NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA, G.B. NAGAR

(An Autonomous Institute)

Estd.2001
Get Future Ready

NIET
Greater Noida
An Autonomous Institute

Affiliated to

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, UTTAR PRADESH LUCKNOW

Evaluation Scheme & Syllabus

For

Second Year

Bachelor of Technology in Computer Science and Engineering
(Artificial Intelligence and Machine Learning-Collaboration & Twining Program)
(Effective from the Session: 2026-27)

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA, G.B. NAGAR

(AN AUTONOMUS INSTITUTE)

(Artificial Intelligence and Machine Learning-Collaboration & Twining Program)

Evaluation Scheme

Semester III

S. NO	SUBJECT CODES	SUBJECT NAME	TYPE OF SUBJECT	PERIODS			EVALUATION SCHEMES				END SEMESTER		TOTAL	CREDIT
				L	T	P	CT	TA	TOTAL	PS	TE	PE		
1 WEEK'S COMPULSORY INDUCTION PROGRAM														
1	CCSCC0301	Employability Skill Development-I	Mandatory	2	0	0	60	40	100				100	2
2	CAS0301	Statistics and Probability	Mandatory	3	1	0	30	20	50		100		150	4
3	CCSE0303A	Operating Systems	Mandatory	2	0	0	30	20	50		50		100	2
4	CCSE0301	Data Structures and Algorithm-II	Mandatory	3	0	0	30	20	50		100		150	3
5	CCS0301	Machine Learning	Mandatory	2	0	0	30	20	50		50		100	2
6	CCSAI0302	High Performance Computing	Mandatory	3	0	0	30	20	50		100		150	3
7	CCSE0353A	Operating Systems Lab	Mandatory	0	0	4				50		50	100	2
8	CCSE0351	Data Structures and Algorithm-II Lab	Mandatory	0	0	4				50		50	100	2
9	CCS0351	Machine Learning Lab	Mandatory	0	0	2				25		25	50	1
10	CCSE0355	Object Oriented Techniques using Java	Mandatory	0	0	6				50		100	150	3
11	CCSE0359	Internship-I	Mandatory	0	0	2				50			50	1
12	CNC0301 / CNC0302	Artificial Intelligence and Cyber Ethics / Environmental Science	Compulsory Audit	2	0	0	30	20	50				50	NA
13		MOOCs (For B.Tech. Hons. Degree)												
		TOTAL		**17**	**1**	**18**	**210**	**140**	**350**	**225**	**400**	**225**	**1200**	**25**

List of MOOCs Based Recommended Courses for Second year B. Tech Students

S. NO.	SUBJECT CODE	COURSE NAME	UNIVERSITY/ INDUSTRY PARTNER NAME	NO. OF HOURS	CREDITS
1.	CMC0027	Object Oriented Programming Using Python	Infosys Wingspan (Infosys Springboard)	46h 13m	3.5
2.	CMC0028	Probability and Statistics using Python	Infosys Wingspan (Infosys Springboard)	16h	1
3	CMC0029	Cyber Security Essential	NASSCOM	70h	4

PLEASE NOTE: -

- **A 3–4-week Internship shall be conducted during summer break after semester-II and will be assessed during semester-III.**
- **Compulsory Audit (CA) Courses (Non-Credit - CNC0301/CNC0302)**
  - All Compulsory Audit Courses (a qualifying exam) do not require any credit.
  - The total and obtained marks are not added to the grand total.

Abbreviation Used:

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam.
CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit,
MOOCs: Massive Open Online Courses.

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA, G.B. NAGAR

(AN AUTONOMUS INSTITUTE)

(Artificial Intelligence and Machine Learning-Collaboration & Twining Program)

