1.1 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)	SDG Mapping	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)	SDG-4,8,9,12	CSE	CS401L	Design and Analysis of Algorithms	L	3	0	0	60	15	75	75	150	3
2	PC	Major (Core)	SDG-4,8,9,12,16	CSE	CS206L	Operating System	L	3	0	0	60	15	75	75	150	3
3	MC	Value Added	SDG-3,4,16	ASH	HS112L	Universal Human Values	L	3	0	0	60	15	75	75	150	3
4	HS	AEC	SDG-4,8,10	ASH	HS113L	Aptitude-2	L	1	0	0	-	25	25	25	50	1
5	HS	AEC	SDG-4,8,10	ASH	HS114L	Soft Skills Essential-2	L	1	0	0	-	25	25	25	-	NC
Blended																
6	PC	Major (core)	SDG-4,9,11	CSE(AIML) & CSE(AI)	AI308B	AI Driven Full Stack Development	B	2	0	2	60	15	75	75	150	3
7	PC	Major (core)	SDG-4,9,11	CSE(AIML) & CSE(AI)	AI104B	Data Engineering Essentials	B	2	0	2	60	15	75	75	150	3
8	PC	Major (core)	SDG-4,9,11	CSE(AIML) & CSE(AI)	AI311B	Deep learning Essentials	B	2	0	2	60	15	75	75	150	3
9	PE	Major (Core)/SEC	-	-	-	Professional Elective-II	B	3	0	2	80	20	100	100	200	4
Lab/Practical																
10	PC	Major (core)	SDG-4,8,9,12	CSE	CS401P	Design and Analysis of Algorithms Lab	P	0	0	2	-	25	25	25	50	1
11	PC	Major (Core)	SDG-4,8,9,12,16	CSE	CS206P	Operating System Lab	P	0	0	2	-	25	25	25	50	1
Total Hours : 32 hrs.								**20**	**0**	**12**					**1250**	**25**

Professional Electives (PE)

Course Type (PE)	Artificial Intelligence and Machine Learning	Cloud Ops and Security	Data Engineering (Powered by AWS)
SDG Mapping	**SDG-4,9**	**SDG-4,9**	**SDG-4,9**
BOS	**CSE(AI) & CSE(AIML)**	**CSE(AI) & CSE(AIML)**	**CSE(AI) & CSE(AIML)**
PE I-(3thSem)	Cloud Foundations (AI103E)	Cloud Foundations (AI103E)	Cloud Foundations (AI103E)
PE II-(4thSem)	Introduction to Computer Vision (AI309E)	AWS Cloud Operations (AI310E)	AWS Data Engineering (AI307E)

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 using 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 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: 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: 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 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: AI103E	Course Name: Cloud Foundations	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.

1. 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: 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: AI308B	**Course Name: AI Driven Full Stack Development**	**L**	**T**	**P**	**C**
		2	**0**	**2**	**3**

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

Course Objectives:

1. Introduce the core concepts of Node.js, the MERN stack, and server-side development using asynchronous programming and event-driven architecture.
2. Develop the ability to build backend services using Express.js and MongoDB, including routing, middleware, validation, and database operations.
3. Enable students to design interactive frontend applications using React components, hooks, state management, and API integration.
4. Familiarize students with full-stack application deployment using cloud platforms, environment configuration, and production best practices.
5. Expose learners to the integration of AI-powered features such as semantic search, chatbots, recommendation systems, and automated content generation within web applications.

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

1. Apply Node.js concepts including asynchronous programming, event loop, and core modules to develop server-side applications.

2. Build RESTful services using Express.js and MongoDB with routing, middleware, validation, and data modeling.
3. Develop interactive frontend applications using React components, hooks, state management, and API integration.
4. Deploy production-ready MERN applications on cloud platforms using environment configuration and CI/CD workflows.
5. Integrate AI capabilities into web applications using tools for semantic search, chatbots, recommendation systems and content generation

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

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

Unit 1	**Node.js Fundamentals**	**14 hours**

Introduction to Internet & WWW, Client–Server Architecture & HTTP/HTTPS, Overview of MERN Stack (MongoDB, Express, React, node.js), Introduction to Node.js, V8 Engine, Node Installation, Git and GitHub, blocking vs Non-Blocking Architecture, npm, nvm, package. json, Asynchronous Programming: Callbacks, Promises, Async/Await, Event Loop, Node Core Modules: fs, http, path, url, os, crypto.

