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

Course Code	Course Name	Subject Area	Category	Periods Per Week			Credits	Scheme of Examination Max Marks		
				L	T	P		CIA	SEE	Total
THEORY										
AM301PC	Mathematical and Statistical Foundations	PCC	CORE	3	1	0	4	40	60	100
AM302PC	Database Management Systems	PCC	CORE	3	0	0	3	40	60	100
AM303PC	Computer Organization and Architecture	PCC	CORE	3	0	0	3	40	60	100
AM304PC	Software Engineering	PCC	CORE	3	0	0	3	40	60	100
AM305PC	Object Oriented Programming through Java	PCC	CORE	3	0	0	3	40	60	100
PRACTICAL										
AM306PC	Database Management Systems Lab	PCC	CORE	0	0	2	1	40	60	100
AM307PC	Java Programming Lab	PCC	CORE	0	0	2	1	40	60	100
AM308PC	Node JS/ React JS/ Django	PCC	CORE	0	0	2	1	40	60	100
AM310PC	Software Engineering Lab	PCC	CORE	0	0	2	1	40	60	100
MANDATORY COURSE										
*MC309	Gender Sensitization Laboratory	MC-III	MC	0	0	2	0			
			Total Credits	**15**	**1**	**10**	**20**			

COURSE CONTENT

MATHEMATICAL AND STATISTICAL FOUNDATIONS								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM301PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**1**	**-**	**4**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: 16	Practical Classes: Nil				**Total Classes: 64**		
Prerequisite: Mathematics courses of the first year of study.								

1. COURSE OVERVIEW

This course provides CSE (AI & ML) students with a strong mathematical and statistical foundation required for advanced computing and machine learning applications. It begins with **number theory concepts** such as divisibility, Euclidean algorithms, prime factorisation, and congruences, which are essential in cryptography and security.

Students are then introduced to **linear regression and correlation**, enabling them to model relationships between variables and apply inferential techniques in predictive analytics. The course also covers **probability theory and distributions**—both discrete (Binomial, Poisson) and continuous (Normal distribution)—which are fundamental in data science, AI models, and uncertainty handling.

The concepts of **sampling distributions, estimation, and hypothesis testing** are explored to develop statistical inference skills, essential for analysing experimental data, validating models, and decision-making in machine learning.

In the final unit, students study **Stochastic Processes and Markov Chains**, which provide powerful mathematical tools for analysing dynamic systems and form the backbone of algorithms in reinforcement learning, natural language processing, queueing models, and computer networks.

2. COURSE OBJECTIVE

The students will try to Learn:

1) The Number Theory basic concepts useful for cryptography, etc.
2) The theory of Probability, and probability distributions of single and multiple random variables.
3) The sampling theory and testing of hypotheses and making inferences.
4) Stochastic processes and Markov chains.

3. COURSE OUTCOMES

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

CO 1	Apply the number theory concepts to the cryptography domain.
CO 2	Apply the concepts of probability and distributions to some case studies.
CO 3	Correlate the material of one unit to the material in other units.
CO 4	Resolve the potential misconceptions and hazards in each topic of study.
CO 5	Stochastic Process, Markov Process & Markov Chains.

4. COURSE CONTENT

UNIT – I: Greatest Common Divisors and Prime Factorisation 10 L

Greatest common divisors, the Euclidean algorithm, the fundamental theorem of arithmetic, Factorisation of integers and the Fermat numbers. Congruences: Introduction to congruences, Linear congruences, The Chinese remainder theorem, Systems of linear congruences.

UNIT - II Simple Linear Regression and Correlation 10L

Introduction to Linear Regression, The Simple Linear Regression Model, Least Squares and the Fitted Model, Properties of the Least Squares Estimators, Inferences Concerning the Regression Coefficients, Prediction, Simple Linear Regression Case Study.

Random Variables and Probability Distributions: Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions, Statistical Independence. Discrete Probability Distributions: Binomial Distribution, Poisson Distribution.

