Sanjay Ghodawat University Kolhapur



School of Engineering and Technology

Department of Computer Science and Engineering

B. Tech. Computer Science and Engineering

Curriculum Structure and Syllabus (R25)

Semester: III-IV

Programme Structure for B Tech.: Second Year Computer Science and Engineering (R25)

Semester III: Teaching Scheme

Course code	Course name	Teaching scheme (Hrs/week)			Credits assigned			Total Credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
UCSBS301	Discrete Structures and Graph Theory	3	-	-	3	-	-	3
UCSPC302	Computer Organization and Architecture	3	-	-	3	-	-	3
UCSPC303	Data Structures	2	-	-	2	-	-	2
UCSPC304	Information Theory and Coding	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor I	2	-	-	2	-	-	2
UXXOE01X	Open Elective I	2	-	-	2	-	-	2
UCMEM001	Engineering Economics	2	-	-	2	-	-	2
UCSPC305	Computer Organization and Architecture Lab	-	2	-	-	1	-	1
UCSPC306	Data structures Lab	-	4	-	-	2	-	2
UCSFP301	Field Project	-	4	-	-	2	-	2
Total		**17**	**10**	**-**	**17**	**5**	**-**	**22**

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				
UCSBS301	Discrete Structures and Graph Theory	10	10	5	25	50	-	-	75
UCSPC302	Computer Organization and Architecture	10	10	5	25	50	-	-	75
UCSPC303	Data Structures	-	10	5	15	35	-	-	50
UCSPC304	Information Theory and Coding	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
UCSPC305	Computer Organization and Architecture Lab	-	-	-	-	-	25	-	25
UCSPC306	Data structures Lab	-	-	-	-	-	25	25	50
UCSFP301	Field Project	-	-	-	-	-	25	25	50
								Total	**550**

Programme Structure for B Tech.: Second Year Computer Science and Engineering (R25)

Semester IV: Teaching Scheme

Course code	Course Name	Teaching scheme (Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
UCSPC401	Operating System	3	-	-	3	-	-	3
UCSPC402	Database Management	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
UXXOE02X	Open Elective II	2	-	-	2	-	-	2
UCMEM003	Project Management	2	-	-	2	-	-	2
UCSVS401	Software Proficiency I (Java Programming)	-	4	-	-	2	-	2
UCSPC403	Database Management System Laboratory	-	4	-	-	2	-	2
UCSVE003	Open-Source Technologies		2	-		1	-	1
UHSAE008	Employability Skills	1	2	-	1	1	-	2
USTSVE005	AI Ethics and Governance	1	-	1	1	-	1	2
Total		**14**	**12**	**1**	**14**	**6**	**1**	**21**

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				
UCSPC401	Operating System	10	10	5	25	50	-	-	75
UCSPC402	Database Management	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor II	-	10	5	25	35	-	-	50
UXXOE02X	Open Elective II	-	10	5	15	35	-	-	50
UCMEM003	Project Management	-	10	5	15	35	-	-	50
UCSVS401	Software Proficiency I (Java	-	-	-	-	-	25	25	50
UCSPC403	Database Management System	-	-	-	-	-	25	25	50
UCSVE003	Open-Source Technologies	-	-	-	-	-	25	-	25
UHSAE008	Employability Skills	10	10	5	25	-	25	-	50
USTSVE005	AI Ethics and Governance	10	10	5	25	-	25	-	50
								Total	**525**

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 80credits 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		
UCSPC301	Discrete Structures and Graph Theory	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
UCSPC301	Discrete Structures and Graph Theory	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		10	10	5	25	40%	50	40%	75

Course Description

This course introduces fundamental concepts of discrete mathematics essential for computer science and engineering. It covers logic, set theory, relations, functions, Combinatorics, and graph theory. The course emphasizes mathematical reasoning and problem-solving techniques used in computing and algorithm design.

Course Objectives

- To develop logical thinking and mathematical reasoning skills.
- To introduce the fundamental concepts sets, relations, and functions in discrete systems.
- To familiarize students to apply combinatorial techniques in problem solving.
- To help students to understand graph structures and their applications in computing

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

- **CO1** **Explain**[2] concepts of mathematical logic and set theory.
- **CO2** **Apply**[3] relations and functions in problem solving
- **CO3** **Solve**[3] combinatorial problems using counting techniques.
- **CO4** **Apply**[3] graph theory concepts and analyze graph structures.
- **CO5** **Apply**[3] graph algorithms and representations to solve problems.

Course Contents

Module	Unit	Description	Hours
1.0		**Mathematical Logic**	**9**
	1.1	Propositions, logical connectives, truth tables, tautologies, contradictions	
	1.2	Logical equivalence, implication, predicate logic, quantifiers, Normal forms (CNF, DNF), inference theory, rules of inference.	
2.0		**Set Theory, Relations and Functions**	**9**
	2.1	Sets, operations on sets, Venn diagrams, laws of sets.	
	2.2	Relations: types, properties, equivalence relations, partial ordering	
	2.3	Functions: types, composition, inverse functions.	
3.0		**Combinatorics and Counting Techniques**	**9**
	3.1	Permutations and combinations, binomial theorem	
	3.2	Pigeonhole principle, recurrence relations, generating functions. Recurrence relations, introduction to generating functions	
4.0		**Graph Theory Fundamentals**	**9**
	4.1	Graphs: definitions, types, degree, paths, cycles. Trees, spanning trees, binary trees, graph traversal (BFS, DFS)	
	4.2	Connectivity, components, Euler and Hamiltonian graphs. Planar graphs, graph coloring, chromatic number.	
5.0		**Advanced Graph Theory**	**9**
	5.1	Graph representation: adjacency matrix, adjacency list, incidence matrix. Properties of incidence matrix, degree relations, representation comparison, Graph properties and structures: graph isomorphism, subgraphs, degree sequence, connected components.	
	5.2	Fundamental algorithms (intro): shortest path (Dijkstra's algorithm), minimum spanning tree (Prim's and Kruskal's algorithms), Advanced applications: network flow basics, real-world applications in routing, scheduling, and communication networks	

Text Books

1. J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, New York, NY, USA: McGraw-Hill, 2017.
2. K. H. Rosen, Discrete Mathematics and Its Applications, 8th ed., New York, NY, USA: McGraw-Hill, 2019.

References

1. N. Deo, Graph Theory with Applications to Engineering and Computer Science, New Delhi, India: PHI, 2016.
2. S. Lipschutz and M. Lipson, Discrete Mathematics, New York, NY, USA: Schaum's Outline, 2018.
3. D. B. West, *Introduction to Graph Theory*, 2nd ed., New Delhi, India: Pearson, 2017.