Unit 2	**Express.js and MongoDB**	**14 hours**

Introduction, Features, Project Structure, Routing: GET, POST, PUT, DELETE, PATCH, Middleware: Built-in, Custom, Third-party (cors), Error Handling in Express ,API Testing using Postman or Thunder Client, Introduction to MongoDB: Data Types, BSON, Collections & Documents, Installation (Local & MongoDB Atlas), CRUD Operations in MongoDB, Sorting, Filtering, Indexing, Mongoose ODM: Schema, Models, Validation, Mongoose Middleware (pre/post hooks).

Unit 3	**React.js**	**14 hours**

Introduction to Modern React (Vite Setup), Components, JSX, Functional Components, State vs Props, Hooks: useState, useEffect, use Context, use Reducer, React Router, Component Design & Optimization, Connecting React Frontend with Express Backend (axios),Environment Variables & Config Management, Security in MERN:CORS ,Password Hashing (bcrypt),RESTful Services: Introduction to REST Architecture, HTTP Methods & Status Codes, API Versioning, Pagination & Filtering, Authentication & Authorization, JSON Web Tokens (JWT), Access & Refresh Tokens.

Unit 4	**Deployment Using Render**	**9 hours**

Preparing Express.js for Production, Preparing React App for Deployment, Environment Variables in Production, Deploying Express.js Backend on Render: Creating Render Web Service, Connecting GitHub Repository, Build & Start Commands, Setting Environment Variables, Connecting Backend to MongoDB Atlas, Deploying React Frontend on Render, Updating API Base URLs to Render Domain, Final Integration Testing of Live MERN App.

Unit 5	**Integrating AI with Web Projects**	**9 hours**

Introduction to AI for Web Applications, Basics of Semantic Search and Implementing Simple AI Search, Building Basic Chatbots Using OpenAI/Gemini APIs, Introduction to Recommendation Systems and Simple Recommendation Implementation, Automated Content Generation & AI-Based SEO (Meta Tags, Keywords).

	Total Lecture Hours	**60 hours**

Textbook:

1. Beginning MERN Stack, Greg Lim (First edition).
2. Pro MERN Stack, Vasan Subramanian (Second Edition).

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 (First Edition).

Mode of Evaluation

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

Course Code: AI104B	Course Name: Data Engineering Essentials	L	T	P	C
		2	0	2	3

Pre-requisite: NA

Course Objectives:

1. Master the complete data analytics lifecycle with hands-on data cleaning, ETL pipeline building, advanced EDA, and interactive dashboard creation using Python, Power BI, and Tableau.
2. Gain expertise in time-series analysis & forecasting by building and comparing ARIMA, SARIMA, LSTM, Prophet, and GARCH models with proper stationarity testing, back testing, and statistical evaluation.
3. Design and deploy production-grade big data & cloud pipelines using Hadoop ecosystem, PySpark, and AWS services (Glue, EMR, S3, Athena, Hive) on multi-TB datasets and prepare for AWS DEA-C01 certification

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

1. Perform data cleaning, feature engineering and build reusable Python-based ETL pipelines on real-world messy datasets
2. Apply statistical methods and create advanced interactive dashboards using Power BI & Tableau for business insights.
3. Develop, compare and evaluate multiple time-series forecasting models (ARIMA, LSTM, Prophet, GARCH) with proper statistical tests.
4. Implement big data processing using Hadoop ecosystem and PySpark on multi-TB datasets (Wikipedia, NYC Taxi, etc.).
5. Build production-grade data engineering pipelines on AWS (Glue, EMR, S3, Hive, Athena) aligned with DEA-C01 certification.

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

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

Unit 1	Introduction to Data Analytics	12 hours

1.Sources, Types & Characteristics of Data + Need for Analytics

Structured, semi-structured, unstructured data; sources (databases, APIs, logs, sensors, web, social media); 5 V's of data; analytics vs reporting; business impact of clean data.

2.Evolution of Analytics & Analytics Lifecycle

From traditional BI → big data → modern analytics; full analytics lifecycle (collection → preparation → analysis → insight → action).

3.Data Cleaning & Preprocessing

Handling missing values (imputation/deletion), outlier detection & treatment, noise reduction, data transformation (log, scaling), normalization vs standardization, encoding categorical variables.

4.Data Wrangling with Python

Pandas (filtering, groupby, merge, pivot, melt, handling duplicates), NumPy basics, reshaping & combining multiple datasets.

5.ETL Concepts & Pipeline Design

Extract → Transform → Load workflow, batch vs real-time concepts, building reusable cleaning + transformation pipelines in pure Python.

