COURSE BOOK B. TECH. II YEAR

CSE/CS/IT/CSIT/CSE(AI)/CSE(AI&ML)

KIET GROUP OF INSTITUTIONS
Connecting Life with Learning
www.kiet.edu
Delhi-NCR, Ghaziabad
CELEBRATING 25 YEARS OF EXCELLENCE

CURRICULUM STRUCTURE & SYLLABUS

Effective from the Session: 2025-26

1.1 Computer Science and Engineering (CSE)/Computer Science (CS)/ Computer Science and Information Technology (CSIT)

B.Tech (CSE/CS/CSIT) 3rd Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (Core)	IT	IT301L	Database Systems	L	3	0	0	60	15	75	75	150	3
2	PC	Major (Core)	CSE	CS206L	Operating System	L	3	0	0	60	15	75	75	150	3
3	PC	Major (Core)	CS	CS301L	Object Oriented Programming using Java	L	3	0	0	60	15	75	75	150	3
4	BS	Major (Core)	ASH	MA105L	Probability and Statistics	L	3	0	0	60	15	75	75	150	3
5	MC	Value Added	ASH	HS109L	Constitution of India	L	2	0	0	25	-	25	25	-	NC
6	HS	AEC	ASH	HS110L	Aptitude-1	L	1	0	0	-	25	25	25	50	1
7	HS	AEC	ASH	HS111L	Soft Skills Essential-1	L	1	0	0	-	25	25	25	-	NC
Blended															
8	PC	Major (Core)	CSE	CS302B	Advance Data Structure	B	3	0	2	80	20	100	100	200	4
9	PC	Major (Core)	CS	CS205B	Artificial Intelligence and It's Applications	B	3	0	2	80	20	100	100	200	4
Lab/Practical															
10	PC	Major (Core)	IT	IT301P	Database Systems Lab	P	0	0	2	-	25	25	25	50	1
11	PC	Major (Core)	CSE	CS206P	Operating System Lab	P	0	0	2	-	25	25	25	50	1
12	PC	Major (Core)	CS	CS301P	Object Oriented Programming using Java Lab	P	0	0	2	-	25	25	25	50	1
13	PW	Summer internship	CSIT	IT105P	Social Internship Assessment	P	0	0	0	-	50	50	-	50	1
Total Hours : 32 hrs.							**22**	**0**	**10**					**1250**	**25**

B. Tech (CSE/CS/CSIT) 4th Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (core)	CSE	CS401L	Design and Analysis of Algorithms	T	3	0	0	60	15	75	75	150	3
2	ES	Major (core)	IT	IT302L	Computer Networks	T	3	0	0	60	15	75	75	150	3
3	PC	Major (core)	CSE	CS208L	Web Technology	T	2	0	0	40	10	50	50	100	2
4	MC	Value Added	ASH	HS112L	Universal Human Values	T	3	0	0	60	15	75	75	150	3
5	HS	AEC	ASH	HS113L	Aptitude-2	T	1	0	0	-	25	25	25	50	1
6	HS	AEC	ASH	HS114L	Soft Skills Essential 2	T	1	0	0	-	25	25	25	-	NC
Blended															
7	PC	Major (core)	CSIT	IT202B	Data Analytics	B	2	0	2	60	15	75	75	150	3
8	PC	Major (core)	CS	CS303B	ANN and Machine Learning	B	2	0	2	60	15	75	75	150	3
9	PE	Major (core)/SEC	-	-	Professional Elective-I	B	3	0	2	80	20	100	100	200	4
Lab/Practical															
10	PC	Major (core)	CSE	CS401P	Design and Analysis of Algorithms Lab	P	0	0	2	-	25	25	25	50	1
11	ES	Major (core)	IT	IT302P	Computer Networks Lab	P	0	0	2	-	25	25	25	50	1
12	PC	Major (core)	CSE	CS208P	Web Technology Lab	P	0	0	2	-	25	25	25	50	1
Total Hours : 32 hrs.							**20**	**0**	**12**					**1250**	**25**

Professional Electives (PE)

S.No.	Course Type (PE)	DevOps: Development Operations	Full Stack Development	AI and Data Science	Cloud Ops and Security	Full Stack DevOps Engineering
1	**BOS**	**CSE**	**CSE**	**CSE(AI)**	**CSE(AI)**	**CSE(AI)**
	PE I-(4thSem)	DevOps Fundamentals (CS304E)	UI & UX Specialist (CS307E)	Cloud Foundation (AI103E)	Cloud Foundation (AI103E)	Cloud Foundation (AI103E)

1.2 Information Technology (IT)

B.Tech (IT) 3rd Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (Core)	IT	IT301L	Database Systems	L	3	0	0	60	15	75	75	150	3
2	PC	Major (Core)	CS	CS301L	Object Oriented Programming using Java	L	3	0	0	60	15	75	75	150	3
3	BS	Major (Core)	ASH	MA105L	Probability and Statistics	L	3	0	0	60	15	75	75	150	3
4	MC	Value Added	ASH	HS109L	Constitution of India	L	2	0	0	25	-	25	25	-	NC
5	HS	AEC	ASH	HS110L	Aptitude-1	L	1	0	0	-	25	25	25	50	1
6	HS	AEC	ASH	HS111L	Soft Skills Essential-1	L	1	0	0	-	25	25	25	-	NC
Blended															
7	PC	Major (Core)	CSE	CS302B	Advance Data Structure	B	3	0	2	80	20	100	100	200	4
8	PC	Major (Core)	IT	IT107B	Artificial Intelligence Essentials	B	3	0	2	80	20	100	100	200	4
9	PE	Major (Core)/SEC	-	-	Professional Elective-I	B	3	0	2	80	20	100	100	200	4
Lab/Practical															
10	PC	Major (Core)	IT	IT301P	Database Systems Lab	P	0	0	2	-	25	25	25	50	1
11	PC	Major (Core)	CS	CS301P	Object Oriented Programming using Java Lab	P	0	0	2	-	25	25	25	50	1
12	PW	Summer internship	CSIT	IT105P	Social Internship Assessment	P	0	0	0	-	50	50	-	50	1
Total Hours : 32 hrs.							**22**	**0**	**10**					**1250**	**25**

B. Tech (IT) 4th Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (core)	CSE	CS401L	Design and Analysis of Algorithms	L	3	0	0	60	15	75	75	150	3
2	ES	Major (core)	IT	IT302L	Computer Networks	L	3	0	0	60	15	75	75	150	3
3	PC	Major (core)	CSE	CS208L	Web Technology	L	2	0	0	40	10	50	50	100	2
4	PC	Major (Core)	CSE	CS206L	Operating System	L	3	0	0	60	15	75	75	150	3
5	MC	Value Added	ASH	HS112L	Universal Human Values	L	3	0	0	60	15	75	75	150	3
6	HS	AEC	ASH	HS113L	Aptitude-2	L	1	0	0	-	25	25	25	50	1
7	HS	AEC	ASH	HS114L	Soft Skills Essential 2	L	1	0	0	-	25	25	25	-	NC
Blended															
8	PE	Major (Core)/SEC	-	-	Professional Elective-II	B	3	0	2	80	20	100	100	200	4
Lab/Practical															
9	PC	Major (core)	CSE(AIML)	AI104P	Data Engineering Essentials	P	0	0	4	-	50	50	50	100	2
10	PC	Major (core)	CSE	CS401P	Design and Analysis of Algorithms Lab	P	0	0	2	-	25	25	25	50	1
11	ES	Major (core)	IT	IT302P	Computer Networks Lab	P	0	0	2	-	25	25	25	50	1
12	PC	Major (core)	CSE	CS208P	Web Technology Lab	P	0	0	2	-	25	25	25	50	1
13	PC	Major (Core)	CSE	CS206P	Operating System Lab	P	0	0	2	-	25	25	25	50	1
Total Hours : 33 hrs.							**19**	**0**	**14**					**1250**	**25**

Professional Electives (PE)

S.No.	Course Type (PE)	AI and Data Science	Cloud Ops and Security	Full Stack DevOps Engineering
1	**BOS**	**CSE(AI)**	**CSE(AI)**	**CSE(AI)**
	PE I-(3rd Sem)	Cloud Foundations (AI103E)	Cloud Foundations (AI103E)	Cloud Foundations (AI103E)
2	**BOS**	**CSE(AI)**	**CSE(AI)**	**IT**
	PE II-(4th Sem)	Generative AI Foundations & Applications (AI301E)	Cloud Computing & Operations (AI302E)	Foundations of Web Development (IT303E)

1.3 Computer Science and Engineering (AI)/ Computer Science and Engineering (AI&ML)

B.Tech (CSE(AI)/CSE(AIML)) 3rd Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (Core)	IT	IT301L	Database Systems	L	3	0	0	60	15	75	75	150	3
2	PC	Major (Core)	CS	CS301L	Object Oriented Programming using Java	L	3	0	0	60	15	75	75	150	3
3	BS	Major (Core)	ASH	MA105L	Probability and Statistics	L	3	0	0	60	15	75	75	150	3
4	MC	Value Added	ASH	HS109L	Constitution of India	L	2	0	0	25	-	25	25	-	NC
5	HS	AEC	ASH	HS110L	Aptitude-1	L	1	0	0	-	25	25	25	50	1
6	HS	AEC	ASH	HS111L	Soft Skills Essential-1	L	1	0	0	-	25	25	25	-	NC
Blended															
7	PC	Major (Core)	CSE	CS302B	Advance Data Structure	B	3	0	2	80	20	100	100	200	4
8	PC	Major (Core)	CSE(AIML)	AI201B	Machine Learning Essentials	B	3	0	2	80	20	100	100	200	4
9	PE	Major (Core)/SEC	-	-	Professional Elective-I	B	3	0	2	80	20	100	100	200	4
Lab/Practical															
10	PC	Major (Core)	IT	IT301P	Database Systems Lab	P	0	0	2	-	25	25	25	50	1
11	PC	Major (Core)	CS	CS301P	Object Oriented Programming using Java Lab	P	0	0	2	-	25	25	25	50	1
12	PW	Summer internship	CSIT	IT105P	Social Internship Assessment	P	0	0	0	-	50	50	-	50	1
Total Hours : 32 hrs.							**22**	**0**	**10**					**1250**	**25**

B. Tech (CSE(AI)/CSE(AIML)) 4th Sem

S No.	Course Category (AICTE)	Course Category (UGC)	BOS	Course Code	Course Name	Type	Academic Learning (AL)			Continuous Internal Examination (CIE)			End Sem Examination (ESE)	Total Marks	Total Credits
							L	T	P	MSE	CA	TOTAL		CIE+ ESE	
1	PC	Major (core)	CSE	CS401L	Design and Analysis of Algorithms	L	3	0	0	60	15	75	75	150	3
2	ES	Major (core)	IT	IT302L	Computer Networks	L	3	0	0	60	15	75	75	150	3
3	PC	Major (core)	CSE	CS208L	Web Technology	L	2	0	0	40	10	50	50	100	2
4	PC	Major (Core)	CSE	CS206L	Operating System	L	3	0	0	60	15	75	75	150	3
5	MC	Value Added	ASH	HS112L	Universal Human Values	L	3	0	0	60	15	75	75	150	3
6	HS	AEC	ASH	HS113L	Aptitude-2	L	1	0	0	-	25	25	25	50	1
7	HS	AEC	ASH	HS114L	Soft Skills Essential 2	L	1	0	0	-	25	25	25	-	NC
Blended															
8	PE	Major (Core)/SEC	-	-	Professional Elective-II	B	3	0	2	80	20	100	100	200	4
Lab/Practical															
9	PC	Major (core)	CSE(AIML)	AI104P	Data Engineering Essentials	P	0	0	4	-	50	50	50	100	2
10	PC	Major (core)	CSE	CS401P	Design and Analysis of Algorithms Lab	P	0	0	2	-	25	25	25	50	1
11	ES	Major (core)	IT	IT302P	Computer Networks Lab	P	0	0	2	-	25	25	25	50	1
12	PC	Major (core)	CSE	CS208P	Web Technology Lab	P	0	0	2	-	25	25	25	50	1
13	PC	Major (Core)	CSE	CS206P	Operating System Lab	P	0	0	2	-	25	25	25	50	1
Total Hours : 33 hrs.							**19**	**0**	**14**					**1250**	**25**

Professional Electives (PE)

S.No.	Course Type (PE)	AI and Data Science	Cloud Ops and Security	Full Stack DevOps Engineering
1	**BOS**	**CSE(AI)**	**CSE(AI)**	**CSE(AI)**
	PE I-(3rd Sem)	Cloud Foundations (AI103E)	Cloud Foundations (AI103E)	Cloud Foundations (AI103E)
2	**BOS**	**CSE(AI)**	**CSE(AI)**	**IT**
	PE II-(4th Sem)	Generative AI Foundations & Applications (AI301E)	Cloud Computing & Operations (AI302E)	Foundations of Web Development (IT303E)

1. Theory Courses Detail Syllabus

Course Code: IT301L	Course Name: Database Systems	L	T	P	C
		3	0	0	3

Pre-requisite: Concepts of any programming language

Course Objectives:

1. To develop a strong foundation in database management concepts.
2. To equip students with practical skills in database design, normalization, transaction management, and recovery techniques.

Course Outcome: After completion of the course, the student will be able to

1. Acquire knowledge of database design methodology for real-life applications
2. Design an information model using the concept of ER diagram
3. Apply the concept of SQL on real-life databases
4. Analyze the redundancy problem in the database and reduce it using normalization.
5. Identify the broad range of database management issues including data integrity, security, and recovery transactions, as well as enforce entity integrity, referential integrity, key constraints, and domain constraints on the database.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	1	1	1	1	1	1	2
CO2	2	2	3	2	3	1	1	1	2	2	2	2
CO3	3	3	2	1	3	1	1	1	1	1	1	2
CO4	3	3	2	2	2	1	1	1	1	1	1	2
CO5	3	3	2	2	3	2	1	2	1	1	1	2

Unit 1	Introduction to Database System	09 hours

Database System vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence and its Types, Overall Database Structure.

Entity Relationship Model: ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Key attribute, Generalization, Aggregation, Reduction of an ER Diagrams to Tables.

Use Case: Discuss one or two case studies like Banking System, and National Hockey League (NHL).

Unit 2	Relational data Model	09 hours

Relational Data Model Concepts, type of keys, Integrity Constraints- Entity integrity and referential integrity, Keys Constraints, Domain Constraints, Relational Algebra-Unary Relational Operations- SELECT and PROJECT, Binary Relational Operations-CROSS, JOIN and DIVISION, Queries in Relational Algebra.

Database Implementation using SQL: Introduction to SQL, Characteristics of SQL, SQL Data Types, Basic Queries in SQL- create, select Insert, Delete and Update Statements, concepts of group by and having, order by, Sub Queries, Aggregate Functions, Joins, Unions, Intersection, Minus, Views.

Note: Hands-on in the class/ Lab

Unit 3	Database Design and Normalization	09 hours

Functional Dependencies, Inference rules, Closure of attributes, FD equivalence and Minimal cover.

Normalization: Normal forms, first, second, third normal forms, and BCNF. Lossless join decompositions, Dependency Preservation, Multi-valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Note: Discuss one or two case studies for the Normalization. e.g.

1. In this a table would be given and students will be asked to normalize in a higher normal form

Schema along with FDs set will be given and students will be asked to normalize in a higher normal form

Unit 4	Transaction Processing	09 hours

Transaction and its States**,** ACID property, Transaction Scheduling, Serializability of scheduling, Conflict, and View Serializability

Concurrency Control Techniques: Concurrency Control, Locking Techniques for Concurrency Control, Two-phase locking techniques for concurrency control, Time Stamping Protocols for Concurrency Control

Unit 5	Database Recovery Techniques	09 hours

Recovery Concepts, Recoverability, Log Based Recovery, Checkpoints, Shadow Paging, The ARIES recovery, Deadlock Handling

PL/SQL: Introduction, features, syntax, DDL within Pl/SQL, DML in PL/SQL, Cursors, stored procedures, stored function, database triggers, indexing, **Case Study-Microsoft Azure SQL.**

Note: Hands-on in the class/ Lab

Total Lecture Hours	**45 hours**

Textbook:

1. Elmasri, Navathe, " Fundamentals of Database Systems", Addision Wesley
2. Korth, Silbertz, Sudarshan," Database Concepts", McGraw Hill
3. Date C J, "An Introduction to Database Systems", Addision Wesley

Reference Books:

1. Bipin C. Desai, "An Introduction to Database Systems", Gagotia Publications 8. Majumdar & Bhattacharya, "Database Management System", TMH
2. RAMAKRISHNAN"Database Management Systems", McGraw Hill
3. Rafe Colburn, Using SQL, PHI

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3 (ATTN)		
30	30	6	6	3		
60		15			75	150

Course Code: CS206L	Course Name: Operating System	L	T	P	C
		3	0	0	3

Pre-requisite: Basic knowledge on Computer System and system memory, Computer Organization and Logic Design (COLD)

Course Objectives:

This course aims to provide a comprehensive understanding of operating systems, their components, structures, and functionalities. It covers fundamental concepts such as process management, memory management, file systems, and concurrency. The course also emphasizes practical aspects, such as system calls, Linux commands, shell scripting, and scheduling algorithms. By the end of this course, students will have the knowledge and skills to understand, analyze, and apply operating system principles in real-world scenarios.

Course Outcome: After completion of the course, the student will be able to

1. Understand the need, evolution and design issues of various categories of operating systems.
2. Apply different CPU scheduling algorithms and deadlock handling methods.
3. Analyze the principles of concurrency control and process synchronization problem.
4. Analyze various memory management techniques for efficient memory allocation.
5. Apply the concept of various I/O management, Disk scheduling and file system.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	-	-	-	-	-	2
CO2	3	3	3	3	3	1	-	-	-	-	-	2
CO3	3	3	2	3	3	2	-	-	-	-	-	2
CO4	3	3	2	3	3	2	-	-	-	1	-	2
CO5	3	3	2	2	2	2	-	-	-	1	-	2

Unit 1	Introduction of Operating System	09 hours

Introduction: Operating system Components and its services, Classification of Operating systems- Batch system, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, Reentrant Kernels, Monolithic and Microkernel Systems. System Calls, Elementary Linux commands and Shell Scripting.