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 be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of Semester End 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		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UCSPC302	Computer Organization and Architecture	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
UCSPC302	Computer Organization and Architecture	10	10	5	25	40%	50	40%	75

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 fundamental structure, components, and organization of a computer system.
- To impart basic knowledge of arithmetic operations, algorithms, and Arithmetic Logic Unit (ALU) design.
- To enable students to understand CPU architecture, instruction execution, and control unit organization.
- To familiarize students with memory organization, cache memory, and virtual memory techniques.
- To develop competency in analyzing input/output organization and parallel processing concepts.

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 (hardwired and microprogrammed).
- **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**	**9**
	1.1	Introduction of computer system and its sub modules, Basic organization of computer and block level description of the functional units. Von newmann model,	
	1.2	Introduction to buses and connecting I/O devices to CPU and memory, Asynchronous and synchronous bus, PCI, SCSI.	
2.0		**Arithmetic and Logic Unit**	**9**
	2.1	Arithmetic and logical unit hardware implementation, Booth's Recoding, Booth's algorithm for signed multiplication,	
	2.2	Restoring division and non-restoring division algorithm, IEEE floating point number representation and operations.	
3.0		**Central processing unit.**	**9**
	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.	
	3.2	Multiplier control unit, Micro programmed control unit, basic concepts, Microinstruction sequencing and execution, Micro operations, concepts of nanoprogramming, Introduction to RISC and CISC architectures, design issues and examples of RISC processors.	
4.0		**Memory Organization.**	**9**
	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.	
	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.	
5.0		**I/O Organization and Multiprocessor Configurations.**	**9**
	5.1	Input/output systems, Programmed I/O, Interrupt driven I/O, I/O channels, DMA, Peripheral Devices, U.S.B.	
	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.	

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.
3. Carl Hamacher, Zvonko Vranesic and Safwat Zaky "Computer Organization", McGraw Hill, 6th Edition,2022.

References

1. John L. Hennessy and David Patterson," Computer Architecture A Quantitative Approach", Morgan Kaufman, 7th Edition,2025
2. Andrew S. Tanenbaum," Structured Computer Organization", Prentice-Hall India, 6th 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. FET shall be assessed for 5 marks separately.

Semester End Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of Semester End 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		
UCSPC303	Data structures	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
UCSPC303	Data structures	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		-	10	5	15	40%	50	40%	50

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 analyse problems and design efficient data structures and algorithms for solving them.

Pre-requisites: Programming using C/C++.

Course Objectives

- To introduce the fundamental concepts of data, data structures, abstract data types, and algorithm analysis.
- To develop problem-solving skills using linear data structures such as arrays and linked lists.
- To enable students to analyse and implement 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 competency in implementing nonlinear data structures such as trees and graphs for solving complex computational problems.

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

- **CO1** **Define**[1]fundamental concepts of data structures, abstract data types, and algorithm complexity**.**
- **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** **Analyse**[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**	**6**
	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.	
2.0		**Linked List**	**6**
	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 ListsReversing One-wayLinked List, Circular Linked List (CLL) and its operations.	
	2.2	Implementation of Doubly Linked List (DLL), Operations (Insertion, Deletion, searching) of DLL. Application of Linked List (SLL, CLL, DLL).	
3.0		**Stack and queue**	**6**
	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.	
	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.	
4.0		**Searching and Hashing**	**6**
	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.	
	4.2	Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and Rehashing.	
5.0		**Nonlinear Data Structure**	**6**
	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.	
	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.	

Text Books

1. Data Structures- A Pseudocode Approach with C by Richard F. Gilberg and Behrouz A. Forouzan, Cengage Learning ,2nd Edition ,2015.
2. Data Structures with C Schaum's Outlines Series by S. Lipschutz Tata McGraw-Hil, 2nd Edition, 2017.
3. Data Structure using C by Reema Thareia, Oxford, 2nd edition ,2014.

References

1. Data Structure using C A. M. Tenenbaum, Y. Langsam, M.J. Augenstein Prentice- Hall of India Pvt. Limited 2023.
2. Understanding Pointers in C Yashavant Kanetkar BPB Publication I 2019.

Internal Assessment (T1, T2 and FET)

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

Semester End Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of Semester End 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		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UCSPC304	Information Theory and Coding	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
		T1	T2	FET	Total	Min. Passing %	Marks	Min pass	Total (Marks)
UCSPC304	Information Theory and Coding	10	10	5	25	40%	50	40%	75

Course Description:

This course introduces the fundamentals of information theory, source coding, data compression, error control coding, number theory, and cryptography. It focuses on efficient data representation, reliable communication techniques, and the mathematical foundations required for modern encryption and secure data transmission.

Course Objectives:

- To introduce the fundamental principles of information theory and source coding.
- To develop skills in analyzing and implementing various data compression techniques.
- To enable students to understand error detection and correction methods for reliable communication.
- To apply number theory concepts in modern cryptographic systems.

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

- **CO1** **Illustrate**[2] concepts of information theory, entropy, source coding, and channel capacity.
- **CO2** **Apply**[3] compression algorithms such as Huffman, LZW, and JPEG for efficient data representation.
- **CO3** **Analyse**[4] error control coding techniques including linear block, cyclic, and convolution codes.
- **CO4** **Apply**[3] number theory and cryptographic techniques for secure communication systems.
- **CO5** **Analyze**[4] cryptographic techniques including symmetric and asymmetric key algorithms for secure communication.

Course Contents

Module	Unit	Description	Hours
1.0		**Information Theory & Source Coding**	
	1.1	Introduction to Information Theory, Entropy & Types of Entropy.	**6**
	1.2	Source Coding, Prefix Coding, Channel Capacity	
2.0		**Compression Algorithms**	
	2.1	Optimal Compression, Compression Algorithms, Huffman Coding, Adaptive Huffman Compression, Dictionary Based Compression	**12**
	2.2	Speech Compression, Sliding Window Compression, LZW, RLE, Lossy & Lossless Compression Schemes, Image Compression – GIF, JPEG	
3.0		**Error Control Coding Techniques**	
	3.1	Types of Codes. Error Checking & Correcting Codes	**6**
	3.2	Linear Block Codes, Cyclic Codes, BCH Codes, Convolution Codes	
4.0		**Number Theory**	
	4.1	Modular Arithmetic, Solving ax+by=d, Congruences, Chinese Remainder Theorem, Modular Exponentiation, Fermat's Little and Euler Theorem	**9**
	4.2	Prime Number Generation, Random Number Generation, Primitive Roots, Legendre and Jacobi Symbols, Discrete Probability, Discrete Logarithms	
5.0		**Cryptographic Techniques**	
	5.1	Security Goals, Threats and Attack on Information, Classic Cryptography, Symmetric Key Cryptography – Stream Ciphers, Block Cipher, Stream Cipher, DES, Triple DES, AES	**12**
	5.2	Public and Private Key Cryptography – RSA, Diffie-Hellman, Hash Function – MD5, SHA-1, Digital Signature	

Textbooks

1 William Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education, 8th Edition, 2022.

2 Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, Wiley, 2nd Edition, 2006.