Hands-on Labs

Load & explore 3 raw datasets (CSV + JSON + messy Excel) → summary stats + 5 V's mapping
Complete cleaning: missing values + outliers + encoding + normalization (one notebook)
Data wrangling: merge customer + orders + product data → create final analysis-ready table
Build a reusable Python ETL script/function (input folder → clean → output Parquet/CSV)

Mini-Project

Take any real-world messy dataset (e.g., Kaggle Amazon reviews / NYC taxi / airline delay / IMDb movies):
→ Build and run a complete Python ETL pipeline (raw → bronze → silver → gold layers)
→ Submit Jupyter notebook + clean final dataset + 2-page report + 3-min video walkthrough (GitHub link mandatory)

Unit 2	Statistics, Probability & Advanced EDA (Power BI + Tableau)	12 hours

Core Statistics & Probability

Random variables, key distributions, mean/median/mode, variance, skewness, confidence intervals, hypothesis testing, p-value, simple regression, MLE basics, Bayes' Theorem

Power BI – Advanced EDA & Statistical Storytelling

Power Query (full data cleaning & shaping), DAX deep dive (time intelligence, statistical measures, running totals, YoY/MoM, dynamic banding), Key Influencers, Decomposition Tree, AI visuals, What-If parameters, Forecast sheet, anomaly detection, R/Python visuals integration

Tableau – Advanced EDA & Statistical Analysis

Calculated fields & table calculations, LOD expressions (FIXED, INCLUDE, EXCLUDE), dynamic sets & parameters, clustering, built-in forecasting, trend lines & statistical tests, cohort & retention analysis, dashboard actions, storytelling with data

Hands-on Labs

Stats basics + hypothesis testing + CI on real dataset (Python notebook)
Power BI only: Full retail EDA → 15+ DAX measures + Key Influencers + Decomposition Tree + dynamic anomaly detection dashboard
Power BI only: Executive KPI dashboard with Forecast sheet + What-If + R script visual for outlier detection
Tableau only: Same dataset → LOD cohort + clustering + forecasting + dashboard actions
Tableau only: Advanced dashboard with sets, parameters, trend significance tests

Mini-Project

A – Power BI Specialist
Build complete "Sales & Profitability Insights" dashboard (must use Key Influencers, Decomposition Tree, Forecast sheet, 10+ complex DAX) + statistical summary page
B – Tableau Specialist
Build same insights using LOD, clustering, forecasting, actions, sets + statistical significance tests
Deliverables: .pbix OR .twbx file + 3-min video walkthrough + GitHub repo + 2-page report on insights found (confidence intervals, test results, anomalies)

Unit 3	Time Series Analysis & Forecasting	12 hours

Stationarity & Non-Stationarity Removal

Trend, seasonality, cycles; classical decomposition (additive/multiplicative); differencing & seasonal differencing; unit root tests (ADF, KPSS, PP); ACF/PACF interpretation.

Core Time Series Modelling

Moving average smoothing, ARIMA(p,d,q), SARIMA(p,d,q)(P,D,Q)s, ARIMAX (exogenous variables), intervention & outlier handling, LSTM/GRU basics for time series.

Advanced Modelling & Evaluation

State-space models & Kalman filter (trend + cycle + seasonal), GARCH/EGARCH for volatility, VAR, Granger causality, cointegration & VECM; forecast metrics (MAE, RMSE, MAPE, sMAPE, MASE), Diebold-Mariano test, back testing, rolling-window evaluation.

Hands-on Labs

Stationarity + decomposition + ADF/KPSS + make 3 real datasets stationary
Auto-ARIMA vs manual SARIMA/ARIMAX + LSTM forecasting + error comparison
GARCH on stock returns + VAR/VECM on macro data + Kalman filter (Unobserved Components)
Full back testing framework + Diebold-Mariano test in Python

Mini-Project

Build & compare an end-to-end forecasting system on a real dataset (retail sales / energy demand / stock price):
→ Full stationarity treatment → SARIMA + ARIMAX + LSTM + GARCH volatility → ensemble forecast → backtest + statistical tests → final Jupyter notebook + detailed report + 3-min video (GitHub link mandatory

Unit 4	Big Data Fundamentals & Distributed Processing	12 hours

Introduction to Big Data

5 V's, challenges (volume, velocity, variety, veracity), real-world use cases, evolution from traditional RDBMS to big data

Hadoop Ecosystem

HDFS architecture, data replication, NameNode/DataNode
MapReduce programming model (word count, matrix multiplication examples)
YARN resource management

Hadoop Ecosystem Tools

Pig (scripting), Hive (SQL on Hadoop), HBase (NoSQL column store), Sqoop (RDBMS ↔ Hadoop transfer), Oozie (workflow)

Apache Spark (Core + SQL)