Unit 2	Process Scheduling and Resource Management	09 hours

Introduction to Process: Process States, State Transition Diagram, Schedulers, Process Control Block (PCB), Threads and their management,

CPU Scheduling: Concepts, Performance Criteria, Scheduling Algorithms.

Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.

Unit 3	Concurrent Processes	09 hours

Concurrent Processes: Principle of Concurrency, Critical Section Problem, Mutual Exclusion, Dekker's solution, Peterson's solution, Semaphores, Monitors, Test and Set operation; Classical Problem in Concurrency- Producer / Consumer Problem, Reader Writer Problem, Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models (IPC).		
Unit 4	**Memory Management**	**09 hours**
Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions and variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing.		
Unit 5	**I/O Management and Disk Scheduling:**	**09 hours**
File Systems and I/O Management of Windows and Linux. Disk Scheduling: Disk storage structure and disk scheduling algorithms, RAID. Case Study: Introduction to Android and Mac Operating System, The Evolution of Mobile Operating Systems: iOS vs. Android		
	Total Lecture Hours	**45 hours**

Text Book:

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley
2. Harvey M Dietel, " An Introduction to Operating System", Pearson Education
3. William Stallings, "Operating Systems: Internals and Design Principles ", 6th Edition, Pearson Education

Reference Book:

1. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education
2. D M Dhamdhere, "Operating Systems : A Concept based Approach", 2nd Edition, TMH.
3. Andrew S. Tanenbaum and Herbert Bros, Modern Operating Systems (4thEdition), Pearson

Mode of Evaluation

MSE		CA			ESE	Total	
MSE1	MSE2	CA1	CA2	CA3 (ATTN)			
30	30	6	6	3	75	150	
60		15					

Course Code: CS301L	**Course Name: Object Oriented Programming with Java**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Pre-requisite: NA

Course Objectives:

1. To familiarize students with the basic and advance Java Programming Language.
2. To learn modern tools to develop java-based web applications.

Course Outcome: After completion of the course, the student will be able to

1. Implement core Java concepts that model real world entities.
2. Develop programs based on new java features.
3. Apply a collection framework to build modular java programs.
4. Construct dynamic web pages using JDBC, Servlets and JSP.
5. Implement web and RESTful Web Services with Spring Boot using Spring Framework concepts.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	2	2	2	-	-	-	-	-	-	2
CO2	2	3	2	2	2	-	-	-	-	-	-	2
CO3	2	3	2	2	2	-	-	-	-	-	-	3
CO4	2	2	2	2	3	-	-	-	-	-	2	3
CO5	2	3	3	3	2	-	-	-	-	-	2	3

Unit 1	**Java Basics**	**09 hours**
Introduction to Java, JVM, JRE, Java Environment, Java Source File Structure, and Compilation. Defining Classes in Java, Constructors, Methods, Access Specifies, Static Members, Final Members, Class, Object, Abstraction, Inheritance,		

Encapsulation, Polymorphism, Exceptions: Use of try, catch, finally, throw, throws, In-built and User Defined Exceptions, Checked and Un-Checked Exceptions.		
Unit 2	**Java Features**	**09 hours**
Thread, Thread Life Cycle, Creating Threads, Thread Priorities; Advance Features: Functional Interfaces, Lambda Expression, Method References, Stream API, Default Methods, Static Method, For Each Method, Try-with-resources, Type Annotations, Repeating Annotations, Java Module System, Diamond Syntax with Inner Anonymous Class, Local Variable Type Inference, Switch Expressions, Yield Keyword, Records, Sealed Classes.		
Unit 3	**Java Collections**	**09 hours**
Collection in Java, Collection Framework in Java, Hierarchy of Collection Framework, Iterator Interface, Collection Interface, List Interface, ArrayList, LinkedList, Vector, Stack, Queue Interface, Set Interface, HashSet, LinkedHashSet, Sorted Set Interface, TreeSet, Map Interface, HashMap Class, Linked HashMap Class, TreeMap Class, Hashtable Class, Sorting, Comparable Interface, Comparator Interface, Properties Class in Java.		
Unit 4	**Advance Java**	**09 hours**
Java Database Connectivity (JDBC): Merging Data from Multiple Tables: Joining, Manipulating, Databases with JDBC, Prepared Statements, Transaction Processing, Stored Procedures, Servlet Overview and Architecture, Interface Servlet and the Servlet Life Cycle, Handling HTTP get Requests, Handling HTTP post Requests, Redirecting Requests to Other Resources, Session Tracking, Cookies, Session Tracking with Http Session, Java Server Pages, Implicit Objects, Scripting, Standard Actions, Directives.		
Unit 5	**Spring and Spring boot**	**09 hours**
MVC, Spring Core Basics, Spring Dependency Injection concepts, Spring Inversion of Control, AOP, Bean Scopes- Singleton, Prototype, Request, Session, Application, Web Socket, Auto wiring, Annotations, Life Cycle Call backs, Bean Configuration styles; Spring Boot Build Systems, Spring Boot Code Structure, Spring Boot Runners, Logger, BUILDING RESTFUL WEB SERVICES, Rest Controller, Request Mapping, Request Body, Path, Variable, Request Parameter, GET, POST, PUT, DELETE APIs, Build Web Applications.		
	Total Lecture Hours	**45 hours**

Textbook:

1. Herbert Schildt, "Java The complete reference", McGraw Hill Education
2. Steven Holzner, “Java Black Book”, Dreamtech.
3. Balagurusamy E, “Programming in Java”, McGraw Hill
4. Java: A Beginner’s Guide by Herbert Schildt, Oracle Press
5. Greg L. Turnquist “Learning Spring Boot 2.0 - Second Edition”, Packt Publication

Reference Books:

1. Kathey Sierra, “Head First Java”, O’Reilly
2. AJ Henley Jr (Author), Dave Wolf, “Introduction to Java Spring Boot: Learning by Coding”, Independently Published
3. Craig Walls, “Spring Boot in Action” Manning Publication

Mode of Evaluation

MSE		CA			ESE	Total	
MSE1	MSE2	CA1	CA2	CA3 (ATT)			
30	30	6	6	3	75	150	
60		15					

Course Code: MA105L	Course Name: Probability & Statistics	L	T	P	C
		3	0	0	3

Pre-requisite: X+2

Course Objectives:

1. To familiarize the graduate engineers with the concept of Statistics and Probability.
2. It aims to analyze the practical/ real life problems and solve them in scientific manner.

Course Outcome: After completion of the course, the student will be able to

1. Employ the concept of measure central tendency and regression analysis.
2. Apply knowledge of probability on distribution function.
3. Apply the concept of probability density function and normal distribution.
4. Apply the concept of random variable and time series.

5. Employ the knowledge of hypothesis by means of Chi-square and ANOVA test.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	-	-	-	-	1	-	-	-	1
CO2	2	2	2	-	-	-	-	1	-	-	-	1
CO3	2	2	1	-	-	-	-	1	-	-	-	1
CO4	2	2	1	-	-	-	-	1	-	-	-	1
CO5	2	2	2	-	-	-	-	1	-	-	-	1

Unit 1	**Basic Statistics**	**09 hours**

Introduction to Descriptive Statistics, Measure of Central Tendency, Histogram in sampling, Method of least square (basic concept), Fitting of Straight line and exponential curve, Correlation, Rank correlation and Regression Analysis.

Unit 2	**Probability I**	**09 hours**

Probability, Law of total Probability, Conditional Probability, Baye's Theorem, Discrete Random Variable, Probability Mass function. Binomial Distribution, Poisson Distribution., Introduction to confusion matrix.

Unit 3	**Probability II**	**09 hours**

Continuous Random Variable, Probability density function, Properties of Probability density function, Expectation and variance, Normal Distribution and its applications.

Unit 4	**Bivariate Random Variable and Time Series**	**09 hours**

Introduction to two dimensional random variable, Joint probability density function and its properties, Marginal probability distribution. Introduction to Time series, component of time series, Measure of trend (Graphic method, method of Averages)

Unit 5	**Sampling Theory**	**09 hours**

Introduction to Inferential Statistics, Testing of Hypothesis: Introduction, Sampling Theory (Small and Large), Hypothesis, Null hypothesis, Alternative hypothesis, Testing a Hypothesis, Level of significance, Confidence limits, t-test, Chi-square test, one way analysis of variance (ANOVA).

Total Lecture Hours	**45 hours**

Textbook:

1. B. V. Ramana, Higher Engineering Mathematics, McGraw-Hill Publishing Company Ltd., 2017
2. B. S. Grewal, Higher Engineering Mathematics, Khanna Publisher, 2017.
3. R K. Jain & S R K. Iyenger, Advance Engineering Mathematics, Narosa Publishing House 2002.
4. S. C. Gupta & V. K. Kapoor, Fundamental of Mathematical Statistics, Sultan Chand & Sons.

Reference Books:

1. Seymour Lipschutz, John Schiller, Introduction to Probability and Statistics, McGraw Hill
2. Peter V. O'Neil, Advance Engineering Mathematics, Thomson (Cengage) Learning, 2007.
3. TKV Iyenger, B. Krishna Gandhi, S. Ranganatham, MVSN Prasad, Probability and Statistics (S. Chand Publishing House).
4. E. Kreyszig, Advance Engineering Mathematics, John Wiley & Sons, 2005.

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA4 (ATT)		
30	30	6	6	3	75	150
60		15				

Course Code: HS109L	Course Name: Constitution of India	L	T	P	C
		2	0	0	NC

Pre-requisite: NA

Course Objectives:

1. To acquaint the students with legacies of constitutional development in India and help those to understand the most diversified legal document of India and philosophy behind it.
2. To make students aware of the theoretical and functional aspects of the Indian Parliamentary System.
3. To channelize students' thinking towards basic understanding of the legal concepts and its implications for engineers.
4. To learn procedure and effects of emergency, composition and activities of election commission and amendment

procedure.

Course Outcome: After completion of the course, the student will be able to

1. Understand basic features and modalities about Indian constitution.
2. Clarify the functioning of Indian parliamentary system at the center and state level.
3. Understand the aspects of Indian Legal System and its related bodies.
4. Apply different laws and regulations related to engineering practices.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	-	1	2		-	-	2
CO2	-	-	-	-	-	-	1	1	1	-	-	2
CO3	-	-	-	-	-	-	1	1	1	-	1	2
CO4	-	-	-	-	-	-	1	2	1	1	1	2

Unit 1	**Basic Information about Indian Constitution**	**08 hours**

Meaning of the constitution law and constitutionalism, Historical Background of the Constituent Assembly, Government of India Act of 1935 and Indian Independence Act of 1947, Enforcement of the Constitution, Indian Constitution and its Salient Features, The Preamble of the Constitution, Fundamental Rights, Fundamental Duties, Directive Principles of State Policy, Parliamentary System, Federal System, Centre-State Relations, Amendment of the Constitutional Powers and Procedure, The historical perspectives of the constitutional amendments in India, Emergency Provisions: National Emergency, President Rule, Financial Emergency, and Local Self Government – Constitutional Scheme in India.

Unit 2	**Union Executive and State Executive**	**08 hours**

Powers of Indian Parliament Functions of Rajya Sabha, Functions of Lok Sabha, Powers and Functions of the President, Comparison of powers of Indian President with the United States, Powers and Functions of the Prime Minister, Judiciary – The Independence of the Supreme Court, Appointment of Judges, Judicial Review, Public Interest Litigation, Judicial Activism, Lok Pal, Lok Ayukta, The Lokpal and Lok ayuktas Act 2013, State Executives – Powers and Functions of the Governor, Powers and Functions of the Chief Minister, Functions of State Cabinet, Functions of State Legislature, Functions of High Court and Subordinate Courts.

Unit 3	**Basic Information about Legal System**	**07 hours**

The Legal System: Sources of Law and the Court Structure: Enacted law -Acts of Parliament are of primary legislation, Common Law or Case law, Principles taken from decisions of judges constitute binding legal rules. The Court System in India and Foreign Courtiers (District Court, District Consumer Forum, Tribunals, High Courts, Supreme Court). Arbitration: As an alternative to resolving disputes in the normal courts, parties who are in dispute can agree that this will instead be referred to arbitration. Contract law, Tort, Law at workplace.

Unit 4	**Election provisions, Emergency provisions, Amendment of the constitution**	**07 hours**

Election Commission of India-composition, powers and functions and electoral process. Types of emergency-grounds, procedure, duration and effects. Amendment of the constitution- meaning, procedure and limitations

	Total Lecture Hours	**30 hours**

Textbook

1. Brij Kishore Sharma: *Introduction to the Indian Constitution*, 8th Edition, PHI Learning Pvt. Ltd.
2. Granville Austin: *The Indian Constitution: Cornerstone of a Nation (Classic Reissue)*, Oxford University Press.
3. S.G Subramanian: *Indian Constitution and Indian Polity*, 2nd Edition, Pearson Education 2020.
4. Subhash C. Kashyap: *Our Constitution: An Introduction to India's Constitution and constitutional Law*, NBT, 2018.
5. Madhav Khosla: *The Indian Constitution*, Oxford University Press.
6. PM Bakshi: *The Constitution of India*, Latest Edition, Universal Law Publishing.
7. V.K. Ahuja: *Law Relating to Intellectual Property Rights* (2007)
8. Suresh T. Viswanathan: *The Indian Cyber Laws, Bharat Law House*, New Delhi-88
9. P. Narayan: Intellectual Property Law, Eastern Law House, New Delhi
10. Executive programme study material Company Law, Module II, by ICSI (The Institute of Companies Secretaries of India) (Only relevant sections i.e., Study 1, 4 and 36).https://www.icsi.edu/media/webmodules/publications/Company%20Law.pdf
11. Handbook on e-Governance Project Lifecycle, Department of Electronics & Information Technology, Government of India, https://www.meity.gov.in/writereaddata/files/e-Governance_Project_Lifecycle_Participant_Handbook-5Day_CourseV1_20412.pdf
12. Companies Act, 2013 Key highlights and analysis by PWC.

https://www.pwc.in/assets/pdfs/publications/2013/companies-act-2013-key-highlights- and-analysis.pdf

Reference Books

1. Keshavanand Bharati V. State of Kerala, AIR 1973 SC 1461.
2. Maneka Gandhi V. Union of India AIR, 1978 SC 597.
3. S.R. Bammai V. Union of India, AIR 1994 SC 1918.
4. Kuldip Nayyar V. Union of India, AIR 2006 SC312.
5. A.D.M. Jabalpur V. ShivkantShakla, AIR 1976 SC1207.
6. Remshwar Prasad V. Union of India, AIR 2006 SC980.
7. Keshav Singh in re, AIR 1965 SC 745.
8. Union of India V. Talsiram, AIR 1985 SC 1416.
9. Atiabari Tea Estate Co.V. State of Assam, AIR 1961SC232.
10. SBP & Co. Vs. Patel Engg. Ltd. 2005 (8) SCC 618.
11. Krishna Bhagya Jala Nigam Ltd. Vs. G. Arischandra Reddy (2007) 2 SCC 720.
12. Oil & Natural Gas Corporation Vs. Saw Pipes Ltd. 2003 (4) SCALE 92 – 185.
13. Contemporary Newer case studies can be developed using AI tools

** (Other relevant case studies can be consulted by the teacher as per the topic). Prescribed Legislations:

1. Information Technology Act, 2000 with latest amendments. **Compare this with GDPR of Europe**
2. RTI Act 2005 with latest amendments.
3. Information Technology Rules, 2000
4. Cyber Regulation Appellate Tribunal Rules, 2000

Suggested aid for Students and Pedagogic purpose

- RSTV debates on corporate law, IPR and patent issues
- NPTEL lectures on IPR and patent rights

Episodes of 10 -part mini TV series "Samvidhan: The Making of Constitution of India" by RSTV.

Mode of Evaluation

MSE		CA			ESE	Total	
MSE	MSE2	CA1	CA2	CA4 (ATT)			
-	25	-	-	-	25	NC	
25		-					

Course Code: HS110L	Course Name: Aptitude-1	L	T	P	C
		1	0	0	1

Pre-requisite: NA

Course Objectives:

1. To provide adequate exposure to the students regarding the use of aptitude tests in the recruitment process and competitive examinations.
2. To improve the logical & numerical ability of the students

Course Outcome: After completion of the course, the student will be able to

1. Illustrate their comprehension by solving the given problems
2. Apply the learned concepts to new problems and solve them aptly.
3. Make use of their thought process to interpret and draw inferences from the given data to reach logical conclusions.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	-	1	-	1		-	-	-	-	1
CO2	1	1	-	1	-	2		-	-	-	-	1
CO3	1	1	-	1	-	1		-	-	-	-	2

Unit 1	Series , Coding and Decoding	04 hours

Importance and overview of Quantitative Aptitude and Logical Reasoning, Number Series, Letter Series, Analogies , Coding and Decoding.

Unit 2	Data Arrangement	04 hours

Ranking and Order, Direction Sense, Linear and Circular sitting arrangement.

Unit 3	Blood Relation and Puzzles	03 hours

Basic concepts, definition and terminology related to blood relationships, Conversation based blood relationships, Family Tree based problems, Coded relationships and related puzzles.		
Unit 4	**Critical and Non Verbal Reasoning**	**04 hours**
Statement arguments, course of action, classification and grouping of images, Figure series, Mirror image, Water image, Paper cutting, Paper folding, Embedded figures.		
	Total Lecture Hours	**15 hours**

Textbook

1. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publication.
2. https://www.geeksforgeeks.org/most-important-aptitude-topics-for-placements/

Reference Books

1. "How to Prepare for Logical Reasoning for the CAT" by Arun Sharma, TMH Publication
2. https://www.indiabix.com/logical-reasoning/questions-and-answers/
3. https://testbook.com/placement-aptitude/test-series

Mode of Evaluation

MSE		CA			ESE	Total	
MSE	MSE2	CA1	CA2	CA3 (ATT)			
-	-	10	10	5	25	50	
-		25					

Course Code: HS111L	Course Name: Soft Skills Essentials-1	L	T	P	C
		1	0	0	NC

Pre-requisite:

- Students should have foundational knowledge of grammar, vocabulary, and sentence structure to participate effectively in tasks like extempore, scenario writing after studying Communication skills subject in first year.
- Prior exposure to basic communication concepts (like verbal/non-verbal communication and listening skills) helps students to enhance persuasion, negotiation, and professional etiquette.