References

1 Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press, 3rd Edition, 2020.

2 David J. C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2013.

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 (MCQ quiz) will be assessed for 5 marks separately.
3. Internal Term work of 25 mark of tutorial.

Semester End Examination:

1. Question paper will comprise of 5 questions, each carrying 10 marks.
2. The duration of semester examination end 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		
UCMEM001	Engineering Economics	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
UCMEM001	Engineering Economics	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		-	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.

Course Objectives

- To introduce the importance of economy in engineering field.
- To develop the ability to apply knowledge of economic comparisons for selecting best alternative.
- To introduce the concept of depreciation.
- To explain the importance of working capital management

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

- **CO1** **Recall**[1] the importance of economic principles in engineering practice and **relate**[2] their role in project feasibility.
- **CO2** **Define**[1] and **explain**[2] national income concepts (GDP, GNP, NNP, per capita income) and methods of measurement, **Analyze**[4] the causes and effects of inflation and Business cycle.
- **CO3** **Apply**[3]comparative economic methods to **analyze**[4]and select optimal alternatives in engineering decision-making.
- **CO4** **Analyze**[4] different depreciation methods and **Evaluate**[5] working capital management techniques.

Course Contents

Module	Unit	Description	Hours
1.0		**Fundamentals of Engineering Economics**	**6**
	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		**Macroeconomics in Engineering**	**6**
	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**	**6**
	3.1	Present worth comparisons, Comparisons of assets with equal, unequal lives, comparison of deferred investments, Annual worth method, Rate of return method, Future worth method, NPV Method.	
4.0		**Decision Tools for Financial Viability**	**6**
	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**	**6**
	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. G. N. Gregory Mankiw, Microeconomics, 7th ed. Mason, OH, USA: South-Western Cengage Learning, 2021
2. C. S. Chan S. Park, Fundamentals of Engineering Economics, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2020.
3. R. Rudiger Dornbusch, S. Stanley Fischer, and R. Richard Startz, Macroeconomics, 10th ed. New York, NY, USA: McGraw-Hill Irwin, 2020.
4. R. R. R. R. Gupta, Engineering Economics and Financial Management. New Delhi, India: S. Chand Publishing, 2022.

Reference Books:

1. Panneer Selvam, R, "Engineering Economics," Prentice Hall of India Ltd, New Delhi, 2021.

Internal Assessment (T1, T2 and FET)

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

Semester End Examination

1. Question paper will be of 35 marks comprised of 5 questions, each carrying 07 marks.
2. The duration of semester examination end 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.2to Q.5) will be selected from all the modules.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
UCSPC305	Computer Organization and Architecture Lab	**Theory**	**Practical**	**Tutorial**	**Theory**	**Practical**	**Tutorial**
		-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
UCSPC305	Computer Organization and Architecture Lab	**Term work**	**Min pass**	**Marks**	**Min pass**	**Total (Marks)**
		25	40%	-	-	25

Course Objectives

- To provide practical knowledge of number system, memory organization, instruction execution and I/O operation using programming and simulation tools.

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

CO1 **Implement**[3] number system conversions and arithmetic algorithms using programming or simulation tools.

CO2 **Demonstrate**[4] memory organization, instruction execution, and I/O operations through practical experiments.

All the experiments will be based on the course content of **UCSPC302 Computer Organization and Architecture**

The following list of experiments is indicative and intended to serve as a reference framework for the practical component of the course. The Course Coordinator, in consultation with the concerned faculty, shall design, modify, and finalize the laboratory experiments based on batch size, available infrastructure, and recent technological advancements.

Efforts should be made to ensure that the experiments:

- Align with the stated Course Outcomes (COs) and program objectives.
- Encourage hands-on learning, problem-solving, and analytical thinking.
- Incorporate the use of modern tools, software, and simulation platforms wherever applicable.
- Promote mini-projects, case studies, or application-based experiments to enhance practical exposure.

The finalized list of experiments may be reviewed periodically to maintain relevance with industry trends and academic standards.

Suggested List of Experiments

1. **Study of Basic Computer Organization**
    - Identify functional units (ALU, CU, Memory, I/O) using block diagrams or simulation tools.
2. **Simulation of Number System Conversions**
    - Implement conversion between binary, decimal, octal, and hexadecimal.
3. **Implementation of Binary Arithmetic Operations**
    - Perform addition, subtraction, multiplication, and division using programs.
4. **Implementation of Booth's Multiplication Algorithm**
    - Write a program to perform signed binary multiplication using Booth's algorithm.
5. **Simulation of Division Algorithms**
    - Implement restoring and non-restoring division algorithms.
6. **Floating Point Representation and Operations**
    - Simulate IEEE floating-point representation and basic arithmetic operations.
7. **Study of Instruction Formats and Addressing Modes**
    - Demonstrate different instruction formats and addressing modes (e.g., Intel-based examples).
8. **Cache Memory Simulation**
    - Implement cache mapping techniques (direct, associative, set-associative).
9. **Simulation of Virtual Memory**
    - Demonstrate paging and segmentation using program or tools.
10. **I/O Techniques and DMA Simulation**
    - Simulate programmed I/O, interrupt-driven I/O, and DMA operations.

Text Books

1. Miles Murdocca, "Computer Architecture and Organization", Wiley India,2021
2. William Stallings, "Computer Organization and Architecture: Designing and performance": Prentice-Hall India,2019
3. Carl Hamacher, Zvonko Vranesic and Safwat Zaky "Computer Organization", McGraw Hill,2022

References

1. John L. Hennessy and David Patterson," Computer Architecture A Quantitative Approach", Morgan Kaufman.2019
2. Andrew S. Tanenbaum," Structured Computer Organization", Prentice-Hall India,2021

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		
UCSPC306	Data structures Lab	**Theory**	**Practical**	**Tutorial**	**Theory**	**Practical**	**Tutorial**
		-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
UCSPC306	Data structures Lab	**Term Work**	**Min pass**	**Marks**	**Min pass**	**Total (Marks)**
		25	40%	25	40%	50

Course Objective:

- To equip students with practical knowledge and programming skills for implementing various data structures and algorithms efficiently.