Spark vs MapReduce, RDD fundamentals, DataFrame/Dataset API, Spark SQL, structured streaming basics, catalyst optimizer, Tungsten

Hands-on Labs

Set up single-node Hadoop → run classic MapReduce word count & log analysis
Hive: create external tables → query 1 GB+ dataset with HiveQL
Pig: data cleaning + multi-step pipeline on same dataset
Spark: convert MapReduce job to PySpark → 10x speed gain demo

Mini-Project

Build a complete open-source big data pipeline on a real large dataset (e.g., Wikipedia pageviews / NYC taxi / Twitter archive):
→ Ingest with Sqoop OR Flume → store in HDFS → clean with Pig → transform & analyze with Spark SQL → save results in Hive/Parquet → visualize top insights
Deliverables: GitHub repo + all scripts (Pig, PySpark, HiveQL) + 3-min demo video + report (submit as zip)

Unit 5	**AWS Data Engineering Basics (Beginner Level – DEA-C01 Prep)**	**12 hours**

Introduction to AWS Data Engineering
Core concepts: 5 V's of big data, data lifecycle (ingest, store, process, analyze), AWS Well-Architected Framework basics, common use cases (ETL, streaming, warehousing).

Data Ingestion & Storage Fundamentals
Batch ingestion (S3 basics: buckets, partitioning, lifecycle policies), real-time streams (Kinesis Data Streams/Firehose intro), databases (DynamoDB NoSQL, RDS relational), data formats (CSV, JSON, Parquet, Avro).

Data Processing & Transformation Basics
ETL with Glue (crawlers, jobs, PySpark intro), serverless compute (Lambda for simple transforms), batch processing (EMR with Hadoop/Spark overview), schema design (star/snowflake)

Data Security, Governance & Monitoring
IAM roles/policies for data access, encryption (S3 SSE, KMS), data quality (Glue Data Catalog, Lake Formation basics), monitoring (CloudWatch metrics/alarms, logging with CloudTrail).

Hands-on Labs
Create S3 bucket → upload/sample data → set lifecycle policy (AWS Console)
Build simple Glue crawler/job on S3 data → transform to Parquet
Set up basic Kinesis stream → simulate data ingest → view in Console
Configure IAM role for Glue → enable CloudWatch logging on a job

Mini-Project
Design a basic ETL pipeline for retail sales data:
→ Ingest CSV to S3 → clean/transform with Glue job → store in partitioned Parquet → add IAM security + CloudWatch monitoring → document architecture + 2-min video demo (Console screenshots + GitHub link)

	Total Lecture Hours	**60 hours**

Textbook:

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.
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.

Reference Books:

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

Mode of Evaluation

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

Course Code: AI311B	Course Name: Deep learning Essentials	L	T	P	C
		2	0	2	3

Pre-requisite: Introduction to AI, Python, Machine Learning Essentials

Course Objectives:

1. Introduce the core concepts of deep learning, including neural networks, optimization and regularization.
2. Develop skills to build and evaluate CNNs, RNNs and transformer-based models for vision and NLP tasks.
3. Familiarize students with generative models and graph neural networks for advanced deep learning applications.
4. Enable students to deploy deep learning models with basic explainability and ethical understanding.

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

1. Explain the core concepts of deep learning, multilayer perceptrons, activation functions, forward/backward propagation and regularization.
2. Build and optimize convolutional neural network models and apply transfer learning for image classification tasks.
3. Implement RNN, LSTM, GRU and sequence-to-sequence models and compare their performance on sequence learning tasks.
4. Fine-tune transformer models and deploy deep learning applications.
5. Develop and evaluate generative models such as Autoencoders, VAEs, GANs and apply GNNs for simple graph 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	
CO1	3	3	2	1	1	1	-	1	3	1	3	
CO2	3	3	3	3	3	3	2	3	3	1	3	
CO3	3	3	3	3	3	3	2	3	3	1	3	
CO4	3	3	2	3	3	3	2	3	3	1	3	
CO5	3	2	3	3	3	3	2	3	3	1	3	

Unit 1 **Foundations of Deep Learning** **12 hours**

Introduction to Deep Learning & Neural Networks, Perceptron and Multilayer Perceptron (MLP), Activation Functions (Sigmoid, ReLU, GELU), Forward & Backward Propagation (Mathematical Derivation), Loss Functions & Optimization Basics (SGD), Overfitting & Regularization Concepts

Case Studies

1. Handwritten Digit Recognition (MNIST) – from-scratch MLP implementation using NumPy
2. Predicting Diabetes Onset – using tabular data (Kaggle dataset) with basic neural network

Practical

1. Introduction to Deep Learning Libraries (Tensorflow, Keras and PyTorch)
2. Implement MLP from scratch using NumPy
3. Visualize activation and loss landscapes.