UNIT – III: Continuous Probability Distributions 10L

Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial, Fundamental Sampling Distributions: Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem, Sampling Distribution of S2, t-Distribution, F-Distribution.

UNIT – IV: Estimation & Tests of Hypotheses 10L

Introduction, Statistical Inference, Classical Methods of Estimation. Estimating the Mean, Standard Error of a Point Estimate, Prediction Intervals, Tolerance Limits, Estimating the Variance, Estimating a Proportion for a single mean, Difference between Two Means, between Two Proportions for Two Samples and Maximum Likelihood Estimation.

UNIT - V: Stochastic Processes and Markov Chains

Introduction to Stochastic Processes- Markov Process. Transition Probability, Transition Probability Matrix, First order and Higher order Markov process, n-step transition probabilities, Markov chain, Steady state condition, Markov analysis.

5. TEXT BOOKS

1) Kenneth H. Rosen, Elementary number theory & its applications, sixth edition, Addison-Wesley, ISBN 978 0-321-50031-1.
2) Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.
3) S.D.Sharma, Operations Research, Kedarnath and Ramnath Publishers, Meerut, Delhi.

6. REFERENCE BOOKS

1) S.C. Gupta & V.K. Kapoor, Fundamentals of Mathematical statistics, Khanna publications.
2) T.T. Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons. Ltd, 2004.
3) Sheldon M Ross, Probability and statistics for Engineers and scientists, Academic Press.

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

DATABASE MANAGEMENT SYSTEMS								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM302PC	**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

The "Database Management Systems" course introduces students to the principles, design, and implementation of database systems. It covers data models, ER modeling, relational algebra, SQL, normalization, transaction management, concurrency control, recovery, and indexing structures. Students will develop both theoretical foundations and practical skills to design, query, and manage databases. By the end of the course, students will be equipped to apply DBMS concepts in real-world applications requiring efficient and reliable data management.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the basic concepts, architecture, and applications of database systems.
2) To master SQL for data definition, manipulation, and control.
3) To apply relational algebra and design normalized database schemas.
4) To understand transaction management, concurrency control, and recovery.
5) To explore indexing and hashing structures for optimized query performance.

3. COURSE OUTCOMES

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

CO 1	Understand database concepts, ER modeling, and convert ER diagrams into relational schemas.
CO 2	Apply relational algebra and SQL to define, manipulate, and query data.
CO 3	Normalize relational schemas using appropriate normal forms to reduce redundancy.
CO 4	Demonstrate understanding of transaction concepts, concurrency control, and recovery techniques.
CO 5	Implement indexing and hashing techniques to improve query processing and data retrieval.

4. COURSE CONTENT

UNIT – I **(10L)**

Database System Applications: A Historical Perspective, File Systems vs DBMS, Data Models, Levels of Abstraction, Data Independence, Structure of a DBMS.

Introduction to Database Design: ER Diagrams, Entities, Attributes, Entity Sets, Relationships and Relationship Sets, Additional ER Features, Conceptual Design using ER Model.

UNIT – II **(10L)**

Relational Model: Integrity Constraints, Enforcing Constraints, Querying Relational Data, Logical Database Design, Views, Altering/Destroying Tables and Views.

Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus.

UNIT – III **(10L)**

SQL: Basic SQL Queries, UNION, INTERSECT, EXCEPT, Nested Queries, Aggregates, NULLs, Integrity Constraints, Triggers, Active Databases.

Schema Refinement: Problems due to redundancy, Decompositions, Functional Dependencies, Normalization (1NF, 2NF, 3NF, BCNF, 4NF, 5NF).

UNIT – IV **(9L)**

Transactions: Transaction Concept, States, Atomicity & Durability.

Concurrency Control: Serializability, Recoverability, Isolation, Lock-based Protocols, Timestamp-based Protocols, Validation-based Protocols, Multiple Granularity.

Recovery: Log-based Recovery, Recovery with Concurrent Transactions.

