GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
COURSE CATALOGUE
REGULATIONS B.TECH – GR - 24
COMPUTER SCIENCE AND ENGINEERING (AI & ML)
II YEAR II SEMESTER

Course Code	Course Name	Subject Area	Category	Periods Per Week			Credits	Scheme of Examination Max Marks		
				L	T	P		CIA	SEE	Total
THEORY										
AM401PC	Discrete Mathematics	PCC	CORE	3	0	0	3	40	60	100
AM402PC	Automata Theory and Compiler Design	PCC	CORE	3	0	0	3	40	60	100
AM403PC	Operating Systems	PCC	CORE	3	0	0	3	40	60	100
AM404PC	Computer Networks	PCC	CORE	3	0	0	3	40	60	100
AM405PC	Introduction to Artificial Intelligence	PCC	CORE	3	0	0	3	40	60	100
PRACTICAL										
AM406PC	Operating Systems Lab	PCC	CORE	0	0	2	1	40	60	100
AM407PC	Computer Networks Lab	PCC	CORE	0	0	2	1	40	60	100
AM408PC	Real-Time Research Project/Field-Based Research Project	PROJ	PROJECT	0	0	4	2	40	60	100
AM409PC	Prolog/ Lisp/ Pyswip	PCC	CORE	0	0	2	1	40	60	100
MANDATORY COURSE										
*MC410	Intellectual Property Rights	MC - IV	MC	3	0	0	0			
Total Credits				**18**	**0**	**10**	**20**			

COURSE CONTENT

DISCRETE MATHEMATICS								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM401PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Mathematics courses of the first year of study.								

1. COURSE OVERVIEW

This course introduces students to the foundational concepts of **Discrete Mathematics**, which form the backbone of computer science and engineering. It equips students with essential tools to reason mathematically, model computational problems, and design efficient algorithms.

The course begins with **Mathematical Logic**, where students learn formal notation, propositional and predicate calculus, and methods of inference to construct precise proofs. It then covers **Set Theory, Relations, and Functions**, which are widely applied in database theory, automata, and programming languages.

Students also explore **Algebraic Structures** such as semigroups, monoids, lattices, and Boolean algebra, which play a vital role in digital logic design, cryptography, and theoretical computer science. The course further develops problem-solving skills through **Combinatorics and Counting Principles**, including permutations, combinations, binomial and multinomial theorems, and the principle of inclusion-exclusion—key tools for algorithm analysis and complexity.

Finally, students are introduced to **Graph Theory**, covering trees, spanning trees, binary trees, planar graphs, Euler and Hamiltonian circuits, chromatic numbers, and the Four-Colour problem, with applications in networking, data structures, and optimisation problems.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Introduces elementary discrete mathematics for computer science and engineering.
2) Topics include formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles, recurrence relations and generating functions.

3. COURSE OUTCOMES

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

CO 1	Understand and construct precise mathematical proofs
CO 2	Apply logic and set theory to formulate precise statements
CO 3	Analyse and solve counting problems on finite and discrete structures
CO 4	Describe and manipulate sequences
CO 5	Apply graph theory in solving computing problems.

4. COURSE CONTENT

UNIT - I Mathematical logic: 10L

Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT - II Set theory: 8L

Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT - III Algebraic Structures: 10L

Introduction, Algebraic Systems, Semigroups and Monoids, Lattices as Partially Ordered Sets, Boolean Algebra.

UNIT - IV Elementary Combinatorics: 10 L

Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Exclusion.

UNIT - V Graph Theory: 10L

Basic Concepts, Isomorphism and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Colour Problem.

5. TEXT BOOKS

1) Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st Edition.
2) Discrete Mathematics for Computer Scientists & Mathematicians: Joe L. Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd Edition.

6. REFERENCE BOOKS

1) Discrete and Combinatorial Mathematics - an applied introduction: Ralph P. Grimald, Pearson Education, 5th edition.
2) Discrete Mathematical Structures: Thomas Kosy, Tata McGraw-Hill Publishing Co.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**										**2**	**3**	**2**
CO 2	**2**	**3**										**2**	**2**	**3**
CO 3	**2**	**3**											**2**	**3**
CO 4	**3**	**-**	**1**		**2**								**3**	**-**
CO 5	**3**	**3**			**1**							**2**	**3**	**3**

COURSE CONTENT

AUTOMATA THEORY AND COMPILER DESIGN								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM402PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course offers a dual focus on the mathematical foundations of computation and the engineering principles behind compiler construction. It equips students with the theoretical tools to model computation and the practical skills to build language translators.