Unit 2 **Optimization of Neural Networks** **12 hours**

Deep Optimization (Momentum, RMSProp, Adam), Weight Initialization & Batch Normalization, Dropout & Data Augmentation, Convolutional Operations (Padding, Stride, Receptive Fields), Base CNN Architectures (LeNet), Transfer Learning and Fine-Tuning.

Case Studies

1. CIFAR 10 dataset with LeNet Architecture.
2. Brain Tumor Detection from MRI Images – transfer learning using pre-trained CNN

Practical

1. Implement CNN on CIFAR-10
2. Fine-tune a pretrained LeNet model
3. Visualize learned filters and class activation maps

Unit 3 **Sequence Modeling** **12 hours**

RNNs, Vanishing/Exploding Gradient Problem, LSTM and GRU Networks, Sequence-to-Sequence.

Case Studies

1. Sentiment Classification (IMDB Dataset) – LSTM model
2. Question Answering using BERT (SQuAD Dataset) – Hugging Face fine-tuning

Practical

1. Compare RNN vs LSTM vs GRU performance
2. Build and train LSTM for text classification
3. Visualize attention heatmaps in Seq2Seq

Unit 4	**Transformers**	**12 hours**

Encoder, Decoder Models, Attention Mechanism (Dot-Product, Multi-Head), Transformer Architecture (Encoder/Decoder), Pretraining & Fine-tuning (BERT, GPT overview)

Case Studies

1. Question Answering using BERT (SQuAD Dataset) – Hugging Face fine-tuning
2. Multi-class text classification using DistilBERT
3. Mini RAG project for custom PDF documents

Practical

1. Fine-tune BERT model for QA/Sentiment tasks
2. Visualize attention weights

Unit 5	**Generative and Graph Models**	**12 hours**

Autoencoders & Variational Autoencoders (VAE), Generative Adversarial Networks (GANs): Theory and Losses, Conditional GANs & Diffusion Models (Overview), Graph Neural Networks, Knowledge Graphs.

Case Studies

1. WIT dataset Autoencoder – Image Denoising
2. Simple GAN for Digit Generation
3. VAE Latent Space Visualization – Digit Separation

Practical

1. Build VAE on WIT dataset and visualize latent space
2. Train GAN for image synthesis
3. Implement GCN on citation data

Total Lecture Hours	**60 hours**

Textbook:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning (MIT Press)
2. Stanford CS231n – Convolutional Neural Networks for Visual Recognition
3. Stanford CS224n – NLP with Deep Learning

Reference Books:

1. MIT 6.S191 – Introduction to Deep Learning
2. Deep Learning Specialization – Andrew Ng (Coursera)

Mode of Evaluation

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

Course Code:AI309E	**Course Name: Introduction to Computer Vision**	**L**	**T**	**P**	**C**
		3	**0**	**2**	**4**

Pre-requisite: Introduction to AI, Python, Machine Learning Essentials, Deep Learning

Course Objectives:

1. Introduce the fundamental principles of image formation, representation, processing and analysis.
2. Develop skills to apply OpenCV and classical vision techniques for feature extraction, segmentation and image manipulation.
3. Enable students to use machine learning and deep learning methods for visual recognition, detection and segmentation tasks.
4. Familiarize students with advanced topics such as vision transformers, 3D vision and multimodal AI, along with responsible and ethical use of vision systems.

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

1. Explain fundamental concepts of image formation, processing, and analysis.
2. Apply OpenCV tools for image manipulation and feature extraction.
3. Use machine learning algorithms for visual recognition tasks.

4. Implement deep learning architectures for image classification, detection, and segmentation.
5. Explore emerging topics such as vision transformers, 3D visions, and multimodal AI.

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

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

Unit 1	**Foundations of Computer Vision**	**15 hours**

Overview of Computer Vision and Applications (Robotics, Medical, Surveillance, AR/VR). Human Vision vs Computer Vision. Image formation (Cameras, Lenses, Perspective projection, and Pinhole model). Colour models (RGB, HSV, LAB, and Grayscale). Digital image representation (Pixels, Sampling, Quantization, Histograms). Basic operations like Image reading, writing, resizing, and transformations (Geometric Transformations, Fourier Transform, Intensity Transformation).