Evaluation Scheme

SEMESTER IV

S.NO	SUBJECT CODES	SUBJECT NAME	TYPE OF SUBJECT	PERIODS			EVALUATION SCHEMES				END SEMESTER		TOTAL	CREDIT
				L	T	P	CT	TA	TOTAL	PS	TE	PE		
1	CASCC0401	Employability Skill Development – II	Mandatory	2	0	0	60	40	100				100	2
2	CCSE0401	Database Management Systems	Mandatory	2	0	0	30	20	50		50		100	2
3	CCSML0402	Deep Learning	Mandatory	3	0	0	30	20	50		100		150	3
4	CCSE0403	Theory of Computation	Mandatory	3	0	0	30	20	50		100		150	3
5	CCSDS0401	Programming for Data Analytics	Departmental Elective	3	0	0	30	20	50		100		150	3
6	CASL0401	Technical Communication	Mandatory	2	0	0	30	20	50		50		100	2
7	CCSE0451	Database Management Systems Lab	Mandatory	0	0	4				50		50	100	2
8	CCSML0452	Deep Learning Lab	Mandatory	0	0	2				25		25	50	1
9	CCSE0455	Web Technologies	Mandatory	0	0	6				50		100	150	3
10	CCSE00459	Mini Project	Mandatory	0	0	2				50			50	1
11	CCSCC0452	Problem Solving Approaches	Mandatory	0	0	2				50			50	1
12	CNC0402 / CN C0401	Environmental Science / Artificial Intelligence And Cyber Ethics	Compulsory Audit	2	0	0	30	20	50				50	NA
13		MOOCs (For B.Tech. Hons. Degree)	*MOOCs											
		TOTAL		**17**	**0**	**16**	**210**	**140**	**350**	**225**	**400**	**175**	**1150**	**23**

List of MOOCs Based Recommended Courses for Second year B. Tech Students

S. NO.	SUBJECT CODE	COURSE NAME	UNIVERSITY/ INDUSTRY PARTNER NAME	NO. OF HOURS	CREDITS
1.	CMC0049	R Programming for Machine Learning	IIHT	21h 36m	1.5
2.	CMC0050	Deep Learning for Devlopers	Infosys Wingspan (Infosys Springboard)	34h 51m	2.5
3	CMC0051	An Introduction to world of AI with Machine Learning	NASSCOM	30h	2

PLEASE NOTE: -

- **Compulsory Audit (CA) Courses (Non-Credit - CNC0401/CNC0402)**
  - All Compulsory Audit Courses (a qualifying exam) do not require any credit.
  - The total and obtained marks are not added to the grand total.

Abbreviation Used:

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam.
CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit,
MOOCs: Massive Open Online Courses.

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA
(An Autonomous Institute)

AICTE Guidelines in Model Curriculum:

A student will be eligible to get Under Graduate degree with Honours only, if he/she completes the additional MOOCs courses such as Infosys Springboard/NASSCOM/CISCO (as applicable) certifications, or any other online courses recommended by the Institute (Equivalent to 20 credits). During Complete B.Tech. Program Guidelines for credit calculations are as follows.

1. For 6 to 12 Hours =0.5 Credit
2. For 13 to18 =1 Credit
3. For 19 to 24 =1.5 Credit
4. For 25 to 30 =2 Credit
5. For 31 to 35 =2.5 Credit
6. For 36 to 41 =3 Credit
7. For 42 to 47 =3.5 Credit
8. For 48 and above =4 Credit

For registration to MOOCs Courses, the students shall follow Infosys Springboard/NASSCOM/CISCO (as applicable) registration details as per the assigned login and password by the Institute these courses may be cleared during the B. Tech degree program (as per the list provided). After successful completion of these MOOCs courses, the students shall provide their successful completion status/certificates to the Controller of Examination (COE) of the Institute through their coordinators/Mentors only.

The students shall be awarded Honours Degree as per following criterion.

i. If he / she secures 7.50 as above CGPA.
ii. Passed each subject of that degree program in the single attempt without any grace.
iii. Successful completion of MOOCs based 20 credits

Course Code: CCSCC0301	Course Name: Employability Skill Development - I	L	T	P	C
Course Offered in: All Bachelor of Technology Branches.		2	0	0	2

Pre-requisite: Students should have successfully completed communication skills courses in the first year.

Course Objectives:

SECTION A: Develop understanding of computer systems, mathematical foundations, and programming concepts while applying structured problem-solving techniques to design logical solutions effectively.
SECTION B: Enhance numerical ability and logical reasoning skills through speed mathematics, number systems, and analytical problem-solving techniques for competitive aptitude development.
SECTION C: To develop the ability to communicate effectively and empathetically in workplace situations while strengthening active listening, adaptability, and cross-cultural awareness for navigating complex professional environments.