2. COURSE OBJECTIVE

The students will try to Learn:

1) The students will try to learn: To introduce the fundamental concepts of formal languages, grammars and automata theory.
2) To understand deterministic and non-deterministic machines and the differences between decidability and undecidability.
3) Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
4) Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, intermediate code generation

3. COURSE OUTCOMES

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

CO 1	Able to employ finite state machines for modeling and solving computing problems.
CO 2	Able to design context free grammars for formal languages.
CO 3	Able to distinguish between decidability and undecidability.
CO 4	Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
CO 5	Acquire skills in using lex and design LR parsers.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA

UNIT – II **(10L)**

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages: Statement of the pumping lemma, Applications of the Pumping Lemma.

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmostand Rightmost Derivations, the Language of a Grammar, Parse Trees, Ambiguity in Grammars and Languages.

UNIT – III **(10L)**

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA and CFG's, Acceptance by final state

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines

UNIT – IV **(10L)**

Introduction: The structure of a compiler,

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex,

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers

UNIT – V **(8L)**

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax- Directed Translation Schemes, Implementing L-Attributed SDD's.

Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code

Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management

5. TEXT BOOKS

1) Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2) Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd Edition, PHI.

6. REFERENCE BOOKS

1) Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
2) Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
3) Introduction to Languages and The Theory of Computation, John C Martin, TMH.
4) lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly Compiler Construction, KennethC. Louden, Thomson. Course Technology.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**3**	**2**	**2**	**1**				**2**	**2**	**2**		
CO 2	**3**	**2**	**2**	**2**	**3**	**1**				**2**	**2**	**2**		
CO 3	**2**	**2**	**3**	**3**	**2**	**2**				**2**	**2**	**3**		
CO 4	**3**	**2**	**3**	**3**	**3**	**2**				**2**	**2**	**3**		
CO 5	**3**	**2**	**3**	**3**	**3**	**2**				**2**	**2**	**3**		

COURSE CONTENT

OPERATING SYSTEMS								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM403PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on Computer Programming and Data Structures. 2. A course on Computer Organization and Architecture.								

1. COURSE OVERVIEW

This course introduces the fundamental principles of **Operating Systems (OS)**, which act as an intermediary between hardware and users. It covers concepts such as **processes, CPU scheduling, synchronization, deadlocks, memory management, file systems, and system calls**, with emphasis on both theoretical foundations and practical aspects of Unix/Linux.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand **operating system concepts** including processes, threads, scheduling, synchronization, deadlocks, memory management, and file systems.
2) To explore the **design and development issues** of modern operating systems.
3) To learn and practice **Unix commands** and system call interfaces for process, communication, and I/O management.

3. COURSE OUTCOMES

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

CO 1	Control access to a computer and files, ensuring secure sharing and protection.
CO 2	Demonstrate knowledge of computer components and their respective roles in computing.
CO 3	Recognize and resolve user problems within standard operating environments.
CO 4	Gain practical knowledge of how programming languages, operating systems, and computer architecture interact effectively.
CO 5	Apply expert system development tools and frameworks to design domain-specific intelligent applications.

4. COURSE CONTENT

UNIT - I **(10L)**

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls.

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads.

UNIT - II **(10L)**

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec.

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT – III (10L)

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT – IV (9L)

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT – V (9L)

File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

5. TEXT BOOKS

1) Abraham Silberchatz, Peter B. Galvin, Greg Gagne, *Operating System Principles*, 7th Edition, John Wiley.
2) W.R. Stevens, *Advanced Programming in the UNIX Environment*, Pearson Education.

6. REFERENCE BOOKS

1) William Stallings, *Operating Systems: Internals and Design Principles*, 5th Edition, Pearson/PHI.
2) Crowley, *Operating System: A Design Approach*, TMH.
3) Andrew S. Tanenbaum, *Modern Operating Systems*, 2nd Edition, Pearson/PHI.
4) Kernighan and Pike, *UNIX Programming Environment*, PHI/Pearson Education.
5) U. Vahalia, *UNIX Internals: The New Frontiers*, Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**3**	**2**	**2**	**1**	**2**	**1**	**1**	**2**	**2**	**2**		
CO 2	**3**	**2**	**2**	**2**	**3**	**1**	**1**	**1**	**1**	**2**	**2**	**2**		
CO 3	**2**	**2**	**3**	**3**	**2**	**2**	**2**	**1**	**1**	**2**	**2**	**3**		
CO 4	**3**	**2**	**3**	**3**	**3**	**2**	**2**	**1**	**1**	**2**	**2**	**3**		
CO 5	**3**	**2**	**3**	**3**	**3**	**2**	**2**	**1**	**2**	**2**	**2**	**3**		