Case Study 1: Smart Traffic Monitoring System using Cameras

Unit 2	**Image Processing using OpenCV**	**15 hours**

Image enhancement (Histogram Equalization, Contrast Enhancement), smoothing, sharpening, thresholding and colour correction. Image augmentation (Flipping, Rotation, Scaling, Noise Addition). Noise Reduction Techniques (Median Filtering, Bilateral Filtering and Wavelet Denoising). Edge detection (Sobel, Laplacian, Canny). Morphological operations (Erosion, Dilation, Opening, Closing, Morphological gradient). Image segmentation (Region Growing, Contour Detection, Watershed). Geometric transformations (Rotation, Scaling, Affine, Perspective). Corner Detection (Harris Corner), Feature Descriptors, SIFT, SURF, ORB.

Case Study 2: Medical X-Ray Image Enhancement System

Unit 3	**Machine Learning for Computer Vision**	**15 hours**

Introduction to ML in Vision, Role of machine learning in CV, Classical vs Deep learning approaches. Scikit-image library and Evaluation metrics (IoU, mAP). Feature Engineering (Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Gabor filters, Colour histogram and Texture features). Dimensionality Reduction (Principal Component Analysis), t-SNE / UMAP for visualization. Supervised Learning (KNN, SVM, DTand RF, LR) Unsupervised Learning (k-Means, Hierarchical clustering, Gaussian Mixture Models). Object Detection (Sliding Window approach, Feature-based Detection (HOG + SVM), Viola Jones method for face detection).

Case Study 3: Face Recognition using HOG + SVM

Unit 4	**Deep Learning for Computer Vision**	**15 hours**

Introduction to Keras, TensorFlow and PyTorch Libraries. Introduction to Neural Networks and CNNs (Convolution, Pooling, Activation, and Normalization). Popular CNN architectures (AlexNet, VGG, ResNet), Transfer learning and Fine-tuning. Object detection (R-CNN, Fast R-CNN, Faster R-CNN, YOLO). Image segmentation (FCN, U-Net, Mask R-CNN). Handling Sequential Data (Videos, Frame Sequences, and Temporal Features), Recurrent Neural Networks (RNNs). Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).

Case Study 4: YOLO-based Helmet Detection for Road Safety

Unit 5	**Advanced Deep Learning for Computer Vision**	**15 hours**

Introduction to 3D-CNN, GCN, Vision Transformer, Swin Transformer, DETR, TimeSformer / Video Swin Transformer, Autoencoders, GANs, Diffusion models (DALL·E, Stable Diffusion), Multimodal Vision (Image Captioning, Visual Question Answering) and Ethics & Responsible AI in Vision.

Case Study 5: Image Captioning using Vision Transformers + LSTM

Total Lecture Hours	**75 hours**

Textbook:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning (MIT Press).
2. Computer Vision: Algorithms and Applications by Richard Szeliski.

References:

1. Deep Learning for Vision Systems by Mohamed Elgendy.
2. Deep Learning with PyTorch by Stevens & Antiga

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

Course Code:AI310E	Course Name: AWS Cloud Operations	L	T	P	C
		3	0	2	4

Pre-requisite: AWS Cloud Foundations; Co-requisite: Operating Systems

Course Objectives:

This course aims to equip students with the skills required to design, deploy, operate, and troubleshoot secure, scalable, and cost-effective solutions on AWS. It builds practical knowledge of core AWS services, automation, monitoring, and best practices, aligning learners with the competencies and preparing them for real-world cloud architecture and operations roles.

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

1. Understand AWS cloud operations, core services, IAM, and automation tools.
2. Apply AWS compute, load balancing, auto scaling, and DNS services.
3. Apply serverless, container, and database services for workloads.
4. Analyze AWS networking, storage, and monitoring solutions.
5. Analyze cost management and infrastructure-as-code deployments on AWS.

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

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

Unit 1 | **AWS Cloud Operations Fundamentals and Automation** | **15 hours**

Overview of AWS Cloud Operations, AWS Global Infrastructure, Regions and Edge Locations, Systems Operations in the Cloud, Core AWS Services, AWS Identity and Access Management (IAM), IAM Policy Simulator, AWS Command Line Interface (CLI), AWS Systems Manager, Administration and Development Tools.

Labs / Activities:

Install and Practice Using the AWS CLI, Using AWS Systems Manager, Create a Static Website on Amazon S3.

Unit 2 | **AWS Compute Services and Scalability** | **15 hours**

Amazon EC2, EC2 Instance types, storage options and lifecycle, securing and managing EC2 instances, Amazon EC2 purchasing options, Elastic Load Balancing (ELB), Amazon EC2 Auto Scaling, Configure EC2 Auto Scaling, Amazon Route 53, Amazon Route 53 Geo Routing.