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Apply fundamental concepts of computer systems, mathematical foundations, and programming constructs to analyze and solve basic computational problems.	K3
CO2	Design and develop structured solutions for problem statements by applying SDLC principles, logical thinking, and flowchart/pseudocode techniques.	K6
CO3	Apply number theory and logical reasoning concepts to solve quantitative and analytical problems efficiently.	K3
CO4	Apply spoken English skills to participate effectively in standard workplace routines, including daily stand-ups, reporting, and collaborative problem-solving.	K3
CO5	Evaluate and adapt communication styles to suit dynamic workplace communication situations and deliver context-sensitive, and logically structured verbal responses under time constraints.	K6

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	-	2	3	1	-	3	2	2	2	1	1	
CO2	3	3	3	3	3	2	1	3	2	2	2	2	2	
CO3	3	3	2	1	-	-	-	1	2	2	2	2	2	
CO4	-	-	-	-	1	2	-	3	2	2	-	-	-	
CO5	-	2	1	-	1	2	-	3	2	2	1	-	-	

Course Contents / Syllabus

Section-A

UNIT 1	FOUNDATIONS OF COMPUTER SYSTEMS AND MATHEMATICAL CONCEPTS	8 hours

Computer System Fundamentals: Introduction to Assembler, Compiler, Interpreter, Role of Loader and Linker in program execution.
Mathematical Foundations for Computing: Sets, Relations, and Functions: definitions and applications, Principle of Mathematical Induction and its use in proofs.
Basic Concepts of Programming Language: Memory model of C/C++ Program, Activation Record of a function, L-Value/ R-Value concept. Operators, Precedence, Order of Evaluation and Associativity. Pointers and Dynamics Memory Concept, Functions (argument passing, concept of scope), Understanding Recursion, Bitwise Operations.

UNIT 2	DESIGN AND DEVELOP STRUCTURED SOLUTIONS FOR PROBLEM STATEMENTS	8 hours

Introduction to Software Development Life Cycle, Step-by-step solution to simple problems, Developing logic/flowchart/pseudocode. Stepwise refinement and Procedural Abstraction.
Case Study*: Understand the problem statement, identify a feasible solution, and develop the logic along with a flowchart for the proposed solution.

SECTION-B		
UNIT 3	**SPEED MATH AND NUMBER SYSTEM**	**16 hours**
Classification of number, Divisibility Rule, Factorization, HCF & LCM, It's Application, Unit digit (Cyclicity), Last two-digit, Remainder theorem, Factorial and Number of zeroes, Highest power. **Analytical and Logical Reasoning**: Direction and Sense, Blood Relation, Number Series and Letter Series, Coding Decoding		
SECTION-C		
UNIT 4	**WORKPLACE INTERACTION ESSENTIALS**	**12 hours**
The Professional Shift: Navigating the transition from academic to professional tone. Using appropriate register when addressing peers vs. supervisors. Active Listening & The Feedback Loop: Moving beyond passive hearing. Practicing the SBI (Situation-Behaviour-Impact) model for peer feedback and clarifying multi-step technical instructions. Structured Micro-Speaking: The art of the concise update. Delivering 1-2 minute "Stand-up" reports (What I did, What I will do, Blockers) and explaining basic technical processes to laymen. Guided Interpersonal Simulations: Role-playing everyday scenarios: politely interrupting a meeting, seeking assistance on a delayed task, and confirming understanding of complex briefs using paraphrasing techniques. Digital & Hybrid Nuances: Verbal signaling in virtual meetings, managing "overtalk," and translating concise written chat messages into spoken professional updates.		
UNIT 5	**WORKPLACE COMMUNICATION IN ACTION**	**12 hours**
Dynamic Problem-Based Communication: Simulating the "War Room." Explaining sudden technical failures, proposing immediate solutions, and negotiating revised deadlines with stakeholders. Listening Under Pressure & Global Adaptability: Interpreting fast, accented, or incomplete instructions (incorporating CEFR-level listening benchmarks). Decoding tone, urgency, and indirect cultural communication styles. Collaborative Conflict Resolution: Managing disagreements in team settings using "I-statements" and the "Yes, and…" technique to build consensus without escalating tension. The Persuasive Professional: Delivering evidence-based arguments using the PREP (Point, Reason, Example, Point) framework to justify a technical choice or resource allocation. Real-Time "Crisis" Simulation (Capstone): A comprehensive role-play where student groups are given a sudden project change or client issue. They must conduct an emergency meeting, delegate tasks verbally, and present a united response to the "client."		
	Total Lecture	**56 Hours**

A list of problem statements will be provided by the mentor; you may choose anyone.
Note: Students may choose an alternative problem statement with prior approval from the mentor.

