students to the ethical, legal, and societal implications of Artificial Intelligence. It focuses on responsible AI development, fairness, accountability, transparency, and governance frameworks. Students will explore real-world challenges, regulatory policies, and ethical decision-making in AI systems.

Course Objectives:

- To introduce fundamental concepts of ethics, morality, and their relevance in Artificial Intelligence systems.
- To help students understand issues related to bias, fairness, transparency, privacy, data governance, and security challenges in AI-driven systems.
- To impart basic knowledge of AI governance frameworks, policies, and regulatory requirements at national and global levels.
- To promote understanding of societal and legal implications of Artificial Intelligence and responsible and ethical AI practices.

Course Outcomes: After the successful completion of the course students will able to:

CO1	**Explain**[2]fundamental concepts of AI ethics, including morality, responsibility, and ethical principles in AI systems
CO2	**Analyze**[3]bias, fairness, and transparency issues in AI models and propose appropriate mitigation techniques
CO3	**Apply**[3]concepts of data privacy, security, and governance to ensure responsible use of AI systems
CO4	**Evaluate**[5]AI governance frameworks, policies, and regulations for ethical AI deployment
CO5	**Analyze**[4] societal and legal impacts of AI applications

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction to AI Ethics**	**6**
	1.1	Ethics vs. morality; importance of AI ethics; human-centered AI; responsibility in technology; key ethical concerns in AI systems; overview of ethical challenges in AI applications	
2.0		**Fairness, Bias and Transparency**	**6**
	2.1	Algorithmic bias; sources of bias in data and models; fairness definitions; bias detection techniques; bias mitigation methods (pre-processing, in-processing, post-processing); explainable AI (XAI); interpretability and transparency in AI systems	
3.0		**Privacy, Security and Data Governance**	**6**
	3.1	Data protection principles; informed consent; surveillance concerns; data minimization; differential privacy; data governance frameworks; security issues in AI; responsible data usage	
4.0		**AI Governance and Regulation**	**6**
	4.1	National and international AI governance frameworks; standards and policy principles; AI regulations; responsible AI deployment; accountability, auditability, and institutional oversight	
5.0		**Societal and Legal Implications of AI**	**6**
	5.1	AI applications in healthcare, education, employment, policing, and public systems; ethics washing; AI safety and robustness; public trust; societal impact; legal responsibilities and compliance	

Text Books

1. Mark Coeckelbergh, “AI Ethics”, MIT Press, 2024.
2. Brent Mittelstadt and Luciano Floridi, “The Ethics of Artificial Intelligence”, Oxford University Press, 2023.

References

1. John Mehan, “Artificial Intelligence: Ethical, Social, and Security Impacts for the Present and the Future”, IT Governance Publishing, 2nd Edition, 2024.

Internal Assessment (T1, T2 and FET)

3. T1 or T2 should be based on First to Fourth modules, for 10 marks.
4. FET shall be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
2. The duration of Semester End Examination shall be Two hours.

3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.