UNIT – V **(9L)**

Storage & Indexing: Data on External Storage, File Organization, Primary/Secondary/Cluster Indexes. Index Data Structures: Hash-based Indexing, Tree-based Indexing, ISAM, B+ Trees, Comparison of File Organizations.

5. TEXT BOOKS

1) Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, 6th Edition, McGraw Hill.
2) Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems*, 7th Edition, Pearson.

6. REFERENCE BOOKS

1) Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems*, Tata McGraw Hill.
2) C.J. Date, *An Introduction to Database Systems*, Pearson Education.
3) Peter Rob, Carlos Coronel, *Database Systems: Design, Implementation, and Management*, 7th Edition, Cengage.
4) Shah & Shah, *Oracle for Professionals*, SPD/PHI.
5) M.L. Gillenson, *Fundamentals of Database Management Systems*, Wiley Student Edition.

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**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 2	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 3	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 4	**3**	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 5	**3**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		

COURSE CONTENT

COMPUTER ORGANIZATION AND ARCHITECTURE								
II Year - II Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM303PC	**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 provides a comprehensive introduction to the fundamental principles of computer organization and architecture. It covers the internal structure and operational behavior of digital computers, focusing on the interaction between hardware and software at a low level. Topics include data representation, instruction set architecture (ISA), processor design, memory hierarchy, input/output systems, and performance evaluation. Students will gain an understanding of how computers execute programs, how hardware supports software, and how performance is impacted by architectural choices.

2. COURSE OBJECTIVE

The students will try to Learn:

1) The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
2) It begins with basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
3) Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.

3. COURSE OUTCOMES

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

CO 1	Demonstrate an understanding of the design of the functional units of a digital computer system.
CO 2	Design of Control unit and Central processing unit.
CO 3	Recognize and manipulate representations of numbers stored in digital computers
CO 4	Understand Input - Output Organization and Memory Organization.
CO 5	Design a pipeline for consistent execution of instructions with minimum hazards.

4. COURSE CONTENT

UNIT - I **(10L)**

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT - II **(10L)**

Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT - III **(10L)**

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT - IV **(9L)**

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT - V **(9L)**

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

5. TEXT BOOKS

1) Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.

6. REFERENCE BOOKS

1) Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition, McGraw Hill.
2) Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
3) Structured Computer Organization – Andrew S. Tanenbaum, 4 th Edition, PHI/Pearson.

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**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 2	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 3	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 4	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 5	**3**	**3**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		

COURSE CONTENT

SOFTWARE ENGINEERING								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM304PC	**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

The **Software Engineering** course provides students with the principles, practices, and methodologies used in the development of reliable, scalable, and maintainable software systems. Students will learn software process models, requirements engineering, software design concepts, UML modeling, testing strategies, risk management, and quality assurance. Emphasis is given to applying theoretical concepts to real-world software development projects, ensuring exposure to both classical and agile approaches to software engineering.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To gain working knowledge of techniques for estimation, design, testing, and quality management of large-scale software projects.
2) To understand various process models and the role of process frameworks such as CMMI.
3) To acquire skills in requirements engineering and preparation of Software Requirement Documents (SRDs).
4) To design and model systems using UML diagrams and software architecture patterns.
5) To understand software testing methodologies, software metrics, risk management, and quality assurance practices.

3. COURSE OUTCOMES

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

CO 1	Translate end-user requirements into structured system and software requirements using UML, and prepare a Software Requirement Document (SRD).
CO 2	Apply suitable software architectures and patterns to design systems, and evaluate alternative design strategies.
CO 3	Develop UML diagrams such as class diagrams, sequence diagrams, and use case diagrams for effective system modelling.
CO 4	Demonstrate knowledge of software testing strategies, apply black-box and white-box techniques, and prepare a testing report.
CO 5	Analyze and apply risk management strategies and software quality assurance practices for reliable software development.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic View of Process: Software engineering as layered technology, process framework, capability maturity model integration (CMMI).

Process Models: The Waterfall model, Spiral model, Agile methodology.