Course Objectives:

To develop students' communication, presentation, and interpersonal skills through interactive activities, elevating confidence and professionalism for academic and workplace success

Course Outcome: After completion of the course, the student will be able to

1. Demonstrate improved self-awareness and communication skills through structured presentations and vocabulary-building activities.
2. Apply effective verbal communication techniques, including pronunciation and elevator pitch delivery, to express ideas clearly and confidently.
3. Exhibit professional behaviour, grooming, and teamwork skills in group discussions, interviews, and workplace-related role plays.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	-	-	-	2	3	-	1
CO2	-	-	-	-	-	-	-	-	2	3	-	2
CO3	-	-	-	-	-	-	-	-	2	3	-	2

Unit 1	**Foundation of Communication and Self-Awareness**	**05 hours**
British Council-English Score Test, Team Presentations on Change Management Models, Presentations on Personality Profiling for professional growth		
Unit 2	**Verbal Communication and Clarity**	**04 hours**
Pronunciation Drill 1 & 2, Elevator Pitch Practice Session 1 & 2		
Unit 3	**Professionalism and Workplace Readiness**	**06 hours**
Professional Grooming and Etiquette, Group Discussion (General Topics), Panel Discussion on workplace scenarios using caselets		
	Total Lecture Hours	**15 hours**

Useful Resources:

1. www.mindtools.com

2. https://englishonline.britishcouncil.org/
3. www.toastmasters.org
4. https://www.futurelearn.com/
5. English Score Test
6. Duo Lingo Test

Mode of Evaluation

MSE		CA			ESE	Total	
MSE1	MSE2	CA1	CA2	CA3(ATT)			
-	-	10	10	5			
-		25			25	50	

Course Code: CS302B	Course Name: Advance Data Structure	L	T	P	C
		3	0	2	4

Pre-requisite: The course requires a background in mathematics and strong programming skills, along with fundamental knowledge of data structures and algorithms.

Course Objectives:

1. This course develops a deep understanding of advanced data structures for efficient problem-solving.
2. Students will analyze, implement, and apply complex structures like trees, heaps, graphs, and hashing.

Course Outcome: After completion of the course, the student will be able to

1. Analyze algorithms using asymptotic notations and apply the Divide & Conquer technique for efficient problem-solving.
2. Implement hashing techniques with collision resolution strategies for optimized searching and storage.
3. Construct and manipulate tree-based data structures, including BST, AVL, Red-Black, B-Trees, and Huffman coding.
4. Apply graph representations and traversal techniques like BFS, DFS, Connected Components, and Topological Sorting.
5. Utilize advanced data structures such as Binomial Heaps, Tries, Skip Lists, and Splay Trees for efficient data processing.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	-	-	-	1	-	-	1	2
CO2	3	3	3	2	-	-	-	1	-	-	1	2
CO3	3	3	3	2	-	-	-	1	-	-	1	2
CO4	3	3	3	2	-	-	-	1	-	-	1	2
CO5	3	3	3	2	-	-	-	1	-	-	1	2

Unit 1 **Introduction & Divide and Conquer Technique** **15 hours**

Introduction: Algorithms, Analyzing algorithms - Orders of Magnitude (Asymptotic notations), Growth of functions (constant, logarithmic, linear, polynomial, exponential). Divide & Conquer Technique: Binary search, Merge sort, Quick sort, Randomized quick sort, Efficiency analysis of sorting.

Problem solving:

1. Write a program to implement the Binary Search algorithm using Divide and Conquer technique on a sorted array. Accept an element from the user and search for it in the array. Display the number of comparisons made.
2. Given an array of integers nums sorted in non-decreasing order, and an integer target, find the starting and ending position of the target value. If the target is not found in the array, return [-1, -1]. Your solution must run in O (log n) time complexity.
3. Write a program to implement Merge Sort using Divide and Conquer. Accept n elements from the user, sort them, and display the sorted output along with the number of recursive calls and comparisons made.
4. Given an array of integers, count the number of inversions in the array. An inversion is a pair of elements (i, j) such that:
   - i < j and
   - arr[i] > arr[j]

   Your task is to write a program that returns the total number of such inversion pairs in the array.
5. Write a program to implement Quick Sort using the last element as the pivot. Accept an unsorted array, sort it, and count the number of comparisons and recursive calls.
6. Write a program to implement Randomized Quick Sort where the pivot is chosen randomly. Compare the number of comparisons and execution time with the regular Quick Sort.
7. Given an integer array nums, find the contiguous subarray (containing at least one number) which has the largest sum,

and return its sum using the Divide and Conquer approach.

Given an integer array nums and an integer k, return the kth largest element in the array. Note that it is the kth largest element in sorted order, not the kth distinct element.

Unit 2	Hashing	15 hours

Hashing & Searching Techniques: Introduction, Hash table, Hash function, Collision resolution technique: Open hashing (Separate Chaining), Closed Hashing - Linear Probing, Quadratic probing, Double Hashing.

Problem solving:

1. Write a program to implement a hash table using the **open hashing (separate chaining)** method. Use an array of linked lists to handle collisions. Support insertion, deletion, and search operations. Test your implementation by inserting multiple keys and observing how collisions are handled using chaining.
2. Given an array of integers, find the frequency of each distinct element using hashing and return a map of element to its count.
3. Given an array and an integer *k*, check whether the array contains duplicate elements within a distance *k* from each other using hashing.
4. Given an array and a value *sum*, determine whether any two numbers in the array add up to the sum using a hash set for efficient search.
5. Given an array and a window size *k*, print the count of distinct elements in every contiguous window of size *k* using hashing.
6. Given an integer array nums and an integer k, return the k most frequent elements. You may return the answer in **any order.**
7. Given an array of strings strs, group the anagrams together. You can return the answer in any order. Two strings are anagrams of each other if they contain the same characters in the same counts, but possibly in a different order.
8. Given an array of integers nums and an integer target, return **indices** of the two numbers such that they add up to the target.

Unit 3	Trees	15 hours

Tree: Binary Trees, Binary Search Trees (BST), Threaded Binary Trees, Huffman coding, Balanced Trees: AVL Tree & its operation (insertion, deletion).

Problem solving:

1. Write a program to implement a **binary tree** with basic operations such as insertion (level-order), in-order, pre-order, and post-order traversals. Test the tree with a set of integers and display the output of each traversal method.
2. Implement a **binary search tree (BST)** with functions to insert, delete, and search a node. Perform and display in-order traversal after each operation to show the structure of the BST.
3. Write a program to build a **Huffman tree** for a given set of characters and their frequencies, and generate Huffman codes for each character. Display the codes and the structure of the tree used in encoding.
4. Create a program to implement an **AVL Tree** that supports insertion of elements. Display the tree structure (in-order traversal) after each insertion and show the rebalancing operations (rotations) performed.
5. **Given** the root of a binary tree, **find** its **maximum depth** (also called **height**). The **maximum depth** is the number of nodes along the **longest path** from the root node down to the **farthest leaf node**.
6. Given two integer arrays, preorder and inorder, where:
   preorder is the Preorder Traversal of a binary tree,
   inorder is the Inorder Traversal of the same tree,
   Construct and return the binary tree.
7. Given the root of a binary tree, return the length of the diameter of the tree. The diameter of a binary tree is the longest path between any two nodes in the tree. This path may or may not pass through the root.
   Note: The length of a path between two nodes is measured by the number of edges between them.
8. Given a binary tree and two nodes p and q, find their lowest common ancestor (LCA). The LCA of two nodes p and q is the lowest node in the tree that has both p and q as descendants (where a node can be a descendant of itself).

Unit 4	Graph	15 hours

Graphs: Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Topological sort.

Problem solving:

1. Implement a graph using an adjacency matrix and provide functions to:
   - Add/remove vertices and edges.
   - Check if an edge exists between two vertices.

   Print the matrix.

2. Implement a graph using an adjacency list (linked lists or dynamic arrays) and support:
   - Edge insertion/deletion.
   - Display neighbors of a vertex.
3. Implement DFS to traverse a graph from a given start vertex and print the visit order. Handle disconnected graphs.
4. Implement BFS to traverse a graph and print vertices level by level.
5. Find all connected components in an undirected graph using DFS/BFS.
6. Perform topological sort on a directed acyclic graph (DAG) using Kahn's algorithm (BFS-based).
7. Implement DFS-based topological sort for a DAG.
8. Determine if a graph is bipartite (can be colored with 2 colors such that no adjacent nodes have the same color).
9. Given an m x n 2D binary grid grid which represents a map of '1's (land) and '0's (water), return the number of islands. An island is a group of adjacent '1's connected horizontally or vertically (not diagonally). You may assume all four edges of the grid are surrounded by water.

Unit 5	**Advanced Data Structure**	**15 hours**

Advanced Data Structure: Binomial Heap, Operations on Binomial Heap, Trie data structures, Priority Queue, Disjoint data structure.

Problem solving:

1. Implement a binomial heap and write a function to merge two binomial heaps.
2. Extract the minimum key from a binomial heap and rebalance the heap.
3. Given an undirected graph, detect if it contains a cycle using Union-Find.
4. Design a Trie (prefix tree) data structure that supports the following operations:
   - insert(word): Inserts the string word into the trie.
   - search(word): Returns True if the string word is in the trie (i.e., it was inserted before), and False otherwise.
5. Design a program that simulates a job scheduler using a Priority Queue, where each job has a priority and a name. It should support three operations: (1) Add a Job with a given priority and name, (2) Execute the highest priority job, resolving ties by insertion order, and (3) Display all pending jobs in descending priority order, maintaining insertion order for equal priorities. The implementation must use an efficient priority queue structure like a heap with a tie-breaker mechanism for insertion order.

Total Lecture Hours	**75 hours**

Text Book:

1. Cormen, Leiserson, Rivest, and Stein, Introduction to Algorithms (3 ed.), MIT Press, 2009. ISBN 978-0262033848.
2. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
4. Reema Thareja, "Data Structure Using C" Oxford Higher Education.
5. AK Sharma, "Data Structure Using C", Pearson Education India.

Reference Book:

1. V. Aho, J. E. Hopcroft, and J. D. Ullman, Data Structures and Algorithms (1 ed.), Pearson, 1983.
2. Dasgupta, Papadimitrou and Vazirani, Algorithms (3 ed.), McGraw-Hill Education, 2006.
3. Horowitz, Sahni, and Rajasekaran, Computer Algorithms (2 ed.), Silicon Press, 2007.
4. Kleinberg and Tardos, Algorithm Design (1 ed.), Pearson, 2005.

Mode of Evaluation

MSE		CA			ESE	Total	
MSE1	MSE2	CA1	CA2	CA3 (ATTN)			
40	40	8	8	4	100	200	
80		20					

Course Code: CS205B	**Course Name: Artificial Intelligence and It's Applications**	**L**	**T**	**P**	**C**
		3	**0**	**2**	**4**

Pre-requisite: Python Programming

Course Objectives:

1. To provide students with a foundational understanding of Artificial Intelligence (AI) concepts, tools, and practical implementations, including search algorithms, knowledge representation, and ethical considerations.

2. To equip students with hands-on experience in designing and implementing agentic AI systems, genetic algorithms, and their real-world applications across diverse domains such as healthcare, finance, and robotics.

Course Outcome: After completion of the course, the student will be able to

1. Demonstrate foundational understanding of Artificial Intelligence, its key domains, tools, and ethical implications by implementing logic-based systems and classical search algorithms
2. Design and simulate intelligent agents and multi-agent systems in diverse environments, incorporating concepts of decision-making, utility, and communication strategies.
3. Apply reinforcement learning techniques to develop learning agents and multi-agent environments, and visualize their behavior in real-world-inspired simulations and games.
4. Construct and optimize Genetic Algorithm-based solutions to solve complex real-world problems such as scheduling, path planning, and feature selection, while exploring hybrid and advanced evolutionary strategies.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	3	3	1	-	-	-	-	-	2
CO2	2	3	3	2	3	1	-	-	-	-	-	2
CO3	2	3	3	2	2	1	-	-	-	-	-	3
CO4	1	2	2	3	2	3	-	-	-	-	-	3
CO5	3	3	2	3	3	1	-	-	-	-	-	2

Unit 1	**Practical Foundations of Artificial Intelligence**	**18 hours**

Introduction to AI and its Tools & Libraries

Definition, History, and Evolution of AI, Key Domains: Machine Learning, Deep Learning, NLP, Computer Vision, AI Applications in Healthcare, Finance, and Robotics

Python for AI: NumPy, Pandas, and Scikit-learn

Introduction to TensorFlow/PyTorch

Using Jupyter Notebooks for Experiments

Search Algorithms with Implementation

Uninformed Search Strategies: BFS, DFS, Informed Search Strategies: A*, Greedy Best-First Search

Constraint Satisfaction Problems (CSPs)

BFS & DFS Implementation in Python

Developing a Maze Solver using Search Algorithms

Implementing A* Algorithm for Path Planning

Knowledge Representation & Ethics

Propositional Logic and First-Order Logic, Knowledge Graphs and Ontologies

Responsible AI Practices, Challenges of Fairness, Transparency, and Explainability

Implementing Logic-Based Systems with Python Libraries

Building a Rule-Based Expert System

Mini Projects:

Implement a Tic-Tac-Toe AI using Minimax Algorithm

Create a Pathfinding Visualizer for BFS/DFS/A*

Unit 2	**Foundations of Agentic AI and Intelligent Agents**	**19 hours**

Introduction to Agents and Agent Environments

Definition and Characteristics of Agents

Types of Agents: Reflex, Goal-Based, Utility-Based

Agent Environments: Fully vs. Partially Observable

Designing and Implementing Simple Agents in Python (4 hours)

Build Reflex and Goal-Based Agents

Smart Vacuum Cleaner Simulation

Self-Navigating Agent in a Virtual Grid

3. Multi-Agent Systems and Distributed AI (3 hours)

Overview of Multi-Agent Systems (MAS)

Coordination and Communication in MAS

Real-World Applications: Smart Grids, Collaborative Robot

4. Decision-Making and Agent Intelligence (4 hours)

Introduction to Markov Decision Processes (MDPs)

Basics of Game Theory and Nash Equilibrium

Designing Rule-Based Decision Logic Mini Project: Build a Reflex Agent for Maze Navigation		
Unit 3	**Learning Agents and Practical Agentic AI Systems**	**19 hours**
Learning Agents and Reinforcement Learning Learning Agents: Characteristics and Architectures Reinforcement Learning Fundamentals Introduction to OpenAI Gym Implementing RL Algorithms Q-Learning: Concepts and Python Implementation Deep Q-Networks (DQN): Architecture and Training Visualizing Agent Learning Progress Agentic AI for Games and Simulation Develop AI Agent for Snake or Flappy Bird Case Study: Self-Learning Bot in a Custom Game Reward Shaping and Environment Design Advanced Multi-Agent Learning and Simulations Cooperative and Competitive Agents Traffic Control Simulation using Multi-Agent Systems Reinforcement Learning in Multi-Agent Scenarios Mini Projects -Solve Frozen Lake Environment using Q-Learning		
Unit 4	**Genetic Algorithms & Applications**	**19 hours**
Fundamentals of Evolutionary Computation Introduction to Evolutionary Algorithms Darwinian Principles: Selection, Mutation, Crossover Genetic Algorithm Workflow and Structure Core Components of Genetic Algorithms Chromosome Representation Techniques Fitness Functions and Selection Strategies Crossover Techniques (One-point, Two-point, Uniform) Mutation Methods Implementing Genetic Algorithms in Python Writing a Basic GA from Scratch Custom Fitness Function Design Visualizing GA Evolution with Matplotlib Advanced Techniques and Hybridization (3 hours) Elitism and Diversity Preservation Parameter Tuning for Performance Optimization Hybrid GAs: Integration with Local Search and Simulated Annealing Introduction to Memetic Algorithms Practical Applications Solving the Traveling Salesman Problem (TSP) Feature Selection in Machine Learning Scheduling and Resource Allocation Problems Autonomous Route Planning (e.g., for vehicles or drones) Mini Project - Solve Optimal Timetable Scheduling for a University		
	Total Lecture Hours	**75 hours**

Textbook:

1. Subrata Saha (PHI Learning)
2. Manaranjan Pradhan & U. Dinesh Kumar (Wiley India)
3. S. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach‖, Prentice Hall, Third Edition, 2009.
4. I. Bratko, —Prolog: Programming for Artificial Intelligence‖, Fourth edition, Addison-Wesley Educational Publishers
5. M. Tim Jones, —Artificial Intelligence: A Systems Approach(Computer Science)‖, Jones and Bartlett Publishers,

Inc.;First Edition, 2008
6. Nils J. Nilsson, ―The Quest for Artificial Intelligence‖, Cambridge University Press, 2009.
7. William F. Clocksin and Christopher S. Mellish,‖ Programming in Prolog: Using the ISO Standard‖, Fifth Edition,Springer, 2003.
8. Gerhard Weiss, ―Multi Agent Systems‖, Second Edition, MIT Press, 2013.
9. David L. Poole and Alan K. Mackworth, ―Artificial Intelligence: Foundations of Computational Agents‖, Cambridge University Press, 2010.
10. Online Resource: NPTEL's free AI Course by Prof. Deepak Khemani, IIT Madras

Reference Books:

1. Artificial Intelligence, Rich and Knight, (TMH)
2. Artificial Intelligence with Python" Prateek Joshi (Indian Author, Packt Publishing)

Mode of Evaluation

MSE		CA			ESE	Total	
MSE1	MSE2	CA1	CA2	CA3 (ATT)			
40	40	8	8	4			
80		20			100	200	

Course Code: CS401L	Course Name: Design & Analysis of Algorithm	L	T	P	C
		3	0	0	3

Pre-requisite: The course requires a strong foundation in mathematics, algorithmic problem-solving, and proficient programming skills.