COURSE CONTENT

COMPUTER NETWORKS								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM404PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on *Programming for Problem Solving* 2. A course on *Data Structures*								

1. COURSE OVERVIEW

The **Computer Networks** course introduces fundamental concepts of networking, layered architectures, and protocols used in modern communication systems. Students will gain an in-depth understanding of the OSI and TCP/IP reference models, network topologies, subnetting, routing mechanisms, and essential protocols for reliable communication. Emphasis is placed on both theoretical knowledge and practical application of networking concepts in real-world scenarios.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the concepts and fundamentals of computer networks.
2) To familiarize with layered models (OSI and TCP/IP) for communication between machines.
3) To learn the functionalities of each network layer and its protocols.
4) To gain knowledge of routing, subnetting, congestion control, and quality of service.
5) To explore application-level protocols for web and multimedia communication.

3. COURSE OUTCOMES

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

CO 1	Explain the fundamental concepts of computer networks and communication models.
CO 2	Describe the functions of each layer in the OSI and TCP/IP reference models.
CO 3	Apply subnetting and routing techniques for efficient network design.
CO 4	Analyze various protocols of data link, network, transport, and application layers.
CO 5	Demonstrate the use of network protocols in real-world applications such as DNS, HTTP, email, and streaming.

4. COURSE CONTENT

UNIT – I (10L)

Introduction: Network hardware, Network software, OSI and TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media – twisted pairs, coaxial cable, fiber optics; Wireless Transmission.

Data Link Layer: Design issues, framing, Error detection and correction.

UNIT – II **(10L)**

Elementary Data Link Protocols: Simplex protocol, Stop-and-wait protocol for error-free and noisy channels.
Sliding Window Protocols: One-bit sliding window, Go-Back-N, Selective Repeat, Example data link protocols.
Medium Access Sublayer: Channel allocation problem, Multiple access protocols (ALOHA, CSMA, collision-free protocols), Wireless LANs, Data link layer switching.

UNIT – III **(10L)**

Network Layer: Design issues, Routing algorithms (shortest path, flooding, hierarchical, broadcast, multicast, distance vector).
Congestion Control Algorithms, Quality of Service (QoS), Internetworking, Network Layer in the Internet.

UNIT – IV **(9L)**

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT – V **(9L)**

Application Layer: Domain Name System (DNS), SNMP, Electronic Mail; The World Wide Web (HTTP), Streaming audio and video.

5. TEXT BOOKS

1) *Computer Networks* – Andrew S. Tanenbaum, David J. Wetherall, 5th Edition, Pearson Education/PHI.

6. REFERENCE BOOKS

1) *An Engineering Approach to Computer Networks* – S. Keshav, 2nd Edition, Pearson Education.
2) *Data Communications and Networking* – Behrouz A. Forouzan, 3rd Edition, TMH.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**1**	**2**	**-**	**-**	**-**	**1**	**2**	**-**	**2**		
CO 2	**3**	**2**	**2**	**1**	**2**	**-**	**-**	**-**	**1**	**2**	**-**	**2**		
CO 3	**3**	**3**	**3**	**2**	**3**	**-**	**-**	**-**	**2**	**2**	**-**	**3**		
CO 4	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**2**	**3**	**-**	**3**		
CO 5	**2**	**2**	**2**	**2**	**3**	**-**	**-**	**-**	**2**	**3**	**-**	**3**		

COURSE CONTENT

<table>
<tr><th colspan="9">INTRODUCTION TO ARTIFICIAL INTELLIGENCE</th></tr>
<tr><td colspan="9"><b>II Year - II Semester: CSE (AI &amp; ML)</b></td></tr>
<tr><td><b>Course Code</b></td><td><b>Category</b></td><td colspan="3"><b>Hours/Week</b></td><td><b>Credits</b></td><td colspan="3"><b>Maximum Marks</b></td></tr>
<tr><td rowspan="2"><b>AM405PC</b></td><td rowspan="2"><b>Core</b></td><td><b>L</b></td><td><b>T</b></td><td><b>P</b></td><td><b>C</b></td><td><b>CIA</b></td><td><b>SEE</b></td><td><b>Total</b></td></tr>
<tr><td><b>3</b></td><td><b>-</b></td><td><b>-</b></td><td><b>3</b></td><td><b>40</b></td><td><b>60</b></td><td><b>100</b></td></tr>
<tr><td>Contact Classes: 48</td><td>Tutorial Classes: Nil</td><td colspan="4">Practical Classes: Nil</td><td colspan="3"><b>Total Classes: 48</b></td></tr>
<tr><td colspan="9"><b>Prerequisite:</b> Nil</td></tr>
</table>