REFERENCE BOOKS:

BOOK TITLE	AUTHOR
A Project-Based Introduction to Programming	Access Point Publishing
Programming: Principles and Practice Using C++	Bjarne Stroustrup
Effective Modern C++	Scott Meyers
Quantitative Aptitude	R S Aggarwal
Verbal & Non-Verbal Reasoning	R S Aggarwal
Crucial Conversations Tools	Kerry Patterson et al.
HBR Guide to Persuasive Presentations	Nancy Duarte

SELF-STUDY CONTENT LINKS:		DURATION (IN HRS)
MOOCs	https://swayam-plus.swayam2.ac.in/courses/course-details?id=F_WADHW_042	**105 hours**
Video Lectures	Employability Skills Training Session - YouTube	**1 hour**
Web References	https://dde.manuu.edu.in/sites/default/files/DDE/DDE-Assignments/Diploma-Sem1-2024/Employability-Skills-1.pdf	

Course Code: CAS0301	Course Name: Statistics And Probability	L	T	P	C
Course Offered in: AI/AIML/AI(TWIN)/AIML(TWIN)/CYS/DS/CS/CSE/CSE(TWIN)/CSE / IT /M.Tech.(Int.)		3	1	0	4

Pre-requisite: B. Tech. First Year Syllabus

Course Objectives:

The objective of this course is to familiarize the students with concepts of Probability and statistical techniques. It aims to equip the students with adequate Knowledge of statistics that will enable them in formulating and solving problems analytically.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Apply the concept of moments, skewness and kurtosis in relevant field.	K3
CO2	Apply the concept of correlation, regression and curve fitting with real world problems.	K3
CO3	Apply the concept of probability and random variable.	K3
CO4	Apply the concept of Mathematical Expectations and Probability Distribution in real life Problems.	K3
CO5	Apply the concept of hypothesis testing and statistical quality control to create control charts.	K3

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	2	2	3	1	1	-	1	2	2	2	3	1	
CO2	3	2	2	3	1	1	-	1	2	2	2	3	2	
CO3	3	2	1	2	-	-	-	-	1	2	2	2	2	
CO4	3	2	2	3	1	1	-	1	2	2	2	3	2	
CO5	3	2	2	3	1	1	-	1	2	2	2	3	2	

Course Contents / Syllabus

UNIT 1	STATISTICAL TECHNIQUES-I	6 HOURS

Introduction: Measures of central tendency: Mean, Median, Mode, Standard deviation, Quartile deviation, Moments, Skewness, Kurtosis.

UNIT 2	STATISTICAL TECHNIQUES-II	9 HOURS

Curve Fitting: Straight line, Second-degree parabola, Exponential curve by method of least squares, Correlation and Rank correlation, Linear regression, nonlinear regression and multiple linear regression.

UNIT 3	PROBABILITY AND RANDOM VARIABLE	10 HOURS

Random Variable: Definition of a Random Variable, Discrete Random Variable, Continuous Random Variable, Probability mass function, Probability Density Function, Distribution functions.
Bivariate Random Variables: Joint probability mass function, joint density and distribution Function, Properties of Joint Distribution function, Marginal Probability mass function, Marginal density Functions, Conditional Distribution for Bivariate Random Variables, Statistical Independence, Central Limit Theorem (Proof not expected)

UNIT 4	EXPECTATION AND PROBABILITY DISTRIBUTIONS	10 HOURS

Expectations of single Random Variable, Mean, Variance, Moment Generating Function, Binomial, Poisson, Normal, Exponential distribution.