Course Objectives:

1. Develop a strong understanding of algorithm design techniques for efficient problem-solving.
2. Analyze algorithm complexity and explore NP-completeness, approximation, and randomized algorithms.

Course Outcome: After completion of the course, the student will be able to

1. Analyse algorithm efficiency using recurrence relations and sorting techniques.
2. Solve optimization problems with Greedy algorithms (MST, Shortest Path).
3. Implement Dynamic Programming for Knapsack, Matrix Chain Multiplication, etc.
4. Apply Backtracking and Branch & Bound to combinatorial problems.
5. Understand NP-Completeness, Approximation, and Randomized algorithms.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	2	-	-	1	-	-	1	2
CO2	3	3	3	2	2	-	-	1	-	-	1	2
CO3	3	3	3	3	2	-	-	1	-	-	1	2
CO4	3	3	2	3	2	-	-	1	-	-	1	2
CO5	3	3	2	3	2	-	-	1	-	-	1	2

Unit 1	Introduction & Sorting in Linear Time	09 hours

Introduction: Algorithms, Algorithm Analysis, Recurrence Relations- substitution, Master's method, Change of Variables, shell sort, Tim sort, Linear time sorting – Counting sort, Radix sort, Bucket sort.

Unit 2	Greedy	09 hours

Greedy Algorithm: General Characteristics, Problem solving using Greedy Algorithm- Fractional Knapsack problem, Activity selection problem. Task scheduling with deadline and penalty problem, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm). Single source shortest paths - Dijkstra's and Bellman Ford algorithms.

Unit 3	` Dynamic Programming	09 hours

Dynamic Programming: Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- optimal binary search trees, 0-1 Knapsack, Coin Change Problem, Assembly Line Scheduling, Longest Common Subsequence, Word

break problem, Matrix chain multiplication, Rod cutting problem, All pair shortest paths – Warshal's and Floyd's algorithms, Resource allocation problem.		
Unit 4	**Backtracking & Branch and Bound**	**09 hours**
Back Tracking: Introduction, Subset sum problem, Permutation, N-queen problem, Graph Coloring, Hamiltonian Cycles and Sum of subsets, analysis of these problems. Branch and Bound: Traveling salesman problem.		
Unit 5	**String Matching, Approximation Algorithm**	**09 hours**
String Matching: Introduction, Naive string matching algorithm, Rabin-Karp algorithm, String Matching with finite automata, KMP (Knuth Morris Pratt) matching algorithm, Boyer Moore String matching. Approximation algorithms: Travelling Salesman problem, Hamiltonian problem, Vertex Cover Problem. Introduction to NP-Completeness: The class P and NP, Polynomial reduction, NP Completeness Problem, NP-Hard Problems. Randomized Algorithm.		
	Total Lecture Hours	**45 hours**

Textbook:

1. Cormen, Leiserson, Rivest, and Stein, Introduction to Algorithms (3 ed.), MIT Press, 2009. ISBN 978-0262033848.

Reference Books:

1. Dasgupta, Papadimitrou and Vazirani, Algorithms (3 ed.), McGraw-Hill Education, 2006. ISBN 978-0073523408.
2. Design and Analysis of Computer Algorithms by Aho, Hopcroft and Ullman, Pearson.
3. Horowitz, Sahni, and Rajasekaran, Computer Algorithms (2 ed.), Silicon Press, 2007. ISBN 978-0929306414.
4. Kleinberg and Tardos, Algorithm Design (1 ed.), Pearson, 2005. ISBN 978-0321295354.

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA4 (ATT)		
30	30	6	6	3		
60		15			75	150

Course Code: IT302L	Course Name: Computer Networks	L	T	P	C
		3	0	0	3

Pre-requisite: Computer Fundamentals

Course Objectives:

The objective of this course is to provide insight about computer network concepts and to gain comprehensive knowledge of the layered communication architectures, its functionalities, key protocols as well as security issues related with each layer.

Course Outcome: After completion of the course, the student will be able to

1. Apply the knowledge of networking concepts and functionality of physical layer.
2. Explore the concept of elementary data link layer protocol to build a robust network.
3. Analyze the concept of routing and IP addressing in network layer.
4. Examine the usage and working of transport layer.
5. Determine the performance of different protocols used at application layer.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	2
CO2	3	3	-	-	2	-	-	-	-	-	-	2
CO3	3	3	-	-	2	-	-	-	-	-	2	2
CO4	3	3	-	-	-	-	-	-	-	-	-	2
CO5	3	3	-	-	-	-	-	-	-	-	-	2

Unit 1	**Introduction -Physical Layer**	**09 hours**
Introduction to computer networks: Network applications, Mode of communications, LAN, MAN, WAN, Internet, network hardware, network software, Design issues of layers, reference models: OSI, TCP/IP layers, and characteristics. **Physical Layer:** Network devices, Network topology, Transmission media, Signal transmission, Network performance and transmission impairments, the public switched telephone network, Switching techniques.		
Unit 2	**Data Link Layer**	**09 hours**

Data Link layer: Design issues, Framing, Error Detection and Correction, Flow control (Elementary Data Link Protocols, Sliding Window protocols).
Medium Access Control and Local Area Networks: Multiple access protocols: Random Access (ALOHA, CSMA, CSMA/CD, CSMA/CA), Controlled Access (Polling, Reservation, Token Passing), Channelization (FDMA, TDMA, CDMA). IEEE standards Wired LANs (Ethernet): Traditional Ethernet, Fast Ethernet, Gigabit Ethernet, Wireless LANs: IEEE 802.11 and Bluetooth.

Unit 3	**Network Layer**	**09 hours**

Network Layer: Network layer design issues, Logical addressing: Basic internetworking, Subnetting, Supernetting, NAT, network layer protocols, IPv4, IPv6
Routing: Static and dynamic routing, Routing algorithms and protocols (Shortest Path Algorithms, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing, Anycast Routing: RIP, OSPF, BGP)

Unit 4	**Transport Layer**	**09 hours**

Transport Layer : Process-to-process delivery, Transport layer protocols (UDP and TCP), SCTP, Connection management,
Congestion control in TCP: Congestion Control: Open Loop, Closed Loop choke packets; Quality of service: techniques to improve QoS: Leaky bucket algorithm, Token bucket algorithm

Unit 5	**Application Layer**	**09 hours**

Session Layer: Roles and responsibility of session layer protocols
Presentation Layer: Cryptography Symmetric-key cryptography, Asymmetric-key cryptography, Digital Signature,
Compression: Lossless and Lossy Compression
Application Layer: Domain Name System, World Wide Web and Hyper Text Transfer Protocol, Electronic mail, File Transfer Protocol, Simple Network Management Protocol, Telnet.

	Total Lecture Hours	**45 hours**

Textbook:
1. Behrouz Forouzan, "Data Communication and Networking" Fourth Edition-2006, Tata McGraw Hill
2. Andrew Tanenbaum "Computer Networks", Fifth Edition-2011, Prentice Hall.
3. William Stallings, "Data and Computer Communication", Eighth Edition-2008, Pearson.

Reference Books:
1. Kurose and Ross, "Computer Networking- A Top-Down Approach", Eighth Edition-2021, Pearson.
2. Peterson and Davie, "Computer Networks: A Systems Approach", Fourth Edition-1996, Morgan Kaufmann
3. Behrouz Forouzan, "TCP/IP Protocol Suite", McGraw Hill.

Mode of Evaluation

MSE		**CA**			**ESE**	**Total**
MSE1	**MSE2**	**CA1**	**CA2**	**CA (ATT)**		
30	**30**	**6**	**6**	**3**		
60		**15**			**75**	**150**

Course Code: CS208L	**Course Name: Web Technology**	**L**	**T**	**P**	**C**
		2	**0**	**0**	**2**

Pre-requisite: Web Development, HTML, CSS, JavaScript

Course Objectives:
1. To familiarize students with the modern technologies used in web technologies.
2. To realize the fundamentals of full stack development & python web development frameworks.

Course Outcome: After completion of the course, the student will be able to
1. To understand the fundamental concepts of web applications, web servers, and Node.js architecture, including asynchronous programming and event-driven architecture.
2. To develop and deploy web applications using Express.js and React, integrating state management techniques and handling errors efficiently.
3. To design and implement RESTful APIs and manage databases using MongoDB, performing CRUD operations and database connectivity in web applications.
4. To develop web applications using Flask and Django, utilizing MVC architecture, form handling, and REST API

development.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	-	-	-	-	-	-	-	2
CO2	3	3	3	2	-	-	-	-	2	2	2	2
CO3	3	3	3	2	-	-	-	-	2	2	2	2
CO4	2	2	2	-	-	-	-	-	2	-	-	2

Unit 1	**Mern Stack: Node.Js**	**07 hours**

Internet and WWW. Introduction to a web-app, basics of web server, fundamental web governing protocols. Introduction to MERN, Introduction to Node.js, Installation Process, blocking vs non-blocking architecture, npm, Event Loop, REPL, concepts of callbacks, Asynchronous Programming, WebSockets, events and EventEmitter, Node.js Streams, file system, http, url, os, console, error handling.

Unit 2	**Mern Stack: Express, React, Deployment**	**07 hours**

Introduction to Express, features and applications, Postman API platform, Routing(GET, POST, PUT, DELETE), concept of middle wares, Templating Engines(EJS, PUG, Handlebars), serving static files with express, Error Handling in Express, React.js: Components, Hooks, State Management (Redux, Context API), Stateless Components, Designing Components, State vs. Props, Connecting React frontend with Express.js backend.

Unit 3	**Mern Stack: Rest And Mongo**	**08 hours**

Introduction to RESTful API, principles, HTTP Methods in REST, SQL vs NoSQL databases, Introduction to MongoDB: installation, datatypes, collections, documents, Create Database, Create Collection, installing and using mongoose Insert, delete, update, join, sort, query, CRUD operations in MongoDB. Connect web application to DBMS.

Unit 4	**Web Development With Python: Django & Flask**	**08 hours**

Introduction to Flask: Installation, basic routing, request and response handling, Jinja2 templating, form handling, validation, Building REST APIs using Flask and Flask-RESTful.

Introduction to Django: MVC architecture, installation, project setup, connecting to databases, Creating models, migrations, querying the databases, Django Views & Templates, URL routing, template rendering, form handling

	Total Lecture Hours	**30 hours**

Textbook:

1. Beginning MERN Stack, Greg Lim.
2. Pro MERN Stack, Vasan Subarmanian
3. Two Scoops of Django, by Daniel Roy Greenfeld, Audrey Roy Greenfeld
4. Flask Web Development: Developing Web Applications with Python, Miguel Grinberg

Reference Books:

1. Ultimate Full-Stack Web Development with MERN: Design, Build, Test and Deploy Production-Grade Web Applications with MongoDB, Express, React and NodeJS, Nabendu Biswas
2. Mastering Flask Web Development: Build enterprise-grade, scalable Python web applications , Second Edition, Gaspar Jack Stouffer

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3 (ATTN)		
20	20	4	4	2		
40		10			50	100

Course Code: HS112L	**Course Name: Universal Human Values**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Pre-requisite: NA

Course Objectives:

1. To help students distinguish between values and skills, and understand the need, basic guidelines, content, and process of value education.

2. To help students initiate a process of dialog within themselves to know what they really want to be in their life and profession
3. To help students understand the meaning of happiness and prosperity for a human being.
4. To facilitate the students to understand harmony at all the levels of human living, and live accordingly.
5. To facilitate the students in applying the understanding of harmony in existence in their profession and lead an ethical life.

Course Outcome: After completion of the course, the student will be able to

1. Understand the significance of value inputs in a classroom, distinguish between values and skills, understand the need, basic guidelines, content, and process of value education, explore the meaning of happiness and prosperity, and do a correct appraisal of the current scenario in the society.
2. Distinguish between the Self and the Body, and understand the meaning of Harmony in the Self and the Co-existence of Self and Body.
3. Understand the value of harmonious relationships based on trust, respect, and other naturally acceptable feelings in human-human relationships and explore their role in ensuring a harmonious society.
4. Understand the harmony in nature and existence, and workout their mutually fulfilling participation in nature.
5. Distinguish between ethical and unethical practices, and start working out the strategy to actualize a harmonious environment wherever they work.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	2	2	2	1	-	1	1
CO2	-	-	-	-	-	2	2	2	1	-	1	1
CO3	-	-	-	-	-	2	2	2	1	-	1	1
CO4	-	-	-	-	-	2	2	2	1	-	1	1
CO5	-	-	-	-	-	2	2	3	1	-	1	1

Unit 1	Introduction to Value Education	10 hours

Understanding the need, basic guidelines, content, and process for Value Education, Self- Exploration–what is it? - its content and process; 'Natural Acceptance' and Experiential Validation –as the mechanism for self-exploration, Continuous Happiness, and Prosperity-A look at basic Human Aspirations, Right understanding, Relationship, and Physical Facilities-the basic requirements for fulfillment of aspirations of every human being with their correct priority, Understanding Happiness and Prosperity correctly – A critical appraisal of the current scenario, Method to fulfill the above human aspirations: understanding and living in harmony at various levels.

Unit 2	Understanding Harmony in the Human Being	10 hours

Understanding human being as a co-existence of the sentient 'I' and the material 'Body', Understanding the needs of Self ('I') and 'Body' - Sukh and Suvidha, Understanding the Body as an instrument of 'I' (I being the doer, seer, and enjoyer), Understanding the characteristics and activities of 'I' and harmony in 'I', Understanding the harmony of I with the Body: Sanyam and Swasthya; correct appraisal of Physical needs, the meaning of Prosperity in detail, Programs to ensure Sanyam and Swasthya.

Unit 3	Understanding Harmony in the Family and Society	10 hours

Harmony in Human-Human Relationship Understanding harmony in the Family-the basic unit of human interaction, Understanding values in the human-human relationship; meaning of Nyaya and program for its fulfillment to ensure Ubhay-tripti; Trust (Vishwas) and Respect(Samman) as the foundational values of relationship, Understanding the meaning of Vishwas; Difference between intention and competence, Understanding the meaning of Samman, Difference between respect and differentiation; the other salient values in a relationship, Understanding the harmony in the society (society being an extension of the family): Samadhan, Samridhi, Abhay, Sah- astitva as comprehensive Human Goals, Visualizing a universal harmonious order in society- Undivided Society (Akhand Samaj), Universal Order (Sarvabhaum Vyawastha) – from family to world family!

Unit 4	Understanding Harmony in Nature and Existence	09 hours

Whole existence as Co-existence Understanding the harmony in Nature, Inter connectedness, and mutual fulfillment among the four orders of nature- recyclability and self-regulation in nature, Understanding Existence as Co-existence (Sah-astitva) of mutually interacting units in all- pervasive space, Holistic perception of harmony at all levels of existence.

Unit 5	Implications of the above Holistic Understanding of Harmony on Professional Ethics	06 hours

Natural acceptance of human values, Definiteness of Ethical Human Conduct, Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order, Competence in Professional Ethics.

	Total Lecture Hours	45 hours

Textbook:

1. R R Gaur, R Asthana, G P Bagaria, 2019 (2nd Revised Edition), A Foundation Course in Human Values and Professional Ethics. ISBN 978-93-87034-47-1, Excel Books, New Delhi.

Reference Books:

1. Ivan Illich, Energy & Equity, The Trinity Press, Worcester and Harper Collins, USA,1974.
2. E.F. Schumacher, Small is Beautiful: a study of economics as if people mattered, Blond & Briggs, Britain,1973.
3. A Nagraj, Jeevan Vidya EkParichay, Divya Path Sansthan, Amarkantak 1998.
4. P L Dhar, RR Gaur, Science and Humanism, Commonwealth Publishers 1990.

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3 (ATT)		
30	30	6	6	3		
60		15			75	150

Course Code: HS113L	Course Name: Aptitude-2	L	T	P	C
		1	0	0	1

Pre-requisite: NA

Course Objectives:

1. To provide adequate exposure to the students regarding the use of aptitude tests in the recruitment process and competitive examinations.
2. To improve the logical & numerical ability of the students.

Course Outcome: After completion of the course, the student will be able to

1. Illustrate their comprehension by solving the given problems
2. Apply the learned concepts to new problems and solve them aptly.
3. Make use of their thought process to interpret and draw inferences from the given data to reach logical conclusions.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	-	1	-	1	-	-	-	-	-	1
CO2	1	1	-	1	-	2	-	-	-	-	-	1
CO3	1	1	-	1	-	1	-	-	-	-	-	2

Unit 1	Analytical Reasoning & Logical Puzzles	04 hours

Definition and Introduction of Concept and Relation of Cube and Cuboids, Cut the cube in different layer and then solve questions accordingly. Problems related with open and closed dice.

Unit 2	Syllogism	03 hours

Understanding of Venn diagram, Problems related with Venn diagram, Statement and Conclusion, Syllogism and reverse syllogism.

Unit 3	Clock and Calendar	04 hours

Definition and Introduction of Concept and Relation of angle and time, Overtaking, overlapping, right-angle and straight Angle with respect to time, Error in clock (faster and slower), Correct time of clock, Mirror and Water Image of clock, Introduction of Calendar, Concept of Normal and Leap Year, Finding Odd days, Finding the day of the week of given date with and without reference.

Unit 4	Data Interpretation and Critical Reasoning	04 hours

Tables (Understand of Table, Fillers in table), **Line Graph** (Understand the graph, Percentage change, Ratio based comparison), **Bar Graph** (Type of Bar Graph, Average and Comparison, Stacked Bar Graph), **Pi Chart** (Conversion of Percentage and Degree, Fillers in Pie chart, Multiple Pie chart), **Mixed Graph (problems** related with combination of various charts**) Critical Reasoning**: Assumptions, Cause and Effect, Assertion and Reason, Statement and Inference

Total Lecture Hours	15 hours

Useful resources:

1. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publication.
2. https://www.geeksforgeeks.org/most-important-aptitude-topics-for-placements/

Reference Books:

1. "How to Prepare for Logical Reasoning for the CAT" by Arun Sharma, TMH Publication.