UNIT – II **(10L)**

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, software requirements document.

Requirements Engineering Process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT – III **(10L)**

Design Engineering: Design process and quality, design concepts, design model.

Architectural Design: Software architecture, data design, architectural styles and patterns.

UML Modeling: Conceptual model of UML, structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT – IV **(9L)**

Testing Strategies: A strategic approach to software testing, conventional software test strategies, black-box and white-box testing, validation testing, system testing, debugging techniques.

Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT – V **(9L)**

Risk Management: Reactive vs. proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM.

Quality Management: Quality concepts, software quality assurance (SQA), reviews, formal technical reviews, statistical SQA, software reliability, ISO 9000 standards.

5. TEXT BOOKS

1) *Software Engineering: A Practitioner's Approach* – Roger S. Pressman, 6th Edition, McGraw Hill.
2) *Software Engineering* – Ian Sommerville, 7th Edition, Pearson Education.

6. REFERENCE BOOKS

1) *The Unified Modeling Language User Guide* – Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
2) *Software Engineering: An Engineering Approach* – James F. Peters, Witold Pedrycz, John Wiley.
3) *Software Engineering: Principles and Practice* – Waman S. Jawadekar, McGraw-Hill.
4) *Fundamentals of Object-Oriented Design using UML* – Meilir Page-Jones, 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**					**2**	**3**	**2**		
CO 2	**3**	**2**	**2**	**1**	**2**					**2**	**3**	**2**		
CO 3	**2**	**3**	**3**	**2**	**2**				**1**	**2**	**3**	**3**		
CO 4	**3**	**3**	**3**	**2**	**2**				**1**	**2**	**3**	**3**		
CO 5	**2**	**2**	**3**	**3**	**3**				**1**	**3**	**3**	**3**		

COURSE CONTENT

OBJECT ORIENTED PROGRAMMING THROUGH JAVA								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM305PC	**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 introduces the fundamentals of **Object-Oriented Programming (OOP)** using Java. It emphasizes key OOP principles such as abstraction, encapsulation, inheritance, and polymorphism while providing practical exposure to Java constructs. Students will learn exception handling, multithreading, event-driven programming, and GUI development using AWT and Swing. The course also covers database connectivity (JDBC), enabling students to develop robust and scalable Java applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the principles of OOP and apply them in solving real-world problems.
2) To illustrate and apply inheritance for program reusability.
3) To implement multitasking using multithreading and event handling.
4) To develop database-driven applications using JDBC.
5) To design console-based and GUI-based applications using Java.

3. COURSE OUTCOMES

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

CO 1	Demonstrate program behavior using control structures, constructors, string handling, and garbage collection.
CO 2	Implement inheritance (multilevel, hierarchical, multiple) using extends and implements.
CO 3	Develop applications with multithreading concepts for inter-process communication.
CO 4	Design graphical user interfaces using AWT and Swing.
CO 5	Build applets and deploy them in client-server environments.

4. COURSE CONTENT

UNIT – I **(10L)**

Object-oriented thinking and Java basics: Need for OOP paradigm, abstraction, agents, responsibility, messages, methods. History and buzzwords of Java. Data types, variables, scope, arrays, operators, expressions, control statements, type conversion and casting. Classes, objects, constructors, methods, access control, this keyword, garbage collection. Overloading, method binding, inheritance, overriding, exceptions, recursion, nested and inner classes. String handling.

UNIT – II **(10L)**

Inheritance, Packages, and Interfaces: Hierarchical abstractions, base/subclass concepts, substitutability, specialization, extension, and combination. Polymorphism, abstract classes, the Object class. Packages: defining, creating, accessing, CLASSPATH, importing. Interfaces: defining, implementing, extending. Exploring java.io.