Labs / Activities:

Creating Amazon EC2 Instances, Troubleshooting Create Instance, Using AutoScaling, Route 53 Failover Routing

Unit 3 | **AWS Containers, Serverless, and Database Services** | **15 hours**

AWS Lambda and serverless computing, APIs and REST, Amazon API Gateway, Containers on AWS, AWS Step Functions, AWS Database Services overview, Amazon RDS, Amazon Aurora, Amazon DynamoDB, AWS Database Migration Service (DMS).

Labs / Activities:

Create an AWS Lambda Function, Migrate to Amazon RDS.		
Unit 4	**AWS Networking, Storage, and Monitoring**	**15 hours**
AWS Cloud networking, Amazon VPC, Amazon VPC Connectivity options, Securing and Troubleshooting networks on AWS, AWS storage services including EBS, EFS, S3 and S3 Glacier, AWS Data Transfer and Migration Services, Monitoring with Amazon CloudWatch, AWS CloudTrail and AWS Config, AWS Service Integration with Amazon Athena. **Labs / Activities:** **Configuring Virtual Private Cloud (VPC), Troubleshoot a VPC, Managing Storage, Work with Amazon S3, Monitoring Your Applications and Infrastructure, Working with AWS CloudTrail**		
Unit 5	**AWS Resource Management and Cloud Architecting**	**15 hours**
AWS resource tagging and cost management, AWS Trusted Advisor, Configuration Management in Cloud, Amazon Machine Images (AMIs) and launch templates, Infrastructure as Code (IaC), Introduction to JSON and YAML, AWS CloudFormation, Automating deployments with AWS CloudFormation, Troubleshooting CloudFormation. **Labs / Activities:** **Managing Resources, Optimize AWS Resource Utilization, Automating Deployments with AWS CloudFormation, Troubleshooting CloudFormation Deployments**.		
	Total Lecture Hours	**75 hours**

Textbook:

1. Piper, B., & Clinton, D. (2022). AWS Certified Solutions Architect study guide with 900 practice test questions: Associate (SAA-C03) exam (4th ed.). Wiley.
2. Mathieu, L. (2024). Mastering AWS Security: Create and Maintain a Secure Cloud Ecosystem. Packt Publishing.

References:

1. https://awsacademy.instructure.com/courses/
2. https://www.youtube.com/watch?v=RLd_XTyt-w8

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

Course Code:AI307E	**Course Name: AWS Data Engineering**	**L**	**T**	**P**	**C**
		3	**0**	**2**	**4**

Pre-requisite: NA

Course Objectives:

1. Introduce key concepts, domains, and AWS services for data engineering, aligned with the AWS Certified Data Engineer - Associate (DEA-C01) exam domains:
2. Build practical skills in data ingestion, transformation, orchestration, and pipeline implementation using Python and core AWS data services.
3. Explore data modeling, lifecycle management, quality assurance, monitoring, and security best practices with AWS tools for scalable data engineering.

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

1. Explain core AWS data engineering concepts, services, and exam domains for the AWS Certified Data Engineer - Associate (DEA-C01).
2. Apply data ingestion and transformation techniques using AWS Glue, Kinesis, and Python libraries.
3. Design and manage data stores, including relational/non-relational databases and data warehouses with Amazon RDS, DynamoDB, and Redshift.
4. Implement data operations, orchestration, and support using AWS Step Functions, Lambda, and monitoring tools like CloudWatch.
5. Evaluate data security, governance, and quality controls with AWS IAM, Lake Formation, and best practices for compliance.

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

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

Unit 1	**Data Ingestion and Transformation**	**15 hours**

Data ingestion considerations: Batch vs. streaming, structured, unstructured, semi-structured, Transformation processes, Scalability challenges, Transformation processes

ETL vs. ELT pipelines: Data cleansing, enrichment, schema evolution

AWS services for ingestion: Amazon Kinesis (Data Streams, Firehose, Analytics), AWS DataSync, Amazon AppFlow (SaaS integration)

Transformation tools: AWS Glue (Crawlers, ETL Jobs, DataBrew), Amazon EMR (Spark/Hadoop processing)

Programming concepts: SQL for querying, Python/Scala for scripting, Data formats: JSON, Parquet, Avro

Hands-on: Set up AWS account, Ingest streaming data with Kinesis Firehose → S3, Create Glue ETL job for transformation, Explore Glue Data Catalog

Case Study: Design an ingestion pipeline for e-commerce clickstream data using Kinesis → Glue → S3

Unit 2	**Data Store Management**	**15 hours**

Data store types: Relational (RDBMS), NoSQL, Data Lakes, Warehouses, Partitioning, indexing, compression