2. https://www.indiabix.com/logical-reasoning/questions-and-answers/
3. https://testbook.com/placement-aptitude/test-series

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3 (ATT)		
-	-	10	10	5		
-		25			25	50

Course Code: HS114L	Course Name: Soft Skills Essentials 2	L	T	P	C
		1	0	0	NC

Pre-requisite:

- Successful completion of the subject 'Soft Skills Essentials-1' of the third semester.

Course Objectives:

To strengthen students' professional communication, cultural intelligence, and emotional awareness through advanced speaking activities, scenario-based discussions, and digital literacy tasks, equipping them for diverse workplace interactions.

Course Outcome: After completion of the course, the student will be able to

1. Apply advanced communication strategies that include vocabulary enhancement, storytelling to improve their cultural sensitivity (DEI).
2. Demonstrate prompt writing for AI-based tools and create effective elevator pitches to convey ideas with clarity and impact.
3. Exhibit interpersonal effectiveness by navigating negotiation, persuasion, and emotional intelligence in professional contexts

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	1	-	-	1	3	-	2
CO2	-	-	-	-	-	-	-	-	1	3	-	2
CO3	-	-	-	-	-	1	-	-	1	3	-	2

Unit 1	Advanced Communication and Cultural Sensitivity	7 hours

Vocabulary Enhancement through Gamification, Story Coining and Presentations Understanding Cross-Cultural, Communication (DEI) using Case Studies, Duo Lingo English Proficiency Tests

Unit 2	Professional Expression and Digital Literacy	4 hours

TMAY through Driver's test, Writing Effective Prompts on Various LLMs, Duo Lingo English Proficiency Tests

Unit 3	Interpersonal Effectiveness and Emotional Intelligence	4 hours

Negotiation & Persuasion Role Plays, Developing Emotional Intelligence via Scenario-Based Discussions

Total Lecture Hours	15 hours

Useful Resources:

1. https://youtu.be/5Wr-uaGzY7c
2. https://youtu.be/NcCwlqBapHo
3. https://youtu.be/SKNmQPIBPIg
4. RAISEC - B. Tech. MCA - Introduction
5. RAISEC - B. Tech. MCA - Social Personality Type
6. RAISEC - B. Tech. MCA - Enterprising Personality Type
7. RAISEC - B. Tech. MCA - Conventional Personality Type

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3		
-	-	10	10	5		
-		25			25	-

Course Code: IT303E	Course Name: Foundations of Web Development	L	T	P	C
		3	0	2	4

Pre-requisite: Computer Fundamentals

Course Objectives:

This course provides a comprehensive foundation in web development, including frontend, backend, databases, authentication, testing, deployment, and full-stack integration, with an emphasis on project-based learning and industry practices.

Course Outcome: After completion of the course, the student will be able to

1. Understand web fundamentals including HTML5, CSS3, and JavaScript (ES6).
2. Apply version control systems and manage projects using Git and GitHub.
3. Develop frontend applications with React.js and Tailwind CSS/Bootstrap.
4. Build backend servers and APIs using Node.js, Express.js, and integrate with MongoDB.
5. Test, deploy, and manage full-stack applications using modern tools.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	2	-	-	-	-	-	2	2
CO2	3	3	2	2	2	-	-	-	-	-	2	2
CO3	3	3	2	2	2	-	-	-	-	-	2	2
CO4	3	2	2	2	2	-	-	-	-	-	2	2
CO5	3	3	2	2	2	-	-	-	-	-	2	2

Unit 1	Web Fundamentals and Version Control	15 hours

Introduction to HTML5, CSS3, and JavaScript (ES6), DOM manipulation and responsive design basics, Introduction to Git and GitHub: repositories, branching, pull requests, Deliverables:Responsive portfolio webpage,Push project to GitHub
Case Study: Build and manage a responsive portfolio site with Git version control.

Unit 2	Frontend Development with React.js and Styling Frameworks	15 hours

Core React concepts: Components, Props, State, Lifecycle Hooks, Styling with Tailwind CSS / Bootstrap,Building UI components and dynamic frontend applications, Deliverables:React To-Do App, Styled product card
Case Study: Develop a React.js-based task manager with professional UI styling.

Unit 3	Backend Development and APIs using Node.js and Express.js	15 hours

Introduction to Node.js and Express.js, Creating RESTful APIs, CRUD operations, Connecting backend with MongoDB using Mongoose, Deliverables:CRUD API for tasks, Connect backend to MongoDB
Case Study: Develop a backend for a task management system integrated with MongoDB.

Unit 4	Authentication, Full Stack Integration, and Testing	15 hours

Authentication with JWT, Bcrypt: login/logout, MERN stack integration (React + Node + MongoDB), Testing APIs using Postman Writing unit tests with Jest, Deliverables:Auth-enabled blog backend, Working MERN App
Case Study: Build a secure blogging platform with authentication and full-stack integration.

Unit 5	Deployment and Capstone Project	15 hours

Deployment strategies for full-stack applications, Cloud platforms overview: Render, Vercel, AWS basics, Best practices for production-ready apps, Capstone Project:Build a real-world full-stack application from scratch, Host and deploy live.
Case Study: Deploy a full-stack application to a cloud service and ensure it's live and accessible.

Total Lecture Hours	75 hours

Textbook:

1. David Herron , Node.js Web Development, Third Edition
2. Kirupa Chinnathambi (2018), Learning React, Addison-Wesley
3. Ethan Brown (2019), Web Development with Node and Express, O'Reilly Media

Reference Books:

1. freeCodeCamp Web Development Certification
2. MDN Web Docs - Web Development
3. Node.js Docs

Mode of Evaluation

MSE		CA			ESE	Total				
MSE1	MSE2	CA1	CA2	CA3						
40	40	8	8	4						
80		20			100	200				

Course Code: AI103E	Course Name: Cloud Foundation	L	T	P	C
		3	0	2	4

Pre-requisite: Computer Fundamentals

Course Objectives:

This course introduces the fundamentals of cloud computing and AWS services, providing hands-on experience with compute, storage, networking, and security components. It aims to develop skills for monitoring and managing cloud environments, explore cloud adoption strategies, and provide students with AWS foundational and cloud career roles.

Course Outcome: After completion of the course, the student will be able to

1. Understand the fundamentals of cloud computing and core AWS services.
2. Configure AWS compute, storage, networking, and database services.
3. Implement security best practices using IAM, encryption, and network monitoring tools.
4. Apply cloud adoption frameworks and infrastructure as code (IaC) strategies in cloud environments.
6. Demonstrate proficiency in AWS learning tools for industry recognition.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	2	-	-	1	1	2	-	2
CO2	3	3	2	2	2	-	-	1	3	2	-	2
CO3	3	3	2	2	2	-	-	1	3	2	-	2
CO4	3	2	2	2	2	-	-	1	3	2	1	2
CO5	3	3	2	2	2	-	-	1	3	2	2	2

Unit 1	Introduction to Cloud Fundamentals and AWS Core Services	15 hours

Introduction and Walkthrough of Skill Builder, Job Roles in the Cloud, Introduction to Amazon EC2: Basic overview of launching, resizing, managing, and monitoring an Amazon EC2 instance, Introduction to AWS Identity and Access Management (IAM): add users to groups, manage passwords, log in with IAM-created users, and see the effects of IAM policies on access to specific services.
Case Study: Launching a Basic Web Server using EC2

Unit 2	AWS Infrastructure, Storage, and Cost Management Essentials	15 hours

AWS Networking Basics, getting started with AWS Storage, data transfer and data protection. Introduction to AWS Billing and Cost Management, AWS Database Offering, AWS Technical Essentials: fundamental AWS concepts related to compute, database, storage, networking, monitoring, and security. Configuring and deploying VPCs with multiple subnets.
Case Study: Create a Secure Storage and Backup System using Amazon S3

Unit 3	Advanced AWS Compute, Storage, and Cloud Adoption Strategies	15 hours

Built with Amazon EC2: configure, launch, and securely access EC2 instances, AWS Lambda Foundations, Introduction to Cloud adoption framework(CAF), Amazon Elastic Block Store (EBS): relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows., AWS Network – Monitoring and Troubleshooting.
Case Study: Serverless File Processing System using AWS Lambda

Unit 4	AWS Performance Optimization and Security Essentials	15 hours

Introduction to Elastic Load Balancer, testing a Network Elastic Load Balancer, Introduction to CloudFront, Data Encryption in AWS, Server Side Encryption (SSE) for Amazon S3, AWS Key Management Service (KMS), and Amazon DynamoDB Encryption Client, AWS Encryption SDK to encrypt and decrypt data.
Case Study: Deploy a Scalable Application with Security using the concepts of Elastic Load Balancer(ELB) and CloudFront.

Unit 5	AWS Learning Tools and Infrastructure as Code (IaC) Foundations	15 hours

Introduction to AWS CloudFormation, launch WordPress on Amazon Web Services, Cloud Essentials - Knowledge Badge, AWS services, security, architecture, pricing, and support, AWS Cloud Quest
Case Study: Deploy a Full Stack Application using AWS CloudFormation
Exam readiness: AWS Cloud Practitioner Certificate (CLF-C02)

Total Lecture Hours	75 hours

Textbook:

1. Piper, B., & Clinton, D. (2020). AWS Certified Solutions Architect Official Study Guide: Associate Exam. Wiley.
2. Wittig, M., & Wittig, A. (2018). Amazon Web Services in Action (2nd ed.). Manning Publications.
3. Wilkins, M. (2021). Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud. Addison-Wesley Professional.

Reference Books:

1. https://awsacademy.instructure.com//courses/100702

2. https://www.youtube.com/watch?v=UmQnenLf_Cs

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3		
40	40	8	8	4		
80		20			100	200

Course Code: CS304E	Course Name: DevOps Fundamentals	L	T	P	C
		3	0	2	4

Pre-requisite: NA

Course Objectives:

The objective of this course is to provide students with a comprehensive understanding of DevOps principles, practices, tools, and culture. It aims to bridge the gap between development and operations by introducing continuous integration, continuous delivery, infrastructure automation, version control systems, and measurement metrics to support faster and more reliable software delivery in modern IT environments.

Course Outcome: After completion of the course, the student will be able to

1. Understands the evolution of software engineering practices from Waterfall to Agile, and the emergence of DevOps.
2. Understands the principles of DevOps, its purpose, benefits, and its alignment with Agile methodologies.
3. Apply CAMS (Culture, Automation, Measurement, Sharing) principles to design scalable and reliable software delivery pipelines.
4. Demonstrate the use of version control tools like Git and GitHub for source code management and collaborative development.
5. Implement continuous integration and continuous delivery pipelines using appropriate tools and best practices.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	2	-	-	1	1	2	-	2
CO2	3	3	2	2	2	-	-	1	3	2	-	2
CO3	3	3	2	2	2	-	-	1	3	2	-	2
CO4	3	2	2	2	2	-	-	1	3	2	1	2
CO5	3	3	2	2	2	-	-	1	3	2	2	2

Unit 1 **Introduction to DevOps** **15 hours**

The Advent of Software Engineering, Waterfall method, Developers vs IT Operations conflict, Rise of Agile methodologies, Agile movement in 2000, Agile Vs Waterfall Method, Iterative Agile Software Development, Individual and team interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan. Definition of DevOps: Challenges of traditional IT systems & processes, History and emergence of DevOps, DevOps definition and principles governing DevOps, DevOps and Agile, The need for building a business use case for DevOps

Purpose of DevOps

Purpose of DevOps, Minimum Viable Product (MVP), Benefits of MVP, Application Deployment, Automated Application Deployment, Application Release Automation (ARA), Components of Application Release Automation (ARA), Continuous Integration, Best Practices of CI, Benefits of CI, Continuous Delivery.

Unit 2 **CAMS (Culture, Automation, Measurement and Sharing)** **15 hours**

CAMS – Culture, Cultural aspects of DevOps, Continuous Improvement and Problem Solving, encourage, Experimentation and Learning, CAMS – Automation, Delivering high value - DevOps way, Continuous Delivery Automation: CAMS – Measurement, Metrics used for tracking , Performance Predictors, Continuous Monitoring, CAMS – Sharing, Test-driven development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness ,Organizational Learning, Test Driven Development, TDD – Categories of Tests, Configuration Management, Source Code Management - Version Control, Infrastructure Automation Tools, Root Cause Analysis.

Unit 3 **Typical Toolkit of DevOps** **15 hours**

DevOps: An Overview, Achieving DevOps, Continuous Practices, how does CI Work? Continuous Integration Practices, Commit Code Frequently, Maintain a Single Source Repository, Don't commit code, Keep the Build Fast, Every Commit Should Build the Mainline, Fix Broken Builds Immediately, Write Automated Developer Tests, Automated deployment helps, Benefits of Continuous Integration, Continuous Delivery, Continuous Delivery Process, Benefits of continuous delivery practices

Unit 4 **Version Control Systems** **15 hours**

History of SCM : Brief History of Version Control Systems, Basic Operations in a VCS, Examples of Version Control Systems, Concurrent Versions System (CVS), Subversion (SVN), Mercurial, Git, The Making - History of Linux and Git, Advantages of Git

Version Control System vs Distributed Version Control System

Local Repository, Centralized Version Control System (CVCS), Distributed Version Control System (DVCS), Advantages of Distributed Version Control System , Private Workspace, Easier Merging, Easy to Scale

Horizontally , Disadvantages of Distributed Version Control System , vs Distributed Version Control Systems, Comparison of CVCS and DVCS, Multiple Repositories Model, Multiple Repositories for Different Services, Resetting the Local Environment, Revert - Canceling out the Changes

Unit 5	**Source Code Management Lab**	**15 hours**

1. Introducing Version Control System -Installing Git CLI on Linux, Mac & Windows
2. Setting up a Git Repository & Initialization
3. Cloning an Existing Repository
4. Introducing GitHub & Exploring GitHub
5. Fork and Push an Existing Repository
6. Working with Git- File States, Project Section, Workflow,
7. Working with Git Operations- git status, git add, git commit, git stage
8. Git Configuration Files -.gitattributes
9. Identifying Binary Files, Diffing Binary Files
10. Working with Git History
11. Merge Resolution in Git
12. Working with Git Branching

Total Lecture Hours	**75 hours**

Textbook:

1. "The Phoenix Project" by Gene Kim, Kevin Behr, and George Spafford A novel-based introduction to DevOps culture and practice.
2. "The DevOps Handbook" by Gene Kim, Jez Humble, Patrick Debois, and John Willis Covers practical implementation and case studies on DevOps.
3. "Accelerate: The Science of Lean Software and DevOps" by Nicole Forsgren, Jez Humble, and Gene KimResearch-based strategies for high-performance DevOps.

Reference Books:

1. DevOps Foundations – LinkedIn Learning(Beginner-friendly videos on core DevOps concepts.)
2. Microsoft Learn – DevOps with GitHub (Hands-on labs and tutorials focused on GitHub and CI/CD pipelines.)
3. Docker & Kubernetes Tutorials – Docker Official Docs (Essential for infrastructure automation and deployment topics.)
4. Git Handbook – GitHub Docs (Reference for Git commands and version control practices.)

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3		
40	40	8	8	4		
80		20			100	200

Course Code: CS307E	**Course Name: UI & UX Specialist**	**L**	**T**	**P**	**C**
		3	**0**	**2**	**4**

Pre-requisite: NA

Course Objectives:

1. Provide foundational knowledge of HTML5, CSS3, and JavaScript for building interactive web pages.
2. Introduce modern frontend architectural concepts including DOM manipulation, ECMAScript (ES6) features, and React.js.
3. Enable students to consume REST APIs, use JSX and manage component state and props in React.
4. Familiarize students with backend integration using Node.js and the development of full-stack JavaScript applications.
5. Equip learners with practical skills to build and deploy responsive and scalable web applications.

Course Outcome: After completion of the course, the student will be able to

1. Design responsive web pages using HTML5 and style them using CSS3.

2. Develop client-side scripts using JavaScript to create interactive functionality.
3. Apply modern frontend development practices including DOM manipulation, ES6+ features, and React basics.
4. Use REST APIs and JSX to implement dynamic React components with state and props management.
5. Develop simple backend services using Node.js and integrate them with frontend components.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	3	-	-	-	2	2	-	2
CO2	3	3	2	2	3	-	-	-	2	2	-	2
CO3	3	3	2	2	3	-	-	-	3	3	1	3
CO4	3	3	3	2	3	-	-	-	3	3	2	3
CO5	3	3	3	2	3	-	-	-	3	3	3	3

Unit 1	**HTML& CSS**	**15 hours**

Introduction to HTML, Introduction, HTML Page Structure, Create HTML document, Understand the various elements available in HTML, HTML Use, Attributes in HTML, Need of Attributes, Common Attributes, HTML forms, Apply validations to the form elements, Creating web pages with HTML5, HTML5 introduced features, HTML5 form validate/no validate, HTML5 canvas, embedding audio, and video in a webpage, drag and drop,HTML5 Local Storage, HTML5 web workers and server sent events, What is CSS, how to insert CSS in HTML, How CSS adds value to HTML, Difference between Semantic and HTML mark-up, CSS 3, CSS Selectors, Buttons, CSS float and clear, CSS align - horizontal and center, CSS Padding, CSS Links, CSS Lists, CSS Tables

Unit 2	**JavaScript**	**15 hours**

What is JavaScript, Importance of JavaScript, What can JavaScript Do?, JavaScript with HTML Attributes, JavaScript with CSS, Operators, JavaScript Syntax, JavaScript Data Types, JavaScript Functions, Setting up Environment, Variables, Control flow, if. Else, switch, loops, JavaScript HTML DOM Elements, JavaScript Syntax, Operators, Data Types, JavaScript String Methods, JavaScript Functions, Arrays, Sorting, Joins, Reduce map

Unit 3	**Frontend Architecture**	**15 hours**

Introduction to Frontend Development, History, MVC, MVP, MVVM& Web Apps, Development of AJAX, Introduction to DOM, Basic DOM Manipulation, Reactive Programming, Refreshing ES6 Specifications and Features ECMA Script, ES6 let and const, The arrow functions, New Literal Syntax, Classes, Inheritance using extends, Default Parameter Values, Spread Operator (...), Iterators and Generators, Features of React, Practical Application, Why need React, How React Works, Leveraging Virtual DOM, Setting up React

Unit 4	**REST API, JSX**	**15 hours**

Why JSX, Embedding JavaScript Expression in JSX, JSX Attributes, JSX Comments, Styling and Representation as Object, The State of the Component, Changing the State, Props of Component, Using Props, Props Validation in React, Similarities Between State & Props REST API : Intro to API, History of API Development, Development of AJAX, CRUD; GraphQL; HTTP , HTTP 1.1, HTTP/2, Stream prioritization, Introduction to React Native, Setting up React Native, The Expo Client, Working up on the First Project, Style, Fexbox Layout

Unit 5	**Node.js**	**15 hours**

Introduction to Node.js, History, Why Node.js, Node.js Architecture, Features, Working of Node.js, Installation & Setting Up Node, setting up React, REPL Environment, REPL Commands, Variable, Components of Node.js, Local Modules, Module Exports: Export Object, Export Class, Loading Module from Separate Folder, Operating System, File Systems

	Total Lecture Hours	**75 hours**

Textbook:

1. "HTML and CSS: Design and Build Websites" – Jon Duckett
2. "Eloquent JavaScript: A Modern Introduction to Programming" – Marijn Haverbeke
3. "Learning React" – Alex Banks and Eve Porcello
4. "Node.js Design Patterns" – Mario Casciaro

Reference Books:

1. JavaScript.info (In-depth JavaScript tutorials with ES6+ features.)
2. React Official Docs (For JSX, component lifecycle, props, state, etc.)
3. Node.js Official Docs (Detailed documentation for modules, REPL, file system APIs.)
4. FreeCodeCamp – Full Stack Certification (Structured learning from basics to full stack web development.)https://www.youtube.com/watch?v=UmQnenLf_Cs

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3		
40	40	8	8	4		
80		20			100	200

Course Code: AI201B	Course Name: Machine Learning Essentials	L	T	P	C
		3	0	2	4

Pre-requisite: Introduction to AI, Python

Course Objectives:

The objective of this course is to have a basic understanding of all the concepts and algorithms of Machine Learning and to apply in real world problems, report on the performance metrics with model optimization.