UNIT – III **(10L)**

Exception Handling and Multithreading: Exception hierarchy, try-catch-throw-throws-finally, built-in and user-defined exceptions. String handling. Exploring java.util. Multithreading vs multitasking, thread lifecycle, priorities, synchronization, inter-thread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT – IV **(9L)**

Event Handling: Events, sources, classes, listeners, delegation model. Mouse and keyboard event handling, adapter classes. AWT hierarchy, UI components: labels, buttons, canvas, scrollbars, text components, checkboxes, choice lists, panels, scrollpane, dialogs, menubar, graphics. Layout managers: border, grid, flow, card, grid bag.

UNIT – V **(9L)**

Applets and Swing: Applet concepts, life cycle, types, creating applets, passing parameters. Swing overview: MVC architecture, advantages over AWT. Swing components: JApplet, JFrame, JComponent, icons, labels, text fields, buttons, checkboxes, radio buttons, combo boxes, tabbed panes, scroll panes, trees, tables.

5. TEXT BOOKS

1) Herbert Schildt, *Java: The Complete Reference*, 7th Edition, TMH.
2) Timothy Budd, *Understanding OOP with Java*, Pearson Education.

6. REFERENCE BOOKS

1) J. Nino, F.A. Hosch, *An Introduction to Programming and OO Design using Java*, Wiley.
2) T. Budd, *An Introduction to OOP*, 3rd Edition, Pearson Education.
3) Y. Daniel Liang, *Introduction to Java Programming*, Pearson Education.
4) R.A. Johnson-Thomson, *Introduction to Java Programming and OOP Development*.
5) Cay S. Horstmann, Gary Cornell, *Core Java 2, Vol I – Fundamentals*, 8th Edition, Pearson Education.
6) Cay S. Horstmann, Gary Cornell, *Core Java 2, Vol II – Advanced Features*, 8th Edition, Pearson Education.
7) R. Buyya, S.T. Selvi, X. Chu, *Object-Oriented Programming with Java*, TMH.
8) John Hunt, *Java and Object Orientation: An Introduction*, 2nd Edition, Springer.
9) Maurach, *Beginning Java2 JDK 5*, SPD.

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**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 2	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 3	**2**	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 4	**2**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		
CO 5	**2**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		

COURSE CONTENT

DATA MANAGEMENT SYSTEMS LAB								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM306PC	**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

The "Database Management Systems Lab" course provides hands-on experience in the design and implementation of databases. Students will practice ER modeling, relational schema design, normalization, and SQL programming to manage and query data. The course also focuses on database programming using procedures, triggers, and cursors, enabling students to develop complete database applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To design conceptual models using ER diagrams and convert them into relational schemas.
2) To implement normalization techniques to reduce redundancy and improve database design.
3) To gain practical skills in SQL for data definition, manipulation, and querying.
4) To apply advanced SQL features like triggers, procedures, and cursors for application development.
5) To strengthen database application development skills through practical problem-solving.

3. COURSE OUTCOMES

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

CO 1	Design database schemas using ER modeling and normalization.
CO 2	Apply SQL commands for data definition, manipulation, and constraints.
CO 3	Use advanced SQL queries including joins, nested queries, and aggregate functions.
CO 4	Implement database triggers, procedures, and cursors for application development.
CO 5	Develop complete database applications integrating SQL features to solve real-world problems.

4. COURSE CONTENT

Practice Sessions / Experiments

1) Concept design with E-R Model.
2) Relational Model implementation.
3) Normalization up to 3NF / BCNF.
4) Practicing DDL commands.
5) Practicing DML commands.
6) **Queries:**
   (i) A. Using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.
   (ii) B. Nested & Correlated subqueries.

7) Queries using Aggregate functions, GROUP BY, HAVING; Creating and Dropping Views.

8) Triggers: Insert, Delete, Update Triggers.

9) Procedures: Writing and executing stored procedures.

10) Cursors: Usage of cursors in database applications.

5. TEXT BOOKS

1) Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems*, 3rd Edition, Tata McGraw Hill.
2) Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, 6th Edition, McGraw Hill.