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

CO1 **Apply**[3] operations on linear and non-linear data structures to solve problems.

CO2 **Apply**[3] searching and sorting techniques for efficient solution design.

All the experiments will be based on the course content of **Data Structure – UCSPC303**

The following list of experiments is indicative and intended to serve as a reference framework for the practical component of the course. The Course Coordinator, in consultation with the concerned faculty, shall design, modify, and finalize the laboratory experiments based on batch size, available infrastructure, and recent technological advancements.

Efforts should be made to ensure that the experiments:

- Align with the stated Course Outcomes (COs) and program objectives.
- Encourage hands-on learning, problem-solving, and analytical thinking.
- Incorporate the use of modern tools, software, and simulation platforms wherever applicable.
- Promote mini-projects, case studies, or application-based experiments to enhance practical exposure.

The finalized list of experiments may be reviewed periodically to maintain relevance with industry trends and academic standards.

Suggested List of Experiments

1. **Implement operations on array using functions and pointers:**
   - 1.1. Develop a program to perform operations on a list of elements using arrays.
   - 1.2. Design a modular program using user-defined functions to perform arithmetic and logical operations.
   - 1.3. Implement swapping of two numbers using pointers and array traversal using pointers.
   - 1.4. Develop a program to define and manipulate structures for storing records of entities (e.g., student details).
2. **Implement Operations on Singly Linked List**
   - 2.1. Creation and Display of SLL
   - 2.2. Insertion at Beg, at end and at position
   - 2.3. Deletion at Beg, at end and at position
   - 2.4. Reverse a Singly Linked List
   - 2.5. Searching a Singly Linked List
3. **Implement operations on Doubly Linked List**
   - 3.1. Creation and Display of DLL
   - 3.2. Insertion at Beg, at end and at position
   - 3.3. Deletion at Beg, at end and at position
4. **Implement operations on Circular Linked List**
   - 4.1. Creation and Display of CLL
   - 4.2. Insertion at Beg, at end and at position
   - 4.3. Deletion at Beg, at end and at position
5. **Implement operations on Stack**
   - 5.1. Develop a program to perform push, pop and display operations using array
   - 5.2. Develop a program to perform push, pop and display operations using linked list
   - 5.3. Develop a program to convert infix to prefix/postfix notation
   - 5.4. Develop a program to perform balanced parenthesis
6. **Implement operations on Queue**
   - 6.1. Develop a program to perform enqueue, dequeue and display operations using array
   - 6.2. Develop a program to perform enqueue, dequeue and display operations using linked list
7. **Implement operations on Special types of Queues**
   - 7.1. Develop a program to perform enqueue, dequeue and display operations on Circular Queue
   - 7.2. Develop a program to perform enqueue, dequeue and display operations on Double Ended Queue
8. **Implement Different Searching and Sorting algorithms**
   - 8.1. Develop a program to search an element using Linear Search and Binary Search
   - 8.2. Develop a program to sort the data using Bubble Sort
   - 8.3. Develop a program to sort the data using Insertion Sort
   - 8.4. Develop a program to sort the data using Selection Sort
9. **Implement Operations on Binary Search Tree**
   - 9.1. Creation of BST
   - 9.2. Inorder, preorder and postorder Traversal of BST
10. **Implement Operations on Graph**
    - 10.1. Creation of BST
    - 10.2. Traversal using Adjacency list

References

1. Data Structure using C A. M. Tenenbaum, Y. Langsam, M.J. Augenstein Prentice- Hall of India Pvt. Limited 2023.
2. Understanding Pointers in C Yashavant Kanetkar BPB Publication I 2019.

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		
UCSFP301	Field Project I	**Theory**	**Practical**	**Tutorial**	**Theory**	**Practical**	**Tutorial**
		-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
UCSFP301	Field Project I	**Term Work**	**Min pass**	**Marks**	**Min pass**	**Total (Marks)**
		25	40%	25	40%	50

Course Description:

The objective of this course is to provide second-year engineering students with a solid foundation in statistical analysis and numerical techniques, enabling them to apply these methods effectively in engineering problem-solving and decision-making contexts.

Prerequisite: Students should have prior knowledge of

1. Basic understanding of social and ethical responsibilities
2. Teamwork and communication skills acquired in prior coursework or group activities
3. Familiarity with problem-solving methodologies and project planning
4. Conversation in local language

Course Objectives

- To establish a mutually beneficial relationship between the college and the community.
- To provide opportunities for students to engage with the local community, fostering empathy, teamwork, and problem-solving skills.
- To develop awareness of the challenges faced by the local community and the role of engineering in addressing them.
- To enable students to apply technical knowledge and skills for designing solutions that create a positive impact on the community.
- To develop skills in evaluating and critically analyzing the outcomes of community engagement activities for sustainable impact.

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 awarded by 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 carried out 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

Project Scope:

- Conduct workshops or awareness drives on topics like digital literacy, environmental sustainability, 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.

Text Books:

1. Waterman, A. Service-Learning: A Guide to Planning, Implementing, and Assessing Student Projects. Routledge, 1997.

2. Beckman, M., and Long, J. F. Community-Based Research: Teaching for Community Impact. Stylus Publishing, 2016.

3. Design Thinking for Social Innovation. IDEO Press, 2015.

4. Dostilio, L. D., et al. The Community Engagement Professional's Guidebook: A Companion to The Community Engagement Professional in Higher Education. Stylus Publishing, 2017

Reference Books:

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based earning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core text book, 2 Indian Edition, by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden

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.