Course Outcome: After completion of the course, the student will be able to

1. Understand fundamental Machine Learning concepts, including supervised, unsupervised, and reinforcement learning.
2. Implement regression, classification, and ensemble learning techniques using industry-standard tools like Scikit-learn and TensorFlow.
3. Apply dimensionality reduction and clustering methods for unsupervised learning and data analysis.
4. Optimize ML models using hyperparameter tuning, probabilistic modeling, and advanced optimization techniques.
5. Deploy ML models with MLOps practices, ensuring fairness, transparency, and cloud-based implementation.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	2	3	3	3	3	3	3
CO2	3	3	3	3	3	2	3	2	3	3	3	3
CO3	3	3	3	3	3	2	3	3	3	3	3	3
CO4	3	3	3	3	3	3	3	3	3	3	3	3
CO5	3	3	3	3	3	3	3	3	3	3	3	3

Unit 1	Supervised Learning-Regression and Classification	15 hours

Overview of ML, types (supervised, unsupervised, reinforcement), and state-of-the-art applications (e.g., predictive maintenance, customer churn prediction).Linear Regression: Hypothesis function, Cost function, gradient descent (batch, stochastic, mini-batch), learning rate tuning, and regularization techniques (Lasso, Ridge) for robustness.Polynomial Regression: Bias-var okiance trade-off, cost functions (MSE, RMSE, MAE, R^2), and cross-validation for model selection.Logistic Regression: Activation functions (sigmoid, SoftMax), and state-of-the-art performance metrics (ROC-AUC, precision-recall curves).Tools: Use of sci-kit-learn, TensorFlow 2.x, and pandas for data handling, reflecting industry-standard libraries.

Case Study: Customer Churn Prediction using Sagemaker Notebooks/ Sagemaker Studio

Unit 2	Supervised Learning-Decision Tree, Ensemble Learning, SVM, Instance-Based Learning, Bayesian Networks	12 hours

Decision Trees: CART (Gini impurity), ID3 (entropy, information gain), and state-of-the-art feature importance analysis. Regularization: Hyperparameter tuning (max depth, min samples split) to prevent overfitting.Ensemble Learning:Bagging: Random Forests with out-of-bag (OOB) evaluation. Boosting: AdaBoost, Gradient Boosting, and XGBoost (state-of-the-art for structured data).

SVM: Linear SVM, nonlinear SVM with polynomial and RBF kernels, and SVM regression.Instance-Based Learning: KNN with distance-weighted voting and efficient implementation via KD-Trees.Bayesian Networks: Probabilistic modeling, state-of-the-art Naive Bayes variants (e.g., Gaussian NB), and Expectation-Maximization (EM) for hidden variables.

Case Study: Decision Tree/ XGBoostusing Sagemaker Notebooks/ Sagemaker Studio

Unit 3	Unsupervised Learning - PCA & Clustering	08 hours

Dimensionality Reduction: PCA, Incremental PCA for large datasets, and visualization with t-SNE. Clustering:K-Means with elbow method and silhouette analysis. Hierarchical clustering with dendrograms.DBSCAN, RT DBSCAN, and Optics for density-based clustering

Case Study: Based on SVM/ KNN using Sagemaker Notebooks/ Sagemaker Studio

Unit 4	Machine Learning Engineering on AWS	25 hours

Machine Learning (ML) Engineering on Amazon Web Services (AWS), build, deploy, orchestrate, and operationalize ML solutions at scale.AWS services such as Amazon SageMaker AI, analytics tools such as Amazon EMR to develop robust, scalable, and production-ready machine learning applications.

Unit 5	Deep Learning and Reinforcement Learning	15 hours

Deep Learning: Overview of Deep Learning (DL) concepts. AWS services for DL Reinforcement Learning: Q Learning, Markov Decision Processes (MDP):Markov Property, State Transition Probability using Amazon Sagemaker RL and AWS

Robomaker, Gaussian Mixture Models (GMM): State-of-the-art anomaly detection with GMM. Optimization: Gradient-based techniques (e.g., Adam optimizer) using AWS Model Tuning and hyperparameter tuning with tools like Optuna.Deep Learning using AWS,Ethics: Bias mitigation with Fairness Indicators, privacy considerations, and transparency in ML models using Sagemaker clarify configure security, and Amazonmacie.

AWS Cloud Quest: Machine Learning

Exam Readiness: AWS Certified Machine Learning Engineer - Associate (MLA-C01)

Exam Prep Official Practice Question Set: AWS Certified Machine Learning Engineer - Associate (MLA-C01)

Exam Prep Enhanced Course: AWS Certified Machine Learning Engineer - Associate (MLA-C01)

Total Lecture Hours	**75 hours**

Textbook:

1. Géron, A. (2019). Hands-on machine learning with scikit-learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.
2. Alpaydin, E. (2020). Introduction to machine learning (4th ed.). MIT Press.

Reference Books:

1. NORVIG, P. R. (2021). Artificial intelligence: A modern approach, 4th edition, Pearson
2. Mitchell, T. M. (1997). Machine learning.

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3 (ATT)		
40	40	8	8	4		
80		20			100	200

Course Code: AI301E	Course Name: Generative AI Foundations & Applications	L	T	P	C
		3	0	2	4

Pre-requisite: Introduction to AI, Python, ML Essentials

Course Objectives:

The objective is to apply key AWS services for text, speech, document processing, and generative AI applications, build AI-powered solutions using Amazon Q, Lex, Kendra, QuickSight, and demonstrate readiness for the AWS Certified AI Practitioner exam through hands-on projects.

Course Outcome: After completion of the course, the student will be able to

1. Apply AWS services like Amazon Textract, Transcribe, and Comprehend for intelligent document processing and NLP tasks.
2. Describe the core concepts, use cases, risks, and business applications of Generative AI on AWS.
3. Design and build conversational interfaces and intelligent search solutions using Amazon Lex and Kendra.
4. Create generative AI-powered business analytics dashboards using Amazon Q and QuickSight.
5. Demonstrate readiness for the AWS Certified AI Practitioner exam through hands-on labs and mock assessments.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	3	-	-	-	-	2	-	2
CO2	2	2	2	2	3	-	-	-	-	2	-	2
CO3	2	2	3	2	3	-	-	-	2	2	2	2
CO4	2	2	3	2	3	-	-	-	2	2	2	2
CO5	3	2	2	2	3	-	-	-	-	2	2	2

Unit 1	**Speech, Text, and Document AI on AWS**	**15 hours**

Amazon Transcribe: Speech to Text, Amazon Textract: Intelligent Document Processing, Amazon Comprehend: Natural Language Processing (NLP)

Case Study: Getting Started with Textract, Transcribe, Comprehend Custom Classification, Entity Recognition,Automated Invoice Processing System

Unit 2	**Generative AI Foundations on AWS**	**15 hours**

Introduction to Generative AI: Art of the Possible, Planning a Generative AI Project, Building a Generative AI-Ready Organization, Amazon Bedrock Overview: Foundation Models (FMs) *Case Study: Planning Generative AI Projects, Content Generator for E-commerce*		
Unit 3	**Practical Generative AI Implementation**	**15 hours**
Amazon Q Introduction, Amazon Q Developer, Amazon Q Business Setup, Generative BI with Amazon Q in QuickSight *Case Study Code Optimization with Q Developer, Business Case Setup, QuickSight Dashboards, Internal AI Assistant for Developers*		
Unit 4	**Specialized AWS AI Services**	**15 hours**
Amazon Lex: Conversational AI, Amazon Kendra: Intelligent Search, Additional Smart Search Solutions *Case Study: Setup of Kendra Search Index, Conversational Bot with Lex, Smart Customer Support Chatbot*		
Unit 5	**AWS AI Certification Preparation**	**15 hours**
Generative AI Essentials on AWS (Deep Dive), Exam Prep Enhanced Course: AWS Certified AI Practitioner, Official Practice Question Set for AWS AI Practitioner *Certification Readiness Project*		
Total Lecture Hours		**75 hours**

Textbook:

1. Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
2. Amazon Web Services. (2020). *AWS Certified Machine Learning Specialty: Official Study Guide*. Wiley.

Reference Books:

1. Cross, J. (2022). *Natural Language Processing with AWS AI Services*. Packt Publishing.
2. Snively, B., & Barth, A. (2024). *Practical Generative AI with Amazon Bedrock* (Draft Edition). O'Reilly Media.
3. Amazon Web Services. (2024). *Architecting Generative AI Solutions on AWS* (Training Material). AWS Training Resources.

Mode of Evaluation

MSE		CA			ESE	Total
MSE1 40	MSE2 40	CA1 8	CA2 8	CA3 (ATT) 4		
80		20			100	200

Course Code:AI302E	Course Name: Foundations of Cloud Computing & Operations	L	T	P	C
		3	0	2	4

Pre-requisite: Computer Fundamentals

Course Objectives:

This course aims to provide students with a strong foundation in Amazon Web Services (AWS) cloud concepts, core services, and operational best practices. Through theoretical learning, hands-on labs, and gamified exercises, students will gain cloud resource provisioning, monitoring, automation, security auditing, infrastructure deployment, and troubleshooting skills. The course prepares learners for practical cloud operations and industry certifications.

Course Outcome: After completion of the course, the student will be able to

1. Understand the fundamental concepts of AWS cloud computing, core services, and architecture.
2. Configure and manage cloud infrastructure components, including compute, storage, databases, and networking services on AWS.
3. Implement security best practices, perform audits, and manage user access using IAM, Trusted Advisor, CloudTrail, and Config.
4. Deploy automated and scalable infrastructure using AWS CloudFormation, Control Tower, and Service Catalog.
5. Monitor, troubleshoot, and optimize cloud applications and operations and develop practical hands-on skills through gamified learning, such as AWS Cloud Quest.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	2	-	-	1	-	-	-	2

CO2	3	3	2	2	2	-	-	1	-	-	-	2
CO3	3	3	2	2	2	-	-	1	-	-	-	2
CO4	3	2	2	2	2	-	-	1	-	-	-	2
CO5	3	3	2	2	2	-	-	1	-	-	-	2

Unit 1	**AWS Managed Services, Monitoring, and Cloud Governance**	**15 hours**

Understand how AWS Managed Services (AMS) operates within cloud operating models, learn about cloud operating models and their value, and know when different models are used, Configure AWS Federated Authentication with Active Directory Federation Services (AD FS) and Identity and Access Management (IAM) to allow AD users and groups to access the AWS Management Console, Learn Amazon EC2 observability, monitoring, and troubleshooting techniques, recognize the importance of EC2 health monitoring, and interact with AWS tools for troubleshooting common issues, Explore AWS Control Tower features and architecture, create and manage landing zones using AWS Organizations for ongoing account management and governance, Design Landing Zone architectures with AWS Control Tower, plan migrations to AWS Cloud, set up landing zones, and streamline AWS accounts using the AWS Well-Architected Framework.

Case Study: Setting Up a Multi-Account AWS Environment with Monitoring and Governance

Unit 2	**AWS Resource Management, Troubleshooting, and Infrastructure Automation**	**15 hours**

Learn the benefits, typical use cases, and technical concepts of AWS Config for resource management and compliance. Understand how to view, troubleshoot, and support AWS CloudFormation stacks by identifying and solving common stack issues. Explore AWS CloudFormation Designer as a visual tool for creating and modifying CloudFormation templates, with a basic understanding of EC2 and CloudFormation prerequisites.

Case Study: Automating and Monitoring a Cloud Infrastructure

Unit 3	**AWS Application Deployment, Monitoring, and Performance Tracing**	**15 hours**

Learn to build and deploy products such as VPCs, S3 buckets, EC2 instances, and RDS databases using AWS Service Catalog. Understand the setup and benefits of AWS CloudTrail, create trails to archive, analyze, and monitor changes in AWS resources, and store logs in Amazon S3.Explore AWS X-Ray for end-to-end tracing of distributed applications, detect performance bottlenecks, and troubleshoot hard-to-detect application issues.

Case Study: Building and Monitoring a Secure Web Application Infrastructure.

Unit 4	**AWS Security Auditing and Cloud Operations Management**	**15 hours**

Learn to audit AWS resources using AWS Trusted Advisor, ensure compliance with security best practices, manage security groups, implement multifactor authentication (MFA), and use Identity and Access Management (IAM). Perform a basic audit of core AWS services using the AWS Management Console and audit resources like EC2, VPC, IAM, Security Groups, CloudTrail, and CloudWatch. Understand cloud operations functions, including installing, configuring, automating, monitoring, securing, maintaining, and troubleshooting AWS networks and systems, and learn best practices and tools for cloud operations.

Case Study: Conducting a Security and Operations Audit in AWS

Unit 5	AWS Cloud Operations and Practical Skill Development	**15 hours**

Learn about managing and operating automatable and repeatable deployments of networks and systems on AWS, including installing, configuring, automating, monitoring, securing, maintaining, and troubleshooting cloud services and systems.Develop practical AWS skills through AWS Cloud Quest, a role-based learning game involving exercises and hands-on activities across various AWS services.

Case Study: Managing Cloud Operations and Building AWS Skills

Total Lecture Hours	**75 hours**

Textbook:

1. Stojanovic, L. (2021). AWS Certified Cloud Practitioner Study Guide: Foundational (CLF-C01) Exam. Sybex.
2. Bashir, W. (2020). Mastering AWS Security: Create and Maintain a Secure Cloud Ecosystem. Packt Publishing.
3. Singh, S., & Jansen, D. (2022). AWS for Solutions Architects: Design Your Cloud Infrastructure by Implementing DevOps, Security, and Serverless Architecture on AWS. Packt Publishing.
4. Varghese, B. (2022). Cloud Computing: Concepts and Technologies. Oxford University Press.
5. Wittig, A., & Wittig, M. (2021). Amazon Web Services in Action (3rd ed.). Manning Publications.

References:

1. https://awsacademy.instructure.com//courses/100702
2. https://www.youtube.com/watch?v=UmQnenLf_Cs

Modeof Evaluation						
MSE		CA			ESE	Total
MSE1 40	MSE2 40	CA1 8	CA2 8	CA3(ATT) 4		
80		20			100	200

Course Code: IT107B	Course Name: Artificial Intelligence Essentials	L	T	P	C
		3	0	2	4

Pre-requisite: Python and Data Structure

Course Objectives:

To equip students with a foundational understanding of core AI concepts and hands-on proficiency in AWS AI services for designing, implementing, and evaluating intelligent solutions while preparing for the AWS Certified AI Practitioner exam.

Course Outcome: After completion of the course, the student will be able to

1. Understand AI fundamentals, its history, key concepts and ethical considerations.
2. Apply state-space search methods, expert-system paradigms, and Amazon Kendra for intelligent search solutions.
3. Use AWS Comprehend, Transcribe, and Textract to perform NLP tasks.
4. Design and implement generative AI applications using Amazon Bedrock and Amazon Q.
5. Design and deploy end-to-end AI solutions by integrating AWS services and formulate a strategy for AWS Certified AI Practitioner exam preparation.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	2	-	-	3	-	-	-	2
CO2	3	3	2	2	3	-	-	-	-	-	-	2
CO3	2	2	2	2	3	-	-	-	-	-	-	2
CO4	2	2	3	2	3	-	-	-	2	-	2	2
CO5	2	2	3	2	3	-	-	-	3	-	3	2

Unit 1	Introduction to AI and AWS AI Services	15 hours

Introduction to AI: Evolution,AI from Turing Test to Humanoids, Applications, AI vs ML vs DL vs Data Science, Overview of AI Technologies: NLP, Computer Vision, Robotics, Future of AI: Tools, Trends, Ethics & Bias in AI, Introduction to AWS AI Practitioner Path, Getting Started with AWS cloud concepts.

Case Study: How Amazon Alexa uses AI.

Unit 2	AI Search Techniques and Rule-Based Systems	15 hours

Problem Solving: State Space, Search, Breadth-First Search (BFS), Depth-First Search (DFS), Heuristic Search: A*, Greedy Algorithm, Introduction to Rule-Based Expert Systems, Mapping AWS: Amazon Kendra for Intelligent Search. Getting started with Amazon Lex.