6. REFERENCE BOOKS

1) Peter Rob, Carlos Coronel, *Database Systems: Design, Implementation, and Management*, 7th Edition, Cengage.
2) Ramez Elmasri, Shamkant Navathe, *Fundamentals of Database Systems*, Pearson Education.
3) C.J. Date, *An Introduction to Database Systems*, Pearson Education.
4) Shah & Shah, *Oracle for Professionals*, SPD/PHI.
5) Shah, *Database Systems Using Oracle: A Simplified Guide to SQL and PL/SQL*, PHI.
6) M.L. Gillenson, *Fundamentals of Database Management Systems*, Wiley Student Edition.

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	**2**	**3**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 2	**2**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 3	**2**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 4	**2**	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 5	**2**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		

COURSE CONTENT

JAVA PROGRAMMING LAB								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM307PC	**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 laboratory course provides **hands-on programming experience in Java**, focusing on **object-oriented programming concepts** such as abstraction, inheritance, multithreading, exception handling, event-driven programming, and GUI development using Swing. Students will learn to work with Java tools such as Eclipse IDE and practice problem-solving through collection framework, applets, and file handling.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To write programs using abstract classes and inheritance.
2) To solve real-world problems using Java Collection Framework.
3) To implement multithreaded applications.
4) To design GUI programs using Swing controls.
5) To introduce Java compiler, Eclipse IDE, and debugging features.
6) To provide practical exposure to core Java concepts.

3. COURSE OUTCOMES

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

CO 1	Develop Java programs for solving real-world problems using Collection Framework.
CO 2	Apply abstraction and inheritance using abstract classes.
CO 3	Implement multithreaded applications with inter-thread communication.
CO 4	Design GUI applications using AWT and Swing controls.
CO 5	Develop file handling applications using Java I/O classes.

4. LIST OF EXPERIMENTS

1) Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2) Write a Java program to demonstrate the OOP principles. [i.e., Encapsulation, Inheritance, Polymorphism and Abstraction].

3) Write a Java program to handle checked and unchecked exceptions. Also, demonstrate the usage of custom exceptions in real time scenario.

4) Write a Java program on Random Access File class to perform different read and write operations.

5) Write a Java program to demonstrate the working of different collection classes. [Use package structure to store multiple classes].

6) Write a program to synchronize the threads acting on the same object. [Consider the example of any reservations like railway, bus, movie ticket booking, etc.]

7) Write a program to perform CRUD operations on the student table in a database using JDBC.

8) Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

9) Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. [Use Adapter classes].

5. REFERENCE BOOKS

1) Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2) Thinking in Java, Bruce Eckel, Pearson Education.
3) Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4) Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

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**	**3**		
CO 2	**3**	**3**	**3**	**2**	**2**	**1**	**-**	**-**	**-**	**2**	**2**	**2**		
CO 3	**3**	**2**	**3**	**3**	**2**	**1**	**-**	**-**	**-**	**2**	**2**	**3**		
CO 4	**3**	**3**	**3**	**3**	**3**	**1**	**-**	**-**	**-**	**2**	**2**	**2**		
CO 5	**3**	**2**	**3**	**3**	**2**	**1**	**-**	**-**	**-**	**2**	**2**	**3**		

COURSE CONTENT

NODE JS / REACT JS / DJANGO LAB								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM308PC	**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 **full stack web development** with emphasis on client-side design, server-side programming, and modern frameworks. Students begin by designing static and responsive websites, move on to database connectivity using Java, and then implement server-side applications with **Node.js**. The course concludes with building **single-page applications (SPAs)** using **React**.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To implement static and responsive web pages using **HTML, CSS, Bootstrap, and JavaScript**.
2) To perform **client-side validation** with JavaScript and use advanced ES6 features.
3) To design and interact with **databases** using JDBC.
4) To implement **server-side applications** using Java and Node.js.
5) To design and develop **single-page applications** with React.