UNIT 5	HYPOTHESIS TESTING AND CONTROL CHARTS	10 HOURS

Testing a Hypothesis, Null hypothesis, Alternative hypothesis, Level of significance, Confidence limits, Test of significance of means by Z-test and t-test, Chi-square test, F-test, One way ANOVA. Statistical Quality Control (SQC), Control Charts, Control Charts for variables (Mean and Range Charts), Control Charts (p, np and C charts)

	Total Lecture Hours	45 HOURS

Textbook:	
Book Details	***Authors Name***
Textbook of Engineering Mathematics-IV, *Laxmi Publications, Latest Edition*	*N. P. Bali*
Advanced Engineering Mathematics, *Narosa Publishing House, Latest Edition*	*R. K. Jain*
Higher Engineering Mathematics, *Khanna Publishers, Latest Edition*	*B. S. Grewal*
Statistical Methods, *Sultan Chand & Sons, Latest Edition*	*S. P. Gupta*
Advanced Engineering Mathematics, *Jones & Bartlett Learning, Latest Edition*	*Dennis G. Zill and Warren S. Wright*

Reference Books:	
Book Details	***Authors Name***
Introduction to Probability Models, *Pearson, Latest Edition*	*Sheldon M. Ross*
Probability, Random Variables and Stochastic Processes, *McGraw-Hill Education, Latest Edition*	*Athanasios Papoulis and S. Unnikrishna Pillai*
Advanced Engineering Mathematics, *John Wiley & Sons, Latest Edition*	*Erwin Kreyszig*

Self-Study Content Links:		**Duration (in Hrs)**
MOOC'S	https://www.futureskillsprime.in/course/probability-and-statistics/	30 hrs
VIDEO LECTURES	https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_ma74 https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_ma77 https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_mg10	13 hrs 13 hrs 13 hrs
WEB REFERENCES	https://www.geeksforgeeks.org/maths/probability-and-statistics/ https://www.geeksforgeeks.org/machinelearning/statistics-for-machine-learning/	3 hrs 2 hrs

Course Code: CCSE0303A	Course Name: Operating Systems	L	T	P	C
Course Offered in: CSE/CSE-R/IT/CS/CSE(AI)/CSE(AIML)/ CSE(IOT)/CSE(DS)/CSE(CYS)/M.Tech.Int. /MCT		2	0	0	2

Pre-requisite: Basic knowledge of computer fundamentals, C programming, Data structures and Computer organization.

Course Objectives:

The objective of the course is to provide a foundational understanding of operating system concepts, including system architecture, process and thread management, concurrency, deadlock, resource management, memory and file systems, Linux shell scripting, and an introduction to virtualization and distributed systems.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Understand operating system architecture and types and use the Linux CLI for basic Operations.	**K2**
CO2	Implement the CPU scheduling algorithms including uses of multithreading models.	**K3**
CO3	Implement concurrency control, process synchronization techniques, and deadlock handling techniques.	**K3**
CO4	Implement memory management strategies to optimize system performance and analyze file systems and configure virtual machines in modern operating systems.	**K3**
CO5	Analyze file systems and configure distributed systems and virtual machines in modern operating systems.	**K4**

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

CO - PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	2	2	1	2	-	-	-	-	1	1	2	1	
CO2	3	3	3	2	2	-	-	-	-	1	1	2	2	
CO3	3	3	3	2	2	-	-	-	-	1	1	2	3	
CO4	3	3	3	2	2	-	-	-	-	1	2	2	2	
CO5	3	3	2	3	3	2	1	-	2	2	2	3	3	

Course Contents / Syllabus

UNIT 1	FUNDAMENTALS & SHELL SCRIPTING	7 HOURS

Fundamentals of Operating Systems: Overview of Operating Systems, Generations of OS, Operating system architecture, Interrupt handling, System call and kernel, Types of Operating System: Batch OS, Multi-programming OS, Multitasking OS, Multiprocessor OS, Real time OS.
Shell Scripting in Linux: Introduction to Linux Operating System & Architecture, Basic Command Line Interface (CLI) Operations in Linux, Shell Scripting Basics: Variables, Control Structures, Functions.
Case study: Automating system administration tasks using shell scripts in Linux.