Data modeling: Star/Snowflake schemas, Normalization vs. denormalization, Time-series data handling

AWS storage services: Amazon S3 (storage classes, lifecycle policies), Amazon Redshift (distribution keys, sort keys), Amazon DynamoDB (provisioned vs. on-demand)

Database services: Amazon RDS / Aurora, DocumentDB, ElastiCache, Timestream, Integration: VPC endpoints, Federated queries with Amazon Athena

Hands-on: Create Redshift cluster, load data from S3, Design DynamoDB table for high throughput, Implement S3 lifecycle rules

Mini-Project: Build a data lake in S3 with partitioned Parquet files → Query with Athena

Unit 3	**Data Operations and Support**	**15 hours**

Orchestration: Workflow automation, dependency management, error handling

Monitoring & troubleshooting: Metrics, logs, alerts, Performance & cost optimization

Support practices: Backup/restore, disaster recovery, auto-scaling

AWS tools: AWS Step Functions (orchestration), AWS Lambda (serverless processing), Amazon CloudWatch (monitoring), AWS X-Ray (tracing)

CI/CD for data pipelines: AWS CodePipeline / CodeBuild,

Hands-on: Orchestrate ETL pipeline with Step Functions + Lambda, set up CloudWatch alarms for Glue job failures, Perform RDS backup & restore

Case Study: Optimize pipeline using EMR auto-scaling and CloudWatch insights

Unit 4	**Data Security and Governance**	**15 hours**

Security controls: Encryption (at rest/in-transit), Access management, network isolation

Governance: Data cataloging, lineage, quality checks, Anomaly detection, validation rules

Compliance: Auditing, tagging, resource policies, PII/sensitive data handling

AWS services: AWS IAM (roles, policies), AWS KMS (encryption), Amazon Macie (data discovery), AWS Lake Formation (governance), Glue Data Catalog (metadata)

Best practices: Least privilege, MFA, VPC security groups

Hands-on: Column-level encryption in Redshift, Lake Formation permissions for data lake, Macie scan for sensitive data in S3

Mini-Project: Design secure pipeline with IAM roles, KMS, governance tags		
Unit 5	**Advanced AWS Data Engineering Concepts**	**15 hours**
Advanced analytics: Integration with SageMaker for ML pipelines, Real-time processing with Amazon MSK **Cost optimization:** AWS Cost Explorer, Budgets, Performance tuning across services **Best Practices:** AWS Well-Architected Framework **Hands-on:** End-to-end ETL workflow, Simulate exam labs: Athena queries, Step Functions **Mini-Project:** Full Pipeline: AppFlow (ingest) → Glue (transform) → Redshift (store) → Lake Formation (secure) → CloudWatch (monitor) **DEA-C01 Exam Review: Task statements, sample questions, scoring**		
	Total Lecture Hours	**75 hours**

Textbook:

1. AWS Certified Data Engineer Associate Exam Guide – Kabelinlo Longkumer, Packt Publishing, 2024
2. Learning AWS Data Engineering – Navin Sabharwal, Packt Publishing, 2023

References:

3. AWS Certified Data Engineer Associate All-in-One Exam Guide – Ben Piper, O'Reilly Media, 2024
4. Data Engineering with AWS – Gareth Eagar, Packt Publishing, 2023.

Mode of Evaluation						
MSE		**CA**			**ESE**	**Total**
MSE1 40	**MSE2** 40	**CA1** 8	**CA2** 8	**CA3(ATT)** 4		
80		**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. Study of hardware and software requirements of different operating systems (UNIX, LINUX, WINDOWS, Android, IOS
2. Execute various UNIX system calls
3. Write a basic program on VI Editor in Linux.
4. Shell Scripting.
5. Implement FCFS CPU Scheduling Policy
6. Implement SJF CPU Scheduling Policy
7. Implement Priority CPU Scheduling Policy
8. Implement Round Robin CPU Scheduling Policy
9. Implementation of Banker's algorithm
10. Implement the solution for Bounded Buffer (producer-consumer) problem using inter process communication techniques-Semaphores
11. Implement the solutions for Readers-Writers problem using inter process communication technique – Semaphore
12. Implementation First Fit contiguous allocation technique
13. Implementation Best Fit contiguous allocation technique
14. Implementation Worst Fit contiguous allocation technique
15. Implement file storage allocation technique:
    -Contiguous (using array)
16. Implement file storage allocation technique:
    - Linked –list (using linked-list)
    -Indirect allocation (indexing)
17. Comparison of Disk Scheduling Algorithms.

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 12	CA2 13	ESE	Total
25		**25**	**50**