3. COURSE OUTCOMES

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

CO 1	Build responsive web applications using HTML, CSS, Bootstrap, and JavaScript.
CO 2	Demonstrate advanced JavaScript features (ES6, promises, async/await) with real-time APIs.
CO 3	Implement database-driven applications using JDBC and servlets.
CO 4	Develop server-side implementations using Node.js and Express framework.
CO 5	Design and deploy **Single Page Applications (SPA)** using React.

4. LIST OF EXERCISES

1) Build a **responsive shopping cart web application** with registration, login, catalog, and cart pages using CSS3 (flex & grid).
2) Enhance the above web application using the **Bootstrap framework**.
3) Perform **client-side validation** using JavaScript for the pages developed in Experiments 1 and 2.
4) Explore **ES6 features** (arrow functions, callbacks, promises, async/await). Implement an application that fetches weather data from *openweathermap.org* and displays it graphically.
5) Develop a **Java standalone application** that connects to a database (Oracle/MySQL) and performs CRUD operations.
6) Create an **XML document** for a bookstore and validate it using both DTD and XSD.
7) Design a **Servlet-based controller** that integrates the shopping cart application with the database created in Experiment 5.
8) Explore and implement **session tracking mechanisms** (Cookies, HTTP Session) for maintaining user transactional history.

9) Create a **custom server in Node.js** using the HTTP module and explore additional Node.js modules (OS, path, event).
10) Develop an **Express.js web application** that interacts with a REST API to perform CRUD operations on student data. (Test using Postman).
11) Secure the above application by creating **authorized endpoints using JWT (JSON Web Token)**.
12) Create a **React application** for a student management system with registration, login, contact, and about pages. Implement **routing** for navigation.
13) Build a **React service** that fetches weather information from *openweathermap.org* and displays current and historical data graphically using **Chart.js**.
14) Create a **React TODO application** with reusable components and deploy it on **GitHub Pages**.

5. REFERENCE BOOKS

1) Jon Duckett, *Beginning HTML, XHTML, CSS, and JavaScript*, Wrox Publications, 2010.
2) Bryan Basham, Kathy Sierra, Bert Bates, *Head First Servlets and JSP*, O'Reilly Media, 2nd Edition, 2008.
3) Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, 2nd Edition, Apress.

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**	**3**	**3**	**2**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 2	**3**	**3**	**3**	**2**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 3	**3**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 4	**3**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 5	**3**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		

COURSE CONTENT

SOFTWARE ENGINEERING LAB								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
AM310PC	**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

A Software Engineering Lab course offers hands-on experience in applying software engineering principles through practical projects, covering requirement analysis, design, implementation, testing, and project management.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

3. COURSE OUTCOMES

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

CO 1	Ability to translate end-user requirements into system and software requirements.
CO 2	Ability to generate a high-level design of the system from the software requirements
CO 3	Will have experience and/or awareness of testing problems and will be able to develop a simple testing report
CO 4	Develop architectural and detailed software designs using standard modeling languages.
CO 5	Manage projects by applying software engineering principles, including configuration and risk management.

4. COURSE CONTENT

Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:

1) Development of problem statements.
2) Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3) Preparation of Software Configuration Management and Risk Management related documents.
4) Study and usage of any Design phase CASE tool
5) Performing the Design by using any Design phase CASE tools.
6) Develop test cases for unit testing and integration testing
7) Develop test cases for various white box and black box testing techniques.

Sample Projects:

1) Passport automation System
2) Book Bank
3) Online Exam Registration
4) Stock Maintenance System

5) Online course reservation system
6) E-ticketing
7) Software Personnel Management System
8) Credit Card Processing
9) E-book management System.
10) Recruitment system

5. TEXT BOOKS

1) Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2) Software Engineering- Sommerville, 7th edition, Pearson Education.
3) The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

6. REFERENCE BOOKS

1) Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2) Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill.