UNIT 2	PROCESS & THREAD MANAGEMENT	7 HOURS

Process Management: Process, Transition Diagram, Process Control Block (PCB), Types of Schedulers: Long Term, Mid Term, Short Term Scheduler,
CPU Scheduling: Pre-emptive and non-pre-emptive Algorithm (FCFS, SJF, SRTF, Non-Pre-emptive Priority, Pre-emptive Priority, Round Robin, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling)
Thread: Processes Vs Threads, Thread states, Benefits of threads, Types of threads, Multithreading Models, Concept of Hyper-Threading.
Case study: Analyse and implement CPU Scheduling in Real-Time Embedded Systems and RTOS.

UNIT 3	CONCURRENCY AND DEADLOCK MANAGEMENT	8 HOURS

Concurrency: Introduction of Concurrency, Types of Process, Race Condition, Critical Section, Inter Process Communication, Producer consumer problem.
Process Synchronization: Lock variable, Peterson's Solution, Strict alternation, Lamport Bakery Solution, Test and set lock, Semaphore- counting, binary and monitor
Classical Problem of Synchronization: Bound Buffer, Dinning Philosopher, Reader writer, Sleeping barber.
Deadlock: Deadlock, Deadlock characterization, Deadlock Prevention, Deadlock Avoidance: Bankers Algorithms, Deadlock Detection, Recovery from Deadlock.
Case study: Deadlock avoidance in database transaction management systems like Oracle or MySQL.

UNIT 4	MEMORY MANAGEMENT	8 HOURS

Memory Management: Memory Management function, Loading and linking Address Binding, Memory management techniques, Contiguous technique- Fixed Partitions, variable partitions, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Non-contiguous, Paging, Segmentation, Segmented paging,
Virtual Memory: Virtual Memory Concepts, Demand Paging, Performance of Demand Paging, Page Replacement Algorithms: FIFO, LRU, Optimal and LFU, Balady's Anomaly, Thrashing
File and Disk Management: File Access Mechanism, File Allocation Method, Disk Scheduling Algorithms.
Modern Operating System: Virtual machines – hypervisor, Introduction to GPU. Introduction of CUDA Programming.
Case study: Virtual memory management in modern OS like Windows 10 and how paging impacts performance.

	Total Lecture Hours	**30 HOURS**

Textbook:

Book Details	**Authors Name**
Operating System Concepts Essentials , Willey Publication,10th Edition,2018.	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne
A practical guide to Linux: Commands, Editors and Shell Programming, CreateSpace Independent Publishing Platform, 4th Edition,2017.	Marks G. Sobell
LINUX for beginners, 1stEdtion,2014.	Jason Cannon

Reference Books:

Book Details	**Authors Name**
Operating Systems: Internals and Design Principles, Pearson Education , 9th Edition, 2019.	William Stallings
Operating System: A Design-oriented Approach , McGraw Hill Education,2017.	Charles Patrick Crowley
Learning Linux Shell Scripting, Packt Publishing ,2nd Edition 2018.	Ganesh Naik

Self-Study Content Links:		**Duration (in Hrs)**
MOOC'S	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013843258047987712344 43_shared/overview	29 hrs
Video Lectures	https://www.youtube.com/watch?v=dTl9QkH4j8o&list=PL6CdojO56mZ3SeRfpzMBMObSnTziA0gfE https://www.youtube.com/watch?v=Q35sMYHaIvQ https://www.youtube.com/watch?v=rnGVincwk30 https://www.youtube.com/watch?v=akU1Ji8Vzdk&list=PLbMVogVj5nJRa3VKt_eyZdJ_DitCz1cvQ https://www.youtube. com/watch?v=pPAK s7tT8sw https://www.youtube. com/watch?v=hBPY P0ZEvS8 https://www.youtube. com/watch?v=UeVQ xzlgd10 https://www.youtube.com/watch?v=9cuwgiHQyoI https://www.youtube.com/watch?v=-NONm-Jq34Y&t=25s https://www.y outube.com/w atch?v=5uyG n147rr0 https://www.y outube.com/w atch?v=utYC 2tKDdoQ https://www.y outube.com/w atch?v=Cuo8 GTklgoE https://www.youtube.com/watch?v=Y14b7_T3AEw https://www.youtube.com/watch?v=bK-VhQA512c https://www.youtube.com/watch?v=o2_iCzS9-ZQ https://www.youtube.com/watch?v=SnKgEuUfV4k https://www.youtube.com/watch?v=86FAWCzIe_4	30 hrs
Web References	https://www.geeksforgeeks.org/operating-systems/operating-systems/ https://studyhubzone.com/operating-systems-notes/	4 hrs