*****

Programme Structure for B Tech.: Second Year Computer Science and Engineering (R25)
Semester IV: Teaching Scheme

Course code	Course Name	Teaching scheme (Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
UCSPC401	Operating System	3	-	-	3	-	-	3
UCSPC402	Database Management	3	-	-	3	-	-	3
UXXMM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
UXXOE02X	Open Elective II	2	-	-	2	-	-	2
UCMEM003	Project Management	2	-	-	2	-	-	2
UCSVS401	Software Proficiency I (Java Programming)	-	4	-	-	2	-	2
UCSPC403	Database Management System Laboratory	-	4	-	-	2	-	2
UCSVE003	Open-Source Technologies		2	-		1	-	1
UHSAE008	Employability Skills	1	2	-	1	1	-	2
USTSVE005	AI Ethics and Governance	1	-	1	1	-	1	2
Total		**14**	**12**	**1**	**14**	**6**	**1**	**21**

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				
UCSPC401	Operating System	10	10	5	25	50	-	-	75
UCSPC402	Database Management	10	10	5	25	50	-	-	75
UXXMM0XX	Multidisciplinary Minor II	-	10	5	25	35	-	-	50
UXXOE02X	Open Elective II	-	10	5	15	35	-	-	50
UCMEM003	Project Management	-	10	5	15	35	-	-	50
UCSVS401	Software Proficiency I (Java Programming)	-	-	-	-	-	25	25	50
UCSPC403	Database Management System Laboratory	-	-	-	-	-	25	25	50
UCSVE003	Open-Source Technologies	-	-	-	-	-	25	-	25
UHSAE008	Employability Skills	-	-	-	-	-	25	25	50
USTSVE005	AI Ethics and Governance	-	-	-	-	-	25	-	25
								Total	**525**

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		
UCPCS401	Operating System	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
UCSPC401	Operating System	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		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 introduce the basic concepts of operating systems and their services.
- To enable students to understand processes, process states, Process Control Blocks (PCB), and various CPU scheduling algorithms for performance evaluation.
- To impart basic knowledge of process synchronization, deadlocks, and deadlock detection and prevention methods.
- To familiarize students with memory management strategies and file system design concepts.

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**	**9**
	1.1	Introduction to Operating Systems, Operating System structure, Types of Operating Systems, Operating System Services, and Views of Operating System.	
	1.2	System calls, Types of system Calls, System programs, Kernel and its types. Comparison of various OS. (Windows, Linux, Mac and Android)	
2.0		**Process Management**	**9**
	2.1	Process concept: Basic concepts, Process States, Process Control Block, Context switch, Operations on processes, Inter-process communication, Threads.	
	2.2	Process Scheduling: Scheduling criteria, Types of schedulers, Scheduling algorithms, Multiple-Processor scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling.	
3.0		**Process Synchronization and Deadlock**	**9**
	3.1	Background, the critical section problem, Peterson's solution, synchronization hardware, Semaphores, classic problems of Synchronization, Monitor.	
	3.2	System model, deadlock characterization, methods for handling deadlocks. Deadlock preventions, deadlock avoidance, deadlock detection, deadlock recovery.	
4.0		**Memory Management**	**9**
	4.1	Memory Management Strategies: Background, swapping, contiguous memory allocation, paging, structure of the page table, Segmentation.	
	4.2	Virtual Memory Management: Background, demand paging, copy-on-write, page replacement, Thrashing.	
5.0		**File Management& I/O Subsystem**	**9**
	5.1	File System: File concept, access methods, Directory and disk structure, Disk Scheduling, file-system mounting, file sharing, and protection.	
	5.2	Overview of I/O system, I/O hardware, Application I/O interface, Kernel I/O subsystem.	

Text Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", John Wiley, 10th Edition, 2018.
2. Operating Systems: Internals and Design Principles William Stallings Pearson Education International 8th Edition, 2014.
3. Operating Systems - A Concept Based approach Dhananjay M Dhamdhere Tata McGraw Hill 4th Edition, 2007.

References

1. Operating System A Design Oriented Approach Charles Crowley Tata McGraw Hill 3rd Edition.
2. Operating System with Case Studies in Unix, Netware and Windows NT Achyut S. Godbole Tata McGraw Hill 5th Edition, 2007.

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 be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of Semester End 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		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UCSPC402	Database Management System	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
UCSPC402	Database Management System	10	10	5	25	40%	50	40%	75

Course Description: This course provides second-year computer engineering students with a comprehensive understanding of Database Management Systems (DBMS). Through theoretical lectures and practical exercises, students will learn the fundamental concepts, principles, and techniques used in the design, implementation, and management of databases.

Course Objectives:

- To introduce the fundamental concepts of Database Management Systems (DBMS).
- To develop proficiency in relational data modeling techniques, including Entity-Relationship (ER) modeling.
- To enable students to apply normalization principles for designing efficient and scalable databases.
- To develop practical skills in using Structured Query Language (SQL), PL/SQL, and NoSQL for database querying and management.
- To familiarize students with advanced database concepts such as transaction management, concurrency control, and data integrity mechanisms for reliable and secure data management.

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

- **CO1** **Explain**[2] database fundamentals and **design** ER and Extended ER models.
- **CO2** **Apply**[3] relational model concepts and **write** SQL queries for data manipulation.
- **CO3** **Analyze**[4] functional dependencies and **normalize** database schemas.
- **CO4** **Demonstrate**[2] data storage, indexing, transaction management, and concurrency control concepts.
- **CO5** **Compare**[2] Relational and NoSQL databases and develop applications using MongoDB.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction to Databases and ER Model**	
	1.1	Introduction to Database: File system v/s Database system, Need of DBMS, Definition of Database and DBMS, Characteristics, Advantages and Disadvantages of DBMS, Database Users and Administrators, Database Architecture (1-tier, 2-tier, 3-tier), Data Independence, Schema and Instance, Data Models.	**6**
	1.2	ER model: Entities, Attributes, Types of attributes: Simple, Composite, Single-valued, Multi-valued, Derived, Keys: Super key, Candidate key, Primary key, foreign key, Advanced Keys, Relationship, Cardinality constraints, Participation constraints, Weak entity sets, Extended ER features- Generalization, Specialization, Aggregation.	
2.0		**Relational Model, SQL and PL/SQL**	
	2.1	Relational Model: Structure of Relational Database, Reduction of ER model into Relational schemas, Integrity constraints: Domain constraints, Entity integrity, Referential integrity, Introduction to relational algebra, Basic operations, Additional operations, Queries using relational algebra.	**12**
	2.2	SQL (Structured Query Language): Introduction to SQL, SQL data types, Basic and Advance Operators, DDL commands, DML commands, Constraints, Aggregate functions, GROUP BY, HAVING, ORDER BY, Advanced Clauses, Sub-queries, Joins, Views, Indexes.	
	2.3	PL/SQL: Procedures, Functions, Packages, Synonyms, Sequences, Triggers, Assertions, Cursor.	
3.0		**Normalization and Indexing**	
3	3.1	Importance of a good schema design, Motivation for normal forms, Functional Dependency and its types, Normalization, Types- 1NF, 2NF, 3NF, BCNF, 4NF and 5NF.	9
	3.2	Indexing: File organization, Organization of records in files, Data Dictionary, Database Buffer, Ordered Indices-Primary, Secondary, Multilevel, B+ Tree Index, Hashing, Hash Indices, Dynamic hashing, Multiple key access, Bitmap Indices.	
4.0		**Transaction Management & Concurrency Control**	
4	4.1	Transaction Processing: ACID properties, Transaction states, Storage Structure, Implementation of atomicity, isolation and durability, Serializability, Testing of Serializability.	9
	4.2	Concurrency Control: Lock based protocols, Timestamp based Protocols, Validation based Protocols, Multiple Granularities, Deadlock handling.	
5.0		**NoSQL Databases**	
	5.1	NoSQL: ACID and BASE for reliable database transactions Achieving horizontal scalability with Data base sharding, Brewers CAP theorem., Eventual Consistency in Distributed systems, Types, of NoSQL databases.	9
	5.2	MongoDB: Definition and need of MongoDB. JavaScript programming, Datatypes in MongoDB. MongoDB Query Language: Insert, save methods, add/remove fields to/from documents, arrays, aggregate functions, Mapreduce function, Mongo import, Mongo export.	