1. COURSE OVERVIEW

This course introduces the key principles and diverse concepts in Artificial Intelligence (AI). It covers the philosophy of AI, modelling problems as AI tasks, and solving them using approaches such as search algorithms (breadth-first, depth-first, hill climbing), logic (propositional and first-order), probabilistic models like Bayesian network. Students learn foundational algorithms and AI paradigms, including classical planning, game playing, constraint satisfaction, and reasoning under uncertainty.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To learn the distinction between optimal reasoning Vs. human like reasoning.
2) To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
3) To learn different knowledge representation techniques.
4) To understand the applications of AI, namely game playing, theorem proving, and machine learning.

3. COURSE OUTCOMES

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

CO 1	Select a search algorithm for a problem and estimate its time and space complexities.
CO 2	Apply AI techniques to solve problems of game playing, theorem proving, and machine learning.
CO 3	Learn different knowledge representation techniques.
CO 4	Understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
CO 5	Comprehend the applications of Probabilistic Reasoning and Bayesian Networks.

4. COURSE CONTENT

UNIT - I

Introduction to AI - Intelligent Agents, Problem-Solving Agents, **(10L)**

Searching for Solutions - Breadth-first search, Depth-first search, Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces.

UNIT-II **(10L)**

Games - Optimal Decisions in Games, Alpha–Beta Pruning, Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Knowledge-Based Agents, **Logic-** Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses.

UNIT-III (10L)

First-Order Logic - Syntax and Semantics of First-Order Logic, Using First Order Logic, Knowledge Engineering in First-Order Logic. Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification, Forward Chaining, Backward Chaining, Resolution.

Knowledge Representation: Ontological Engineering, Categories and Objects, Events.

UNIT-IV (10L)

Planning - Definition of Classical Planning, Algorithms for Planning with State Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches. Hierarchical Planning.

UNIT-V (8L)

Probabilistic Reasoning:

Acting under Uncertainty, Basic Probability Notation Bayes' Rule and Its Use, Probabilistic Reasoning, Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First- Order Probability.

5. TEXT BOOKS

1) Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

6. REFERENCE BOOKS

1) Artificial Intelligence, 3rd Edn., E. Rich and K. Knight (TMH).
2) Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.
3) Artificial Intelligence, Shivani Goel, Pearson Education.
4) Artificial Intelligence and Expert systems – Patterson, Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**1**	**2**				**1**	**2**	**3**	**2**		
CO 2	**3**	**2**	**2**	**1**	**2**				**1**	**2**	**2**	**2**		
CO 3	**3**	**3**	**3**	**2**	**3**				**2**	**2**	**2**	**3**		
CO 4	**3**	**3**	**3**	**3**	**3**				**2**	**3**	**2**	**3**		
CO 5	**2**	**2**	**2**	**2**	**3**				**2**	**3**	**2**	**3**		

COURSE CONTENT

OPERATING SYSTEMS LAB								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM406PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course provides practical exposure to the core concepts of operating systems through programming and simulation. Students will gain hands-on experience in implementing process scheduling, deadlock handling, memory management, and inter-process communication. The lab also emphasizes the use of UNIX/LINUX system calls for process and file management.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To provide an understanding of the design aspects of operating system concepts through simulation.
2) To introduce UNIX commands and system call interfaces for process, memory, and file management.

3. COURSE OUTCOMES

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

CO 1	Simulate and implement CPU scheduling algorithms.
CO 2	Apply UNIX/LINUX system calls for file operations and process management.
CO 3	Demonstrate deadlock avoidance, prevention, and recovery mechanisms.
CO 4	Implement inter-process communication using pipes, FIFOs, message queues, and shared memory.
CO 5	Simulate memory management and page replacement algorithms.

4. LIST OF EXPERIMENTS

1) Write C programs to simulate the following CPU Scheduling algorithms:
   a) FCFS b) SJF c) Round Robin d) Priority Scheduling

2) Write programs using the I/O system calls of UNIX/LINUX operating system:
   open, read, write, close, fcntl, seek, stat, opendir, readdir.