Case Study: Building a chatbot with AWS Lex.

Unit 3	Natural Language Processing (NLP) and AWS Tools	15 hours

Introduction to NLP: Tokenization, Stemming, Lemmatization, Sentiment Analysis, Using AWS Comprehend: Custom Classification and Entity Recognition, Introduction to Amazon Transcribe (Speech to Text), Amazon Textract: Process Documents with Synchronous and Asynchronous Operations.

Case Study: Automating customer feedback analysis using AWS Comprehend.

Unit 4	Introduction to Generative AI	15 hours

Generative AI: Basics, Applications, Foundation Models and Introduction to Amazon Bedrock, Planning a Generative AI Project, Building a Generative AI-Ready Organization, Amazon Q: Quick Introduction, Developer and Business.

Case Study: AI-generated product descriptions using Amazon Bedrock.

Unit 5	Real World AI Applications and Exam Preparation	15 hours

Generative AI Essentials on AWS, Business Intelligence with AWS QuickSight, Smart Applications: Recommendation Systems, Chatbots, Exam Prep Strategies for AWS AI Practitioner Certification, Practice Exam Questions.

Case Study: Personalized shopping experience using Generative AI.

Total Lecture Hours	75 hours

Textbook:

1. Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.
2. Manish Kumar (2021). Artificial Intelligence: Principles and Applications. McGraw Hill Education.
3. Vohra, D. (2022). Learning Amazon QuickSight. Packt Publishing.
4. Tutorials Dojo. (2024). AWS Certified AI Practitioner Study Guide AIF-C01.
5. Amazon Web Services. (2024). AWS Certified AI Practitioner (AIF-C01) Official Study Guide.

References:

1. https://portal.tutorialsdojo.com/product/study-guide-ebook-aws-certified-ai-practitioner-aif-c01/
2. https://aws.amazon.com/certification/certified-ai-practitioner/
3. https://docs.aws.amazon.com/

Mode of Evaluation

MSE		CA			ESE	Total
MSE1	MSE2	CA1	CA2	CA3 (ATT)		
40	40	8	8	4		
		20			100	200

3. Practical's Courses Detail Syllabus

Course Code: IT301P	Course Name: Database Systems Lab	L	T	P	C
		0	0	2	1

Pre-requisite: Concepts of any programming language

Course Objectives:

1. Develop Hands-on SQL Skills
2. Design and Normalize Databases
3. Work with PL/SQL and Advanced Database Concepts

Course Outcome: After completion of the course, the student will be able to

1. Design Logical and Conceptual database schema for real life problem using ERD
2. Apply SQL to store, retrieve and manipulate data in relational database
3. Apply PL/SQL to solve real world database management and automation task

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	3	1	3	1	1	1	2	2	1	2
CO2	3	3	2	1	3	1	1	1	1	1	1	2
CO3	3	3	2	1	3	1	1	1	1	2	1	2

List Of Practical's (Indicative & Not Limited To)

1. Creating Entity-Relationship Diagram using case tool (STAR UML)

[Group formation and assigned project to submit a report at the end of the semester as project-based learning]

1	Insurance Database
2	Banking Enterprise Database
3	Supplier Database
4	Student Faculty Database
5	Airline Flight Database
6	Order Processing Database
7	Book dealer Database
8	Student Enrolment Database
9	Movie Database

10	College Database
11	Hostel Database
12	Library Database
13	Clinic Database
14	Medical Store Database
15	Exam Process Database

2. Introduction to date types and database implementation using Create, insert, and Basic SQL SELECT statements. Create these two tables with following specifications and insert data in the table:

Table Name: Client master

Attribute	Data Type	Size
Client_no	number	10
Client_Name	Varchar2	20
City	Varchar2	15
State	Varchar2	15
Pin	Number	6
Balance_due	Number	10,2

Data for Client master:

CLIENT_NO	Client_NAME	CITY	STATE	PIN_CODE	BAL_DUE
0001	Ivan	Bombay	Maharastra	400057	15000
0002	Vandura	Madras	Tamilnadu	980001	0
0003	Pramod	Bombay	Maharastra	400057	5000
0004	Basu	Bombay	Maharastra	400056	0
0005	Ravi	Delhi	Null	100001	2000
0006	Rukmini	Bombay	Maharastra	900050	0

PRODUCT_MASTER

COLUMN	DATA TYPE	Size
PRODUCT_NO	VARCHAR2	6
DESCRIPTION	VARCHAR2	20
PROFIT%	NUMBER	10
QTY_ON_HAND	NUMBER	10
ORDER_LEVEL	NUMBER	10
SELL_PRICE	NUMBER	10
COST_PRICE	NUMBER	10

Data for Product_Master Table

Product_No	Description	Profit %	Qty_on hand	Reorder_level	Sell price	Cost price
P00001	1.44 floppies	5	100	20	525	500
P03453	Monitors	6	10	3	12000	11200
P06734	Mouse	5	20	5	1050	500
P07865	1.22 floppies	5	100	20	525	500
P07868	Keyboards	2	10	3	3150	3050
P07885	CD drive	2.5	10	3	5250	5100
P07965	540 HDD	4	10	3	8400	8000

P07975	1.44 Drive	5	10	3	1050	1000
P08865	1.22 Drive	5	2	3	1050	1000

Perform following queries on the above data:

a. Find out the name of all the clients.
b. Retrieve the list of names and cities of all the clients.
c. List all the clients who are located in Bombay.
d. Display the information for client no 0001 and 0002.
e. Find the list of all clients who stay in city 'Bombay' or 'Delhi' or 'Madras'.
f. List the name, city, and state of clients not in state of 'Maharashtra'

3. To manipulate data in the existing tables

Using the table client master and product master answer the following queries:

a. Delete the record of Client no. 0001 from the Client master table.
b. Change the city of Client no. 0005 to 'Bombay'.
c. Change the balance due of Client no. 0002 to 1000.
d. Find out the clients who stay in a city or state where second letter is a.
e. Calculate the average balance due of all the clients.
f. Change the selling price of 1.44 floppy drive to Rs. 1150.00.
g. Count the number of products having price greater than or equal to 1500.

4. To create, manage tables with constraints and alter the structure of tables.

CREATE THE TABLES WITH FOLLOWING SPECIFICATIONS AND CONSTRAINTS:

TABLE NAME: SALES MASTER

ATTRIBUTE	DATA TYPE	SIZE	CONSTRAINT
SALESMAN_NO	VARCHAR2	6	PRIMARY KEY, FIRST LETTER IS 'S'
SALES_NAME	VARCHAR2	20	NOT NULL
ADDRESS	VARCHAR2	20	NOT NULL
CITY	VARCHAR2	20	---
STATE	VARCHAR2	20	---
PINCODE	NUMBER	6	---
SAL_AMT	NUMBER	8,2	NOT NULL, CAN'T BE ZERO
Tgt_to_get	NUMBER	6,2	NOT NULL, CAN'T BE ZERO
Ytd_sales	NUMBER	6,2	NOT NULL, CAN'T BE ZERO
Remark	VARCHAR2	30	

TABLE NAME: SALES ORDER

ATTRIBUTE	DATA TYPE	SIZE	CONSTRAINT
S_ORDER_No	VARCHAR2	6	PRIMARY KEY,FIRST LETTER IS 'O'
S_ORDER_DATE	DATE	---	---
CLIENT_NO	NUMBER	10	FOREIGN KEY FROM CLIENT MASTER
SALESMAN_NO	VARCHAR2	26	FOREIGN KEY FROM SALES MASTER
DELIVERY_TYPE	CHAR	1	P FOR PARTIAL AND F FOR FULL, DEFAULT IS F
BILLED_YN	CHAR	1	'Y' FOR YES AND 'N' FOR NO
DELIVERY_DATE	DATE	---	CAN'T BE LESS THAN S_ORDER_DATE
ORDER_STATUS	VARCHAR2	10	IN(IN-PROCESS,FULFILLED,BACK ORDER,CANCELLED)

TABLE NAME: Sales_order_detail

Column	Dadatype	Size	Attributes
S_order_no	Varchar2	6	PK/FK references s_order_no of sales_order
Product_no	Varchar2	6	PK/FK references product_no of product_master
Qty_order	Number	8	
Qty_disp	Number	8	
Product_rate	Number	10,2	

DATA OF SALES_MASTER:

Sales No.	Sales_Name	Address	City	Pincode	State	Salamt	Tgt_to_get	Ytd_sales	Remark
S00001	Kiran	A/14 worli	Bombay	400002	MAH	3000	100	50	Good
S00002	Manish	65, Nariman	Bombay	400001	MAH	3000	200	100	Good
S00003	Ravi	P-7, Bandra	Bombay	400032	MAH	3000	200	100	Good
S00004	Ashish	A/5 Juhu	Bombay	400044	MAH	3500	200	150	Good

DATA OF SALES_ORDER

S_order_no	S_order_date	Client no	Dely type	Bill yn	Salesman no	Delay date	Orderstatus
O19001	12-Jan-96	1	F	N	50001	20-Jan-96	IP
O19002	25-Jan-96	2	P	N	50002	27-Jan-96	C
O16865	18-Feb-96	3	F	Y	500003	20-Feb-96	F
O19003	03-Apr-96	1	F	Y	500001	07-Apr-96	F
O46866	20-May-96	4	P	N	500002	22-May-96	C
O10008	24-May-96	5	F	N	500004	26-May-96	IP

Data for sale_order_detail

S_order_no	Product_no	Qty_order	Qty_disp	Product_rate
O19001	P00001	4	4	525
O19001	P07965	2	1	8400
O19001	P07885	2	1	5250
O19002	P00001	10	0	525
O46865	P07868	3	3	3150
O46865	P07885	10	10	5250
O19003	P00001	4	4	1050
O19003	P03453	2	2	1050
O46866	P06734	1	1	12000
O46866	P07965	1	0	8400
O10008	P07975	1	0	1050
O10008	P00001	10	5	525

1. Make client_no primary key in client_master.
2. Add new column phone_number inclient_master table.
3. Add not null constraint in product master with columns : description, profit_percent, sellprice, costprice
4. Change size of client_no field in client_master.
5. Add check constraint to product_master such that sellprice is always greater than costprice.
6. Select produc no,description where profit percent is between 20 and 30 both inclusive.

5. To implement join concepts.
   1. Find out the product which has been sold to 'Ivan'.
   2. Find out the product and their quantities that will have to be delivered.
   3. Find out the names of clients who have purchased 'CD DRIVE'
   4. List the product_no and s_order_no of customers having qty ordered less than 5 from the order details table for the product '1.44 floppies'.
   5. Find the product and their quantities for the orders placed by 'Vandan' and 'Ivan'.
   6. Find the products and their quantities for the orders placed by client_no 'C00001"
   7. Find the order_no, Client_noanssalesman_no where a client has been received by more than one salesman.

6. To aggregate data using group function and implement the concept of sub-queries.

Perform following queries based on all 5 tables mentioned above:

1. Print the description and total quantity sold for each product.
2. Find the value of each product sold.
3. Find out the products which have been sold to 'Ivan'.
4. Find the names of clients who have 'CD Drive'.
5. Find the products and their quantities for the orders placed by 'Vandana' and 'Ivan'
6. Select product_no, total_qty_ordered for each product.
7. Display the order number and day on which clients placed their order.
8. Display the month and date when the order must be delivered.

To implement concept of sub-queries.

1. Find the product_no and description of non moving products.
2. Find the customer name, address, city and pincode for the client who has placed order no "019001".
3. Find the client name who have placed order before the month of may 2006.
4. Find out if product "1.44 Drive" is ordered by only client and print the client_no, name to whom it was soled.
5. Find the name of client who have placed orders worth Rs. 10000 or more.
6. Select the orders placed by "Rahul Desai".
7. Select the name of person who are in Mr.Pradeep's department and who have also worked on inventory control system.
8. Select all the clients and the slaesman in the city of Bombay.
9. Select slaesman name in Bombay who has atleast one client located at Bombay.
10. Select the product_no, description, qty_on-hand, cost_price of non_moving items in the product_master tab

7. To implement the concept of views and indexes

1. Create an index on the table client_master, field client_no.
2. Create an index on the sales_order, fields_order_no.
3. Create a composite index on the sales_order_details table for the columns_order_no. and product_no.
4. Create view on salesman_master whose sal_amt is less than 3500.
5. Create a view client_view on client_master and rename the columns as name, add1, add, city, pcode, state respectively.
6. Select the client names from client_view who live in city 'Bombay'
7. Drop the view client_view.

8. To implement concept of PL/SQL

1. WAP in PL/SQL for addition of two numbers.
2. WAP in PL/SQL for addition of 1 to 100 numbers.
3. WAP in PL/SQL to inverse a number, eg. NUMBER 5639 when inverted must be display as output 9365.

9. To implement concept of cursor

Create a explicit cursor which updates the salary of an employee such that,

1. If salary > 10000, then increase the salary by 15%
2. If 5000<salary <10000, then increase the salary by 12%
3. Otherwise, increase the salary by 25%

10.To create procedures and functions and triggers in Oracle.

```
DECLARE
      x VARCHAR2(20);
BEGIN
      SELECT RTRIM(TO_CHAR(SYSDATE, 'DAY'), ' ') INTO x from DUAL;
      IF x = 'SUNDAY' THEN
      RAISE_APPLICATION_ERROR (-20001, 'Transaction is not allowed');
END IF;
END;
```

Solution2. This package has one subprogram; procedure, update_sal.

```
CREATE OR REPLACE PACKAGE emp_pack
      IS
                  PROCEDURE update_sal (eno IN NUMBER);

END emp_pack;
CREATE OR REPLACE BODY emp_pack
      IS
          PROCEDURE update_sal (eno IN NUMBER)
                  IS
                        x  EMP.Empno % type
                        y  EMP. Sal % type
                  BEGIN
                  SELECT empno, sal INTO x, y FROM emp WHERE empno=eno;
                        IF y> 3000 THEN
                              UPDATE EMP SET Sal= sal*1.1 WHERE Empno =eno;
                        ELSIF y between 2000 AND 3000 THEN
                              UPDATE EMP SET Sal= Sal*1.05 WHERE Empno =eno;
                        ELSE
                              UPDATE EMP SET Sal= Sal*1.03 WHERE Empno =eno;
                        END IF;
```

Calling a Package

Calling a package means actually referencing one of its elements. Following is the method for calling an element from a package.

```
DECLARE
      x  VARCHAR2(20);
BEGIN
      emp_pack.update_sal (7633);
END;
```

Total Hours: 30 hours

Mode of Evaluation

CA1	CA2	ESE	Total
12	13		
25		25	50

Course Code: CS206P	Course Name: Operating System Lab	L	T	P	C
		0	0	2	1
Pre-requisite: Basic concept of C language and data structure					
Course Objectives: This course provides hands-on experience with operating system concepts, focusing on UNIX/Linux environments. After this course student will be able to apply the operating system algorithms to solve the real-life problems.					
Course Outcome: After completion of the course, the student will be able to					

1. Apply knowledge of basic UNIX System calls and Shell programming.
2. Implement various CPU scheduling algorithms and deadlock handling techniques.
3. Implement memory management, process synchronization techniques, page replacement techniques, disk scheduling

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	3	-	-	1	1	1	1	2
CO2	3	3	1	2	2	-	-	1	2	2	1	2
CO3	3	3	-	2	2	-	-	1	2	2	1	2

List Of Practical's (Indicative & Not Limited To)

1. Execute various UNIX system calls
2. Write a basic program on VI Editor in Linux.
3. Shell Scripting.
4. Implement FCFS CPU Scheduling Policy
5. Implement SJF CPU Scheduling Policy
6. Implement Priority CPU Scheduling Policy
7. Implement Round Robin CPU Scheduling Policy
8. Implementation of Banker's algorithm
9. Implement the solution for Bounded Buffer (producer-consumer) problem using inter process communication techniques-Semaphores
10. Implement the solutions for Readers-Writers problem using inter process communication technique – Semaphore
11. Implementation First Fit contiguous allocation technique
12. Implementation Best Fit contiguous allocation technique
13. Implementation Worst Fit contiguous allocation technique
14. Implement file storage allocation technique:
    -Contiguous (using array)
15. Implement file storage allocation technique:
    - Linked –list (using linked-list)
    -Indirect allocation (indexing)
16. Comparison of Disk Scheduling Algorithms.
17. Study of hardware and software requirements of different operating systems (UNIX, LINUX, WINDOWS, Android, IOS

Total Hours: 30 hours

Mode of Evaluation

CA1	CA2	ESE	Total
12	13		
25		25	50

Course Code: CS301P	Course Name: Object Oriented Programming using Java Lab	L	T	P	C
		0	0	2	1

Pre-requisite: Basic concept of C language and data structure

1. To familiarize students with the basic and advance Java Programming Language.
2. To learn modern tools to develop java-based web applications.