Text Books

1. A. Silberschatz, H.F. Korth, S.Sudarshan, A Database System Concepts, McGraw Hill Education,7thEdition, 2020.
2. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publications, 4th Edition, 2018.
3. Kristina Chodorow, MongoDB: The Definitive Guide, O'Reilly Publications,2019.

Reference Books

1. Ramez Elmasri and Shamkant Navathe, Fundamentals of Database Systems, Pearson Education,7thEdition, 2017.

Internal Assessment (T1, T2 and FET):

1. T2 (Test 2) should be based on first four modules, for 10 marks.
2. Fifth module will be assessed for 5 marks separately.

End Semester Examination:

1. Question paper will comprise of 5 questions, each carrying 7 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 Unit I.
5. Remaining question (Q.2 to Q.5) will be selected from modules 2 to 5 respectively.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UCMEM002	Project Management	02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Semester End Exam (SEE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
UCMEM002	Project Management	-	10	5	15	40%	35	40%	50

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 on the management side of engineering projects.

Pre-requisites: -

Course Objectives

- To introduce the principles of management and their application in engineering contexts.
- To introduce the procedure of analysis of organizational structures, leadership styles, and decision-making processes.
- To develop the skills to apply project management techniques to engineering problems.
- To explain the procedure for evaluation of financial and economic aspects of engineering projects.
- To develop skills in strategic planning, innovation, and technology management.

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

- **CO1** **Explain**[2] the fundamentals of management and organizational behaviour.
- **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 Behaviour**	**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**
	4.1	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 and innovation strategies. Change management in engineering organizations. Case studies in engineering management	

Text Books

1. Pravin Kumar, Engineering Management, Pearson Education, 2013
2. Harold Koontz & Heinz Weihrich – Essentials of Management K. Anil Kumar – Engineering Management McGraw-Hill Education, 10th Edition, 2015

References

1. Peter F. Drucker – Management: Tasks, Responsibilities, Practices Harper & Row (reprint: HarperBusiness), 1973
2. James A. F. Stoner – Management Prentice Hall, 1982
3. Harold Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Wiley, 11th Edition, 2013
4. Samuel J. Mantel Jr., Jack R. Meredith, Scott M. Shafer & Margaret M. Sutton, Project Management in Practice, Wiley, 4th Edition, 2011

Internal Assessment (T1, T2 and FET)

1. T2 should be based on First to Fourth modules, for 10 marks.
2. 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 07 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

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UCSVS401	Software Proficiency-I (Java Programming)	-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term work	Min pass	Marks	Min pass	Total (Marks)
UCSVS401	Software Proficiency-I (Java Programming)	25	40%	25	40%	50

Pre-requisites: - (USTVS001) C++ Programming

Course Objective: To provide practical exposure to object oriented programming concept using Java.

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

- **CO1** **Apply**[3] core Java programming concepts including control statements, methods, constructors, inheritance, interfaces, arrays, strings, packages, exception handling, and multithreading to develop object-oriented programs.
- **CO2** **Design**[6] Java applications using Swing for graphical user interfaces and JDBC for database connectivity to perform data-driven operations.

Course Contents

Module	Unit	Description	Hours
1.0		**Java Basics and Control Statements**	
	1.1	Fundamental Programming: The Java Keywords, Bytecode, The Java Programming Environment- JDK, JVM, JIT Compiler, A Simple Java Program, Source File Declaration Rules, Comments, Data Types, Variables, Operators, Input and Output, Command line arguments, Object Oriented Programming features.	**9**
	1.2	Control Statements: Java Selection Statements- if, switch, Iteration statements- while, do-while, for, Nested Loops, jump statements- break, continue, return.	
2.0		**Methods, Constructors and Inheritance**	
	2.1	Methods: Static and non-static methods, Return type and non-return type methods, Method Overloading.Constructors: Use of Constructor, Types of constructors, Constructor Overloading. Access specifiers, this keyword.	**9**
	2.2	Inheritance: Definition, Use of Inheritance, Types of Inheritance, Super keyword, Method Overriding, final keyword, abstract keyword.	
3.0		**Interfaces, Arrays, Strings and Packages**	
	3.1	Interfaces: Defining and Implementing an Interface. Arrays: One-Dimensional Arrays, Multi-dimensional Arrays Strings: String class, String methods, String manipulations.	**9**
	3.2	Packages: Creating a Package, loading a class, naming a Package, Using Package Members, Managing Source and Class Files. Developing and deploying (executable) Jar File.	
4.0		**Exception Handling and Multithreading**	
	4.1	Exception Handling: Definition, The Classification of Exceptions, try-catch block, Multiple catch clauses, Nested try statements, throw, throws and finally keyword, Generating custom exceptions.	**9**
	4.2	Multithreading: Processes and Threads, Runnable Interface and Thread Class, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep.	
5.0		**Graphical User Interfaces using Swing and JDBC**:	
	5.1	Graphical User Interfaces using Swing: Introduction to AWT, Introduction to swing package**,** Swing containers-JFrame, JPanel, JDialog**,** Basic User Interface components, Event Handling, Introduction to Layout Managers.	**9**
	5.2	JDBC Programming: Basics of DBMS, Structure Query Language, JDBC Introduction, JDBC Drivers, Database Connectivity with Oracle/MySQL, ResultSet, Performing Create, Read, Update and Delete operations on database.	

The following list of experiments is indicative and intended to serve as a reference framework for the practical component of the course. The Course Coordinator, in consultation with the concerned faculty, shall design, modify, and finalize the laboratory experiments based on batch size, available infrastructure, and recent technological advancements.