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**	**3**	**2**	**2**	**2**						**3**	**1**		
CO 2	**3**	**3**	**3**	**2**	**2**						**3**	**1**		
CO 3	**3**	**3**	**3**	**2**	**2**						**3**	**1**		
CO 4	**3**	**3**	**3**	**2**	**2**						**3**	**1**		
CO 5	**3**	**3**	**3**	**3**	**3**						**3**	**1**		

COURSE CONTENT

GENDER SENSITIZATION LAB								
II Year - I Semester: CSE (AI & ML)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
***MC309**	**Mandatory**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**0**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines – such as literature, history, economics, psychology, sociology, philosophy, political science, anthropology and media studies – to examine cultural assumptions about sex, gender, and sexuality. This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women, and of the multiple ways that sex and gender interact with race, class, caste, nationality and other social identities. This course also seeks to build an understanding and initiate and strengthen programmes combating gender-based violence and discrimination. The course also features several exercises and reflective activities designed to examine the concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To develop students' sensibility with regard to issues of gender in contemporary India.
2) To provide a critical perspective on the socialization of men and women.
3) To introduce students to information about some key biological aspects of genders.
4) To expose the students to debates on the politics and economics of work.
5) To help students reflect critically on gender violence.
6) To expose students to more egalitarian interactions between men and women.

3. COURSE OUTCOMES

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

CO 1	Students will have developed a better understanding of important issues related to gender in contemporary India.
CO 2	Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature and film
CO 3	Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
CO 4	Students will acquire insight into the gendered division of labor and its relation to politics and economics.
CO 5	Men and women students and professionals will be better equipped to work and live together as equals.

4. COURSE CONTENT

Unit-I: UNDERSTANDING GENDER

Introduction: Definition of Gender-Basic Gender Concepts and Terminology-Exploring Attitudes towards Gender-Construction of Gender-Socialization: Making Women, Making Men- Preparing for Womanhood. Growing up Male. First lessons in Caste.

Unit – II: GENDER ROLES AND RELATIONS

Two or Many? -Struggles with Discrimination-Gender Roles and Relations-Types of Gender Roles- Gender Roles and Relationships Matrix-Missing Women-Sex Selection and Its Consequences- Declining Sex Ratio. Demographic Consequences-Gender Spectrum: Beyond the Binary.

Unit – III: GENDER AND LABOUR

Division and Valuation of Labour-Housework: The Invisible Labor- "My Mother doesn't Work." "Share the Load."-Work: Its Politics and Economics -Fact and Fiction. Unrecognized and Unaccounted work.

-Gender Development Issues-Gender, Governance and Sustainable Development-Gender and Human Rights-Gender and Mainstreaming

Unit – IV: GENDER - BASED VIOLENCE

The Concept of Violence- Types of Gender-based Violence-Gender-based Violence from a Human Rights Perspective-Sexual Harassment: Say No!-Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: "Chupulu".

Domestic Violence: Speaking OutIs Home a Safe Place? -When Women Unite [Film]. Rebuilding Lives. Thinking about Sexual Violence Blaming the Victim-"I Fought for my Life…."

Unit – V: GENDER AND CULTURE

Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular Literature-Gender Development Issues-Gender Issues-Gender Sensitive Language-Gender and Popular Literature - Just Relationships: Being Together as Equals Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Mothers and Fathers. Rosa Parks- The Brave Heart.

Note: Since it is Interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field from engineering departments.

- Classes will consist of a combination of activities: dialogue-based lectures, discussions, collaborative learning activities, group work and in-class assignments. Apart from the above prescribed book, Teachers can make use of any authentic materials related to the topics given in the syllabus on "Gender".
- ESSENTIAL READING: The Textbook, "Towards a World of Equals: A Bilingual Textbook on Gender" written by A.Suneetha, Uma Bhrugubanda, DuggiralaVasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu published by Telugu Academy, Telangana Government in 2015.

ASSESSMENT AND GRADING:

- Discussion & Classroom Participation: 20%.
- Project/Assignment: 30%
- End Term Exam: 50%

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**	**3**					
CO 2								**3**	**3**					
CO 3								**3**	**3**					
CO 4								**3**	**3**					
CO 5								**3**	**3**					