Course Outcome: After completion of the course, the student will be able to

1. Perform Java oops concepts on an integrated development environment to solve real world problems.
2. Solve problems in context of programming code based on collections and new java features.
3. Develop a solution for case study-based problem using advance java concepts.
4. Design RESTful Web Services with Spring Boot Test using Spring Framework concepts

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	2	2	2	-	-	-	-	-	-	2
CO2	2	3	2	2	2	-	-	-	-	-	-	3

CO3	**2**	**2**	**2**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
CO4	**2**	**3**	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**2**	**3**

List Of Practical's (Indicative & Not Limited To)

1. WAP to insert 3 numbers from the keyboard and find a greater number among 3 numbers.
2. WAP to count the total number of odd numbers between 1-100, and display the sum of them.
3. WAP to Find largest and smallest numbers in an array.
4. WAP to count the number of charcters in a given string, to reverse the string and check whether it is palindrome or not?
5. WAP to find out the sum of command line arguments.
6. WAP to create a class "SimpleObject" and display messages by using the constructor of this class.
7. WAP to create class Number with only one private instance variable as a double primitive type, include the following methods isZero(), isPositive(), isNegative( ), isOdd( ), isEven( ), isPrime(), isAmstrong() in this class and all above methods should return boolean primitive type like for isPositive() should return "Positive = True".
8. WAP to create a Room class, the attributes of this class are roomno, roomtype, roomarea and ACmachine. In this class the member functions are setData and displayData. Use member function to set data and display that data using displayData() method.
9. WAP to create a class named Shape and create three subclasses Circle, Triangle and Square, each class has two-member functions named draw () and erase (). Implement this concept using polymorphism.
10. WAP to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, override the calculateSalary() method to calculate and return the salary based on their specific roles.
11. WAP to implement the concept of method overloading and method overriding. And Abstract Class.
12. WAP to Create a class Employee having members as follows:
    private int empNo
    private String empName
    private int empBasic
    Parameterized constructor to initialize members.
    Getter methods for all instance variables
13. Create a class WriteEmployee having a main method. Ask users to enter details of an employee and set them in an Employee object. Store details of this object in a file emp.txt. Read employee details from the file and display those details.
14. WAP to create a text file in the path c:/Java/abc.txt and check whether that file exists or not. Using the commands exists (), isDirectory(), isFile(), getName() and getAbsolutePath().
15. WAP to Implement three classes: Storage, Counter and Printer.
    The Storage class should store an integer.
    The Counter class should create a thread and start counting from 0 (0,1,2, 3 ...) and store each valuein the Storage class.
    The Printer class should create a thread that keeps reading the value in the Storage class and printing it.
    Write a program that creates an instance of the Storage class and set up a Counter and Printer object to operate on it.
16. WAP to Create a class MyThread derived from Thread class and override the run method. Create a class ThreadDemo having a main method. Create 2 objects of MyThread class and observe the behavior of threads.
17. WAP to Modify the above to create MyThread class by implementing Runnable interface and observe the behavior of threads.
18. WAP to Assign different priorities to the 2 threads and observe the behaviour.
19. WAP to create a producer-consumer scenario using the wait () and notify () methods for thread synchronization.
20. WAP to implement deadlock in Java (Content Beyond Syllabus).
21. WAP to implement the following new features in Java.
    Functional Interface, Lambda Expression: Write a Java program to implement a lambda expression to check if a given string is empty. Method References: Default and Static Method in Interface Inner Class
22. WAP to implement different types of Annotations in JAVA.
23. WAP to filter data by using streams.
24. WAP to Traversing the array elements and to sum the elements using For-each loop.
25. WAP to implement Base64 Encoding and Decoding.
26. WAP to implement Local Variable Type Inference.
27. WAP to implement Sealed Class.
28. WAP to implement Text Blocks and Records.
29. WAP to iterate a linked list in reverse order.
30. WAP to append the specified element to the end of a hash set.

31. WAP to add all the elements of a specified to another tree set.
32. WAP to count the number of key-value (size) mappings in a map.
33. WAP to search for a value in a Tree Map.
34. WAP to Demonstrate Iterator
35. WAP to convert an Iterable to Collection in Java
36. WAP to Create industry-oriented applications using Spring Framework.
37. WAP to test RESTful web services using Spring Boot.

Total Hours: 30 hours

Mode of Evaluation

CA1 12	CA2 13	ESE	Total
25		25	50

Course Code: CS401P	**Course Name: Design & Analysis of Algorithms Lab**	**L**	**T**	**P**	**C**
		0	**0**	**2**	**1**

Pre-requisite: The course requires a background in **mathematics** and **strong programming skills.**

Course Objectives:

1. Implement and analyse Greedy, DP, Backtracking, and Branch & Bound techniques for efficient problem-solving.
2. Evaluate the performance and complexity of sorting, graph, and optimization algorithms through hands-on programming.

Course Outcome: After completion of the course, the student will be able to

1. Implement and analyse Greedy, DP, Backtracking, and Branch & Bound for efficient problem-solving.
2. Evaluate algorithm performance and complexity through hands-on programming in sorting, graphs, and optimization.
3. Develop problem-solving skills by implementing Knapsack, Matrix Chain Multiplication, Graph Colouring, and TSP.
4. Understand NP-Completeness and Approximation Algorithms for tackling computationally hard problems.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	2	-	-	-	-	-	-	2
CO2	3	3	3	3	3	-	-	-	-	-	-	2
CO3	3	3	3	3	3	-	-	-	-	-	-	2
CO4	3	3	3	3	2	-	-	-	-	-	-	2

List Of Practical's (Indicative & Not Limited To)

1. Implement and analyze **Merge Sort** using recurrence relations.
2. Implement **Quick Sort** and **Randomized Quick Sort** and compare efficiency.
3. Implement **Shell Sort** and analyze its time complexity.
4. Implement **Counting Sort, Radix Sort, and Bucket Sort**, and compare their performance.
5. Solve the Fractional Knapsack Problem using the Greedy approach.
6. Implement the Activity Selection Problem using Greedy Strategy.
7. Solve Task Scheduling with Deadline and Penalty using Greedy Strategy.
8. Implement Kruskal's Algorithm to find the Minimum Spanning Tree (MST).
9. Implement Prim's Algorithm to find the MST.
10. Implement 0-1 Knapsack Problem using Dynamic Programming.
11. Solve the Optimal Binary Search Tree (OBST) problem.
12. Implement Matrix Chain Multiplication using Dynamic Programming.
13. Solve the Longest Common Subsequence (LCS) Problem.
14. Implement the All-Pairs Shortest Path Algorithm (Floyd-Warshall).
15. Given a set of coin denominations and a target amount **N**, determine the **minimum number of coins** needed and the **total number of ways** to make **N** using the given denominations.
16. Implement the N-Queens Problem using Backtracking.
17. Solve the Subset Sum Problem using Backtracking.
18. Implement Graph Coloring using Backtracking.
19. Solve the Hamiltonian Cycle Problem using Backtracking.

20. Given an **N×N** chessboard, the Knight's Tour problem requires finding a sequence of moves where a knight visits **every square exactly once** without repetition. The knight moves in an **L-shape** (two squares in one direction and one perpendicular).
21. Implement Traveling Salesman Problem (TSP) using Branch and Bound.
22. Implement the Naïve String Matching Algorithm.
23. Implement Rabin-Karp String Matching Algorithm.
24. Implement Knuth-Morris-Pratt (KMP) String Matching Algorithm.
25. Solve the Vertex Cover Problem using Approximation Algorithms.

Total Hours	**30 hours**

Mode of Evaluation

CA1	CA2	ESE	Total
12	13		
25		25	50

Course Code: IT302P	**Course Name: Computer Networks Lab**	**L**	**T**	**P**	**C**
		0	**0**	**2**	**1**

Pre-requisite: Computer Fundamentals and C programming

Course Objectives:

The objective of this course is to provide students with a theoretical and practical base in computer networks issues and prepare the students for easy transfer from academia into practical life.

Course Outcome: After completion of the course, the student will be able to

1. Understand the fundamental concepts of computer networking and Network topologies.
2. Analyze different types of network devices and simple computer networks.
3. Implement the basic network commands and use techniques, skills, and modern networking tools necessary for engineering practice.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	2	-	-	-	-	-	1	2
CO2	3	2	-	-	2	-	-	-	-	-	1	2
CO3	3	3	2	2	2	-	-	-	-	2	1	2

List Of Practical's (Indicative & Not Limited To)

1. To study of LAN using different Network topologies and different network connecting devices like Hub, Switch, Router etc.
2. To study and learn the handling of networking hardware like: NIC card, RJ-45 connector, crimping tool, CAT-6 cable and Do the following Cabling works in a network (T568A and T568B) a) Cable Crimping b) Standard Cabling and c) Cross Cabling d) IO connector crimping e) Testing the crimped cable using a cable tester.
3. Running and using services/commands like ifconfig, ping, traceroute, nslookup, telnet, ftp, etc.
4. Establish a physical peer to peer network connection using two or more systems using network interface card, network cables, Switch and Router in a LAN.
5. To install the network packet analysis tool Wireshark and also capture and analyze the local traffic.
6. Simulate and Configure a Network topology using packet tracer software.
7. Implement the data link layer framing methods such as character count, character stuffing and bit stuffing.
8. Implement the error correcting code Cyclic Redundancy Check (CRC) of data link layer using various polynomials like CRC-CRC 12, CRC 16 and CRC CCIPP.
9. Implement a program to determine if the IP address is in Class A, B, C, D, or E and translate dotted decimal IP address into 32-bit address.
10. Implement Dijkstra algorithm to compute the shortest path through a given graph.

Total Hours	**30 hours**

Mode of Evaluation

CA1	CA2	ESE	Total

12	13		
25		25	50

Course Code: CS208P	Course Name: Web Technology Lab	L	T	P	C
		0	0	2	1

Pre-requisite: Basic HTML/CSS/JavaScript

Course Objectives:

1. The course aims to provide exposure to the applications of Node.js and its applications.
2. The course aims to train the students on Express.js and React.js for front end development.
3. The course is designed to give the students hands-on experience on integrate the web applications with Mongo DB for real time application development.
4. It aims to train the students with hands on experience on python based Django and Flask frameworks.

Course Outcome: After completion of the course, the student will be able to

1. To understand Node.js and its core modules to develop asynchronous and event-driven server-side applications, demonstrating non-blocking architecture using callbacks, promises, and WebSockets.
2. To apply Express.js to design and implement RESTful APIs with CRUD operations, middleware integration, and templating engines, while testing endpoints using Postman.
3. To apply React.js concepts, including state management (Context API/Redux), component-based development, and frontend-backend integration in a MERN stack application.
4. To understand web application development with python Flask and Django frameworks enabled with CRUD operations and RESTful API integration.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	-	-	-	-	-	-	-	2
CO2	3	3	3	2	-	-	-	-	-	2	-	2
CO3	3	3	3	2	-	-	-	-	-	2	-	2
CO4	2	2	2	-	-	-	-	-	-	-	-	2

List Of Practical's (Indicative & Not Limited To)

1. Install Node.js and npm, set up a new Node.js project, and create a basic "Hello World" program using the HTTP module.
2. Explore built-in Node.js modules (fs, http, os, url). Implement a file-handling systems to read and write to files synchronously and asynchronously.
3. Implement callbacks, promises, and async/await using API calls or file system operations to demonstrate non-blocking architecture.
4. Create an event-driven application using the EventEmitter class and implement WebSockets for real-time communication.
5. Install Express.js, create a simple server, define API routes (GET, POST), and test responses using Postman.
6. Develop a RESTful API with CRUD operations using Express.js and test using Postman. Implement middleware for logging and error handling.
7. Implement dynamic web pages using templating engines like EJS, PUG, or Handlebars. Develop a small web application using one of these engines.
8. Set up a React project using create-react-app, create functional components, pass props, and manage state using useState hook.
9. Implement state management using React Context API or Redux. Build a to-do list application that updates dynamically based on user actions.
10. Develop a full-stack MERN application where the React frontend consumes REST APIs from an Express backend, demonstrating real-time data updates.
11. Install MongoDB, create a database, insert, update, delete, and query documents using the MongoDB shell and Mongoose in Node.js.
12. Build a REST API in Express.js that connects to MongoDB using Mongoose, performing CRUD operations and error handling.
13. Set up a Flask environment, create a basic Flask web application with routing, and display dynamic content using Jinja2 templating.
14. Develop a RESTful API using Flask-RESTful, handle JSON data, and implement request validation. Test API endpoints using Postman.

15. Create a Django web application using the MVC (Model-View-Controller) architecture, define models, connect to a database, and perform CRUD operations.

Total Hours	**30 hours**

Mode of Evaluation

CA1	CA2	ESE	Total
12	13		
25		25	50

Course Code: AI104P	**Course Name: Data Engineering Essentials**	**L**	**T**	**P**	**C**
		0	**0**	**4**	**2**

Pre-requisite: NA

Course Objectives:

1. Introduce key concepts, lifecycle, and tools in data analytics with an emphasis on AWS cloud technologies.
2. Build practical skills in data preprocessing, visualization, and machine learning using Python and AWS services.
3. Explore big data technologies, including Hadoop, Spark, and AWS-based solutions for scalable analytics.

Course Outcome: After completion of the course, the student will be able to

1. Explain key concepts, tools, and applications of data analytics, including AWS services.
2. Apply data cleaning and preprocessing using Python libraries and AWS Glue.
3. Analyze data and visualize insights using BI tools and AWS QuickSight.
4. Implement ML algorithms for classification and clustering with AWS SageMaker.
5. Evaluate big data tools and demonstrate processing with Hadoop and AWS EMR.

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	-		-	-	-	-	-	-	1
CO2	2	3	3	2	2	1	-	-	-	-	1	2
CO3	2	2	3	2	3	1	-	-	2	-	1	2
CO4	2	3	3	3	3	1	-	-	2	-	1	2
CO5	2	3	3	3	2	1			2		1	2

Unit 1	**Introduction to Data Analytics**	**12 hours**

1.Sources and characteristics of data, need for data analytics.
2.Evolution of analytic scalability, analytics lifecycle.
3.Analytic processes, tools, and applications.
4.Analysis vs. reporting, modern data analytic tools.
5. AWS Integration: AWS data services (S3, Redshift, Lake Formation), key roles in AWS analytics project S3: Raw and processed data storage. Redshift: Data warehousing for analytics. Lake Formation: Build and secure data lakes. Mapping 5 V's to AWS services (*Fundamentals of Data Analytics* course).
Hands-on: Set up an AWS account, explore S3 for data storage, create a simple data lake with Lake Formation. Upload sample datasets to S3, analyze metadata with AWS Glue Data Catalog.
Case Study: Design a data analytics pipeline for a retail company using S3, Glue, and Lake Formation.

Unit 2	**Data Collection, Cleaning & Preprocessing**	**12 hours**

1.Data types and sources (structured, unstructured, semi-structured).
2.Handling missing data, outliers, noise, transformation, normalization.
3.Data wrangling with Python (Pandas, NumPy).
4.Introduction to ETL (Extract, Transform, Load) pipelines.
5.AWS Integration: AWS Glue for ETL (Extract, Transform, Load), S3 for data storage, AWS Lambda for preprocessing automation. Batch processing with Amazon EMR
Handson- Data wrangling with Python (Pandas, NumPy) on sample datasets. Build an ETL pipeline with AWS. Use AWS Lambda to automate data preprocessing tasks (e.g., trigger cleaning on new S3 uploads)..
Mini-Project: Preprocess a real-world dataset (e.g., customer reviews) and store it in S3.

Unit 3	**Exploratory Data Analysis & Visualization**	**12 hours**

1.Statistics and Probability: Distributions, mean, median, mode, standard deviation; confidence intervals, regression, hypothesis testing; random variables, Bayes' Theorem.
2.Data Visualization: Power BI and Tableau overview, dynamic reports, dashboards, case studies.

3.AWS Integration: AWS QuickSight for visualization, integrating with S3 and Redshift for interactive insights .Concepts from Visualizing with QuickSight.
Hands-on: Create dashboards in Power BI/Tableau using sample data.
Build interactive visualizations in QuickSight, connecting to S3 and Redshift.
Perform EDA on a dataset (e.g., sales data) to identify trends and outliers.
Practice Visualizing with QuickSight labs
Case Study: Visualize business metrics (e.g., revenue by region) using QuickSight dashboards.

Unit 4	**Machine Learning for Data Analytics**	**12 hours**

1.Supervised learning (Decision Trees, k-NN, Naive Bayes), unsupervised learning (K-Means, Hierarchical Clustering).
2.Dimensionality reduction (PCA), association rule mining (Apriori).
3.Model evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1-Score.
4.AWS Integration: AWS SageMaker for building, training, and deploying ML models; SageMaker JumpStart for pre-built algorithms. JumpStart: Pre-trained models for quick deployment.
Hands-on:Implement classification (e.g., Decision Trees) and clustering (e.g., K-Means) in Python (scikit-learn).
Train and deploy a classification model (e.g., customer churn) using SageMaker.
Evaluate model performance with SageMaker's built-in metrics.
Mini-Project: Build a recommendation system using SageMaker and deploy it as an endpoint.

Unit 5	**Big Data and Advanced Analytics**	**12 hours**

1.Introduction to big data, characteristics, challenges, real-life examples.
2.Hadoop: Components (HDFS, MapReduce, YARN), ecosystem (Pig, Hive, HBase, Sqoop).
3.Apache Spark overview, MapReduce paradigm (e.g., matrix-vector multiplication).
4.AWS Integration: AWS EMR (Elastic MapReduce) for Hadoop/Spark, Redshift for data warehousing, Kinesis for real-time data streams. Amazon OpenSearch Service: Log analytics and monitoring (Getting Started with Amazon OpenSearch Service). Streaming analytics with Amazon MSK (Building Streaming Data Analytics Solutions on AWS).
Hands-on:Run a MapReduce job on Hadoop (e.g., word count).
Process large datasets with AWS EMR (Hadoop/Spark), query with Redshift.
Analyze real-time data streams using AWS Kinesis (e.g., clickstream data).
Explore Amazon OpenSearch Service for log analytics (Getting Started with Amazon OpenSearch Service).
Practice labs from Building Streaming Data Analytics Solutions on AWS.
Mini-Project: Build a big data pipeline—ingest with Kinesis, process with EMR, store in Redshift.
AWS DATA ENGINEER CERTIFICATION
Exam Readiness: AWS DATA ENGINEER ASSOCIATE CERTIFICATION

Total Lecture Hours	**60 hours**

Textbook:

1. Python for Data Analysis (3rd Edition) – Wes McKinney, O'Reilly Media, 2022
2. An Introduction to Statistical Learning (2nd Edition) – Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 2021
3. Data Mining: Concepts and Techniques (4th Edition) – Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, 2022

Reference Books:

1. Kumar, M. (2021). Building Data Streaming Applications with Apache Kafka: Design, develop and stream data applications using Apache Kafka connectors, Kafka Streams API and Kafka SQL. Birmingham, UK: Packt Publishing. ISBN: 9781801073632
2. Crickard, P. (2020).Data Engineering with Python: Work with massive datasets to design data models and automate data pipelines using Python. Birmingham, UK: Packt Publishing. ISBN: 9781800565794

Mode of Evaluation

CA1 25	CA2 25	ESE	Total
50		50	100