Efforts should be made to ensure that the experiments:

- Align with the stated Course Outcomes (COs) and program objectives.
- Encourage hands-on learning, problem-solving, and analytical thinking.
- Incorporate the use of modern tools, software, and simulation platforms wherever applicable.
- Promote mini-projects, case studies, or application-based experiments to enhance practical exposure.

The finalized list of experiments may be reviewed periodically to maintain relevance with industry trends and academic standards.

Suggested List of Experiments

1. Implement basic java programs based on I/O and operators.
2. Implement basic java programs based on control statements.
3. Implement Java programs using constructors and constructor overloading.
4. Implement Java programs using methods (static, non-static, recursive) and method overloading.
5. Implement Java programs to demonstrate Inheritance and method overriding.
6. Implement basic java programs based on abstract keyword, final keyword and interfaces.
7. Implement basic java programs based on Strings.
8. Implement basic java programs based on arrays.
9. Implement a Java program to demonstrate Exception handling.
10. Implement Java programs using pre-defined and user defined packages.
11. Implement Java programs to demonstrate Multithreading using Thread class and runnable Interface.
12. Design a Graphical User Interface (GUI) using Java Swing package.
13. Implement Java Database Connectivity (JDBC) programming.

Textbook:

1. Herbert Schildt, Java: The Complete Reference,12th Edition, McGraw-Hill Education, 2022.

Reference book:

1. Herbert Schildt, Java A Beginner's Guide, 8th Edition, McGraw-Hill, 2020.
2. D.T. Editorial Services, Java 8 Programming Black Book, Dreamtech Press, 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		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UCSPC403	Database Management System Laboratory	-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam(OE/POE)		
		Term work	Min pass	Marks	Min pass	Total (Marks)
UCSPC403	Database Management System Laboratory	25	40%	25	40%	50

Course Objective:

- To equip student with practical experience in database applications.

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

- **CO1** **Design**[5] ER models, convert them into relational schemas, and implement databases using SQL.
- **CO2** **Implement**[3] advanced SQL, normalization, transaction management, and PL/SQL/NoSQL queries.

All the experiments will be based on the course content of **Database Management System UCSPC402.**
The following list of experiments is indicative and intended to serve as a reference framework for the practical component of the course. The Course Coordinator, in consultation with the concerned faculty, shall design, modify, and finalize the laboratory experiments based on batch size, available infrastructure, and recent technological advancements.

Efforts should be made to ensure that the experiments:

- Align with the stated Course Outcomes (COs) and program objectives.
- Encourage hands-on learning, problem-solving, and analytical thinking.
- Incorporate the use of modern tools, software, and simulation platforms wherever applicable.
- Promote mini-projects, case studies, or application-based experiments to enhance practical exposure.

The finalized list of experiments may be reviewed periodically to maintain relevance with industry trends and academic standards.

Suggested List of Experiments

1. Installation and Demonstration of DBMS Software (e.g. Oracle, MySQL etc.).
2. Design **ERdiagram** for a given real-world application (e.g., Library Management System / Student Result System etc.) including entities, attributes, keys, relationships, and constraints.
3. Convert the **ER model into Relational Schema**.
4. Implement DDL, DML and DQL Commands.
5. Implement **SQL** Aggregate functions, GROUP BY, HAVING, ORDER BY and LIMIT clauses.
6. Implement SQL SET Operations with advance operators.
7. Implement SQL subqueries.
8. Implement **Views and Indexes** and analyze their effect on data access.
9. Implement procedures, functions and cursors in PL/SQL.
10. Implement Synonyms, Sequences, Triggers and Packages in PL/SQL.
11. Identify **functional dependencies** and perform **normalization (1NF to BCNF)** for a given database schema.
12. Implement database connectivity with any high-level programming language.
13. Implement CRUD operations in MongoDB.
14. Implement Arrays and Aggregate functions in MongoDB.
15. Demonstrate mongo import and mongo export.

Text Books

1. A. Silberschatz, H.F. Korth, S.Sudarshan, A Database System Concepts, McGraw Hill Education,7thEdition, 2020.
2. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publications, 4th Edition, 2018.
3. Kristina Chodorow, MongoDB: The Definitive Guide, O'Reilly Publications.

Reference Books

1. Ramez Elmasri and Shamkant Navathe, Fundamentals of Database Systems, Pearson Education, 7thEdition, 2017.

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		
UCSVE003	Open-Source Technologies	**Theory**	**Practical**	**Tutorial**	**Theory**	**Practical**	**Tutorial**
		-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
UCSVE003	Open-Source Technologies	**Term Work**	**Min pass**	**Marks**	**Min pass**	**Total (Marks)**
		25	40%	-	-	25

Course Description

This course introduces students to the philosophy, ecosystem, and practical usage of Open-Source Software (OSS). It provides hands-on experience with Linux operating systems, shell scripting, Git-based version control, collaborative development using GitHub, and deployment using open-source tools. The course emphasizes practical implementation and encourages students to participate in open-source communities through mini-project development and repository contributions.

Course Objectives

- To introduce the concepts of open-source philosophy and licensing models.
- To develop skills in installing and working with Linux and open-source tools.
- To enable students to apply Git and GitHub for version control and collaborative development.
- To develop competency in building and deploying simple applications using open-source technologies.

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

- **CO1** **Demonstrate**[2] open-source concepts and install3/configure Linux and related tools in a virtualized environment.
- **CO2** **Use**[3] Git/GitHub and develop4 applications using open-source technologies such as Linux, databases, and Docker.

The following list of experiments is indicative and intended to serve as a reference framework for the practical component of the course. The Course Coordinator, in consultation with the concerned faculty, shall design, modify, and finalize the laboratory experiments based on batch size, available infrastructure, and recent technological advancements.

Efforts should be made to ensure that the experiments:

- Align with the stated Course Outcomes (COs) and program objectives.
- Encourage hands-on learning, problem-solving, and analytical thinking.
- Incorporate the use of modern tools, software, and simulation platforms wherever applicable.
- Promote mini-projects, case studies, or application-based experiments to enhance practical exposure.

The finalized list of experiments may be reviewed periodically to maintain relevance with industry trends and academic standards.