3) Write a C program to simulate Banker’s Algorithm for Deadlock Avoidance and Prevention.

4) Write a C program to implement the Producer–Consumer problem using semaphores with UNIX/LINUX system calls.

5) Write C programs to illustrate IPC mechanisms
   a) Pipes b) FIFOs c) Message Queues d) Shared Memory

6) Write C programs to simulate the following memory management techniques
   a) Paging b) Segmentation

7) Write C programs to simulate Page Replacement policies:
   a) FCFS b) LRU c) Optimal

5. TEXT BOOKS

1) Operating System Principles – Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley.
2) Advanced Programming in the UNIX Environment – W.R. Stevens, Pearson Education.

6. REFERENCE BOOKS

1) Operating Systems: Internals and Design Principles – William Stallings, 5th Edition, Pearson Education/PHI.
2) Operating System: A Design Approach – Crowley, TMH.
3) Modern Operating Systems – Andrew S. Tanenbaum, 2nd Edition, Pearson/PHI.
4) UNIX Programming Environment – Kernighan and Pike, PHI/Pearson Education.
5) UNIX Internals: The New Frontiers – U. Vahalia, Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**3**		**2**					**2**		**3**		
CO 2	**3**	**3**	**2**		**2**					**2**		**2**		
CO 3	**3**	**2**	**3**		**2**					**2**		**3**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**		**2**		
CO 5	**3**	**2**	**3**	**2**	**2**					**2**		**3**		

COURSE CONTENT

COMPUTER NETWORKS LAB								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM407PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course provides practical exposure to computer networks and communication protocols. Students will learn to implement framing, error detection, routing, congestion control, and encryption techniques. They will also gain hands-on experience with Wireshark, Nmap, and NS2 simulators for traffic analysis and performance evaluation of network topologies.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the working principle of various communication protocols.
2) To understand the network simulator environment and visualize a network topology.
3) To analyze traffic flow and the contents of protocol frames.

3. COURSE OUTCOMES

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

CO 1	Implement data link layer framing methods.
CO 2	Analyze error detection and error correction codes.
CO 3	Implement and analyze routing and congestion issues in network design.
CO 4	Implement encoding and decoding techniques used in the presentation layer.
CO 5	Work with different network tools for packet analysis and network simulation.

4. LIST OF EXPERIMENTS

1) Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2) Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC-CCIP.
3) Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4) Implement Dijkstra’s algorithm to compute the shortest path through a network.
5) Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6) Implement distance vector routing algorithm for obtaining routing tables at each node.
7) Implement data encryption and data decryption.
8) Write a program for congestion control using Leaky Bucket algorithm.
9) Write a program for frame sorting techniques used in buffers.

10) Wireshark:

    i. Packet Capture using Wireshark

    ii. Starting Wireshark

    iii. Viewing Captured Traffic

    iv. Analysis, Statistics & Filters

    v. Running Nmap Scan

    vi. Operating System Detection using Nmap

11) NS2 Simulator:

    i. Introduction to NS2 Simulator

    ii. Simulate to find the number of packets dropped

    iii. Simulate to find the number of packets dropped by TCP/UDP

    iv. Simulate to find the number of packets dropped due to congestion

    v. Simulate to compare data rate & throughput

    vi. Simulate to plot congestion for different source/destination

    vii. Simulate to determine the performance with respect to transmission of packets

5. TEXT BOOKS

1) *Computer Networks*, Andrew S. Tanenbaum, David J. Wetherall, 5th Edition, Pearson Education/PHI.

6. REFERENCE BOOKS

1) *An Engineering Approach to Computer Networks*, S. Keshav, 2nd Edition, Pearson Education.
2) *Data Communications and Networking*, Behrouz A. Forouzan, 3rd Edition, TMH.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**		**2**					**1**		**2**		
CO 2	**3**	**3**	**2**		**2**					**1**		**2**		
CO 3	**3**	**2**	**3**	**2**	**2**					**2**		**2**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**		**2**		
CO 5	**3**	**2**	**2**	**3**	**3**					**2**		**3**		

COURSE CONTENT

PROLOG/LISP/PYSWIP								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM409PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course introduces students to the foundational paradigms of logic programming and functional programming, emphasizing their applications in artificial intelligence. Learners will explore Prolog for declarative problem-solving, LISP for symbolic computation.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Write and debug Prolog programs using facts, rules, and queries.
2) Apply LISP's functional programming techniques to AI problems.
3) Integrate Prolog logic into Python applications using PySwip.
4) Solve real-world problems using declarative and symbolic approaches.

3. COURSE OUTCOMES

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

CO 1	Develop and debug Prolog programs using facts, rules, and recursive queries.
CO 2	Apply logical inference and backtracking to solve complex problems.
CO 3	Solve Real time problems
CO 4	Analyze the prolog predicate
CO 5	Understand the fundamentals of logic programming and declarative paradigms.

4. LIST OF EXPERIMENTS

1) Write simple fact for following:
   A. Ram likes mango.
   B. Seema is a girl.
   C. Bill likes Cindy.
   D. Rose is red.
   E. John owns gold
2) Write predicates one converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.
3) Write a program to solve the Monkey Banana problem
4) WAP in turbo prolog for medical diagnosis and show the advantages and disadvantages of green and red cuts.
5) Write a program to solve the 4-Queen problem.
6) Write a program to solve traveling salesman problems.
7) Write a program to solve water jug problems using Prolog.

8) Write simple Prolog functions such as the following. Take into account lists which are too short.
-- remove the Nth item from the list. -- insert as the Nth item.

9) Assume the prolog predicate gt(A, B) is true when A is greater than B. Use this predicate to define the predicate addLeaf(Tree, X, NewTree) which is true if NewTree is the Tree produced by adding the item X in a leaf node. Tree and NewTree are binary search trees. The empty tree is represented by the atom nil.

10) Write a Prolog predicate, countLists(Alist, Ne, Nl), using accumulators, that is true when Nl is the number of items that are listed at the top level of Alist and Ne is the number of empty lists. Suggestion: First try to count the lists, or empty lists, then modify by adding the other counter.

11) Define a predicate memCount(AList,Blist,Count) that is true if Alist occurs Count times within Blist. Define without using an accumulator. Use "not" as defined in utilities.pro, to make similar cases are unique, or else you may get more than one count as an answer.

Examples:

```
memCount(a,[b,a],N).
N = 1 ;
No
 memCount(a,[b,[a,a,[a],c],a],N).
N = 4 ;
no
memCount([a],[b,[a,a,[a],c],a],N).
N = 1 ;
No
```

5. REFERENCE BOOKS

1) PROLOG: Programming for Artificial Intelligence, 3e, by BRATKO, WILEY

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**		**2**					**2**	**2**	**2**		
CO 2	**3**	**3**	**2**		**2**					**2**	**2**	**2**		
CO 3	**3**	**2**	**3**	**2**	**2**					**2**	**2**	**2**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**	**2**	**2**		
CO 5	**3**	**2**	**2**	**3**	**3**					**2**	**2**	**2**		

COURSE CONTENT

INTELLECTUAL PROPERTY RIGHTS								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
***MC410**	**Mandatory**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**-**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite:								

1. COURSE OVERVIEW

The Intellectual Property Rights (IPR) course is designed to provide students with a comprehensive understanding of the legal frameworks that protect creative and innovative works. It covers the principles, laws, and practices surrounding the protection of intellectual property such as: Patents, Trademarks, Copyrights, Designs and Trade Secrets.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Significance of intellectual property and its protection.
2) Introduce various forms of intellectual property.

3. COURSE OUTCOMES

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

CO 1	Distinguish and Explain various forms of IPRs.
CO 2	Identify criteria to fit one's own intellectual work in particular form of IPRs.
CO 3	Apply statutory provisions to protect particular form of IPRs.
CO 4	Appraise new developments in IPR laws at national and international level
CO 5	Understand the new development of intellectual property.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to Intellectual property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT – II **(10L)**

Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

UNIT – III **(10L)**

Law of copyrights: Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, international copyright law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV (9L)

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriations of trade secrets, protection for submission, trade secret litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V (9L)

New development of intellectual property: new developments in trade mark law; copyright law, patent law, intellectual property audits.

International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

5. TEXT BOOKS

1) Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

6. REFERENCE BOOKS

1) Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**		**2**					**2**	**2**	**2**		
CO 2	**3**	**3**	**2**		**2**					**2**	**2**	**2**		
CO 3	**3**	**2**	**3**	**2**	**2**					**2**	**2**	**2**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**	**2**	**2**		
CO 5	**3**	**2**	**2**	**3**	**3**					**2**	**2**	**2**		