Suggested List of Practical's

Practical 1: Study of Open-Source Environment

1. Study of Free Software vs Open-Source Software
2. Identification of commonly used open-source tools in academia and industry
3. Exploration of different open-source licenses (GPL, MIT, Apache)

Practical 2: Installation of Linux Operating System

1. Installation of Linux (Ubuntu/Fedora) using VirtualBox
2. Configuration of system settings
3. Understanding boot process and directory structure

Practical 3: Basic Linux Commands

1. Navigation commands: pwd, ls, cd
2. File operations: mkdir, rmdir, cp, mv, rm
3. Viewing files: cat, more, less, head, tail
4. Searching files: grep, find

Practical 4: File Permissions and User Management

1. chmod, chown, chgrp commands
2. Creating and managing users and groups
3. Understanding read, write, execute permissions

Practical 5: Package Management

1. Installing and removing software using apt/yum
2. Updating system packages
3. Managing dependencies

Practical 6: Shell Scripting – Basics

1. Writing simple shell scripts
2. Variables and operators
3. Conditional statements (if, case)
4. Loops (for, while)

Practical 7: Shell Script Automation

1. Script for file backup automation
2. Script for system monitoring (disk usage, memory usage)
3. Scheduling tasks using cron

Practical 8: Installation and Configuration of Git

1. Installing Git
2. Git configuration (username, email)
3. Understanding Git workflow

Practical 9: Git Repository Management

1. Creating local repository
2. Adding and committing files
3. Viewing logs and status
4. Undoing changes

Practical 10: Branching and Merging in Git

1. Creating and switching branches
2. Merging branches
3. Resolving merge conflicts

Practical 11: GitHub Collaboration

1. Creating GitHub account
2. Creating remote repository
3. Push, pull, clone operations
4. Managing pull requests and issues

Practical 12: Open-Source IDE Usage

1. Installation of VS Code / Eclipse
2. Installing extensions
3. Running and debugging simple programs

Practical 13: Working with Open-Source Database

1. Installation of MySQL/PostgreSQL
2. Creating database and tables
3. Performing CRUD operations

Practical 14: Containerization using Docker

1. Installation of Docker
2. Creating Dockerfile
3. Building Docker image
4. Running containerized application

Practical 15: Mini Project (Compulsory)

1. Students shall:
2. Develop a small application using open-source stack
3. Maintain source code using Git
4. Host project on GitHub
5. Demonstrate deployment using Linux or Docker
6. Submit project report and source code repository link

Text Books

1 C. Negus, Linux Bible, 11th ed., Hoboken, NJ, USA: Wiley Publishing, Nov. 2025

2 S. Chacon and B. Straub, Pro Git, 2nd ed., New York, NY, USA: Apress, 2014.

References

1 E. S. Raymond, The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, Sebastopol, CA, USA: O'Reilly Media, 2001

2 K. Fogel, Producing Open-Source Software: How to Run a Successful Free Software Project, Sebastopol, CA, USA: O'Reilly Media, 2005.

3 E. Nemeth, G. Snyder, T. R. Hein, B. Whaley, and D. Mackin, UNIX and Linux System Administration Handbook, 5th ed., Boston, MA, USA: Addison-Wesley Professional, 2017.

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	Total Credits
UHSAE008	Employability Skills	01	02	-	01	01	-	02

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		Semester End Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
UCSVE003	Open-Source Technologies	25	40%	25	-	50

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.

Pre-requisites: -

Course Objectives

To develop greater self-awareness regarding their strengths, interests, learning gaps, and professional aspirations.

- To communicate more clearly and confidently in academic and workplace-like settings.
- To demonstrate basic professional etiquette, ethical awareness, and responsible conduct.
- To develop the skills of structured thinking to simple problems and participate effectively in team-based activities.
- To develop placement-oriented aptitude, workplace expectations, and early career preparation.
- 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: After the successful completion of the course students will able to:

- **CO1** Explain the importance of employability skills in engineering careers;
- **CO2** Present themselves with greater confidence through self-introduction, written communication, and presentations;
- **CO3** Participate constructively in group discussions and team tasks;
- **CO4** Demonstrate basic aptitude readiness and structured problem-solving ability;
- **CO5** Prepare a basic résumé and articulate career goals;
- **CO6** Identify responsible and productive uses of digital and AI tools in learning and work contexts.

Course Contents

Module	Unit	Description	Hours
1.0		**Employability Awareness and Self-Discovery**	**4**
	1.1	Faculty input on employability, workplace expectations, and factors beyond academic performance (attitude, communication, reliability);	
	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**	**4**
	2.1	Input on professional mindset: discipline, accountability, ownership, and workplace ethics;	
	2.2	Case-based learning on missed deadlines, communication gaps, responsibility, and ethical dilemmas;	
	2.3	Role-play exercises demonstrating weak vs effective workplace behaviour;	
	2.4	Reflection on “habits that influence professional credibility” and behaviour correction strategies;	

3.0		**Analytical Thinking, Aptitude, Emotional Intelligence and Decision-Making**	**04**
	3.1	Introduction to critical thinking: distinguishing facts vs assumptions, identifying problems, and structured analysis;	
	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**	**6**
	4.1	Overview of career pathways: core engineering, IT, analytics, higher studies, and entrepreneurship; preparation of personal career planning sheet;	
	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**	**6**
	5.1	Introduction to group discussion norms and evaluation parameters (content, clarity, listening, structure, and collaboration);	
	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**	**6**
	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)**	02
	7.1	Faculty-led discussion on AI in workplace tasks and engineering contexts	
	7.2	Examples of AI-assisted writing, coding, research support	

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.

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

Semester End Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 07 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

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
UHSVE005	AI Ethics and Governance	01	00	01	01	--	01

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					Term Work		Total (Marks)
		T1	T2	FET	Total	Min pass	Marks	Min pass	
UHSVE005	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 the fundamental concepts of ethics, morality, and their relevance in Artificial Intelligence systems.
- To develop analytical skills for examining issues related to bias, fairness, and transparency in AI models and decision-making processes.
- To enable students to understand privacy, data governance, and security challenges in AI-driven systems.
- To familiarize students with AI governance frameworks, policies, and regulatory requirements at national and global levels.
- To examine the societal and legal implications of AI and promote 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. M. Coeckelbergh, AI Ethics, Cambridge, U.K.: MIT Press, 2024.
2. B. Mittelstadt and L. Floridi, The Ethics of Artificial Intelligence, Oxford, U.K.: Oxford University Press, 2023.

References

1. A. Narayanan and S. Kapoor, AI Snake Oil: What Artificial Intelligence Can Do, What It Can't, and How to Tell the Difference, Princeton, NJ, USA: Princeton University Press, 2024.
2. J. Mehan, Artificial Intelligence: Ethical, Social, and Security Impacts for the Present and the Future, 2nd ed., Ely, U.K.: IT Governance Publishing, 2024.
3. V. Dignum, Responsible Artificial Intelligence, Cham, Switzerland: Springer, 2023.

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.

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.

*****