Course Code: CCSE0301	Course Name: Data Structures And Algorithms-II	L	T	P	C
Course Offered in: CSE/CSE-R/CS/IT/CSE(AI)/CSE(AIML)/CSE(DS)/CSE(IOT)/M.tech.Int/MCT/CSE(CYS)		3	0	0	3

Pre-requisite: C, Python

Course Objectives:

The objective of the course is to learn the basic concepts of algorithm analysis, along with the implementation of non-linear data structures.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Apply tree structures to solve specific problems using various tree operations and algorithms.	K3
CO2	Analyse the graph data structure and evaluate the efficiency of its operations for problem solving.	K4
CO3	Evaluate dynamic programming solutions for efficient problem-solving across diverse contexts.	K4
CO4	Apply efficient backtracking and branch –and –bound techniques across diverse problem-solving scenarios.	K3
CO5	Understand principles of advanced data structures, including their implementation and applications.	K2

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

CO-PO MAPPING	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	1	1	2	1	1	2	1	-	2	2
CO2	3	3	2	1	1	2	1	1	2	1	-	2	2
CO3	3	3	2	2	2	2	1	1	2	2	-	2	2
CO4	3	3	3	2	2	2	1	1	2	1	-	2	2
CO5	3	3	3	2	2	2	1	1	2	2	-	2	2

Course Contents / Syllabus

UNIT 1	DESIGN AND ANALYSIS OF ALGORITHMS: TREE	9 HOURS

Trees: Terminology used with Trees, Binary Tree, Memory representation of Tree, Traversal Algorithms: In-order, Pre-order, and post-order. Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search tree, Binary Heaps, Threaded Binary trees, Traversing Threaded Binary trees, AVL Tree.
Application of Trees: Priority Queue, Heap Sort

UNIT 2	DESIGN AND ANALYSIS OF ALGORITHMS: GRAPH	10 HOURS

Graphs: Terminology used with Graph, Data Structure for Graph Representations: Adjacency matrices, Adjacency List. Graph Traversal: Depth First Search and Breadth First Search. Connected Component, Spanning Trees. Algorithms on Graphs: Minimum Cost Spanning
Trees: Prim' s and Kruskal's algorithm , Transitive Closure and Shortest Path algorithms: Dijkstra Algorithm, Bellman Ford Algorithm, Floyd Warshall's Algorithm.

UNIT 3	DYNAMIC PROGRAMMING	9 HOURS

Dynamic Programming: Dynamic Programming concepts 0/1 Knapsack, Longest Common Sub Sequence, Matrix Chain Multiplication, Resource Allocation Problem.

UNIT 4	BACKTRACKING, BRANCH AND BOUND	9 HOURS

Backtracking: Backtracking, Branch, and Bound with Examples Such as Travelling Salesman Problem, Graph Colouring, n-Queen Problem, Hamiltonian Cycles, and Sum of Subsets.

UNIT 5	ADVANCED- DATA STRUCTURES	8 HOURS

Red-Black Trees, B – Trees, B+ Trees, Binomial Heaps, Fibonacci Heaps

Total Lecture Hours	45 HOURS

Textbook:

Book Details	Authors Name
Data Structures and Algorithms in Python: An Indian Adaptation, 1st Edition, 2021	*Michael T. Goodrich, Roberto Tamassia*
Data Structures Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd, 2nd Edition, 2017.	*Lipschutz*
Introduction to Algorithms, Printice Hall of India, 4th Edition, 2022	*Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest,*

Reference Books:

Book Details	Authors Name
Data Structure Using C, Oxford University Press, 2nd Edition, 2014.	*Reema Thareja*
Data Structure Using C, Pearson Education India, 2nd Edition,2011.	*AK Sharma*
C and Data structure, Wiley Dreamtech Publication, 1st Edition, 2004.	*P. S. Deshpandey*

Self-Study Content Links:		Duration (in Hrs)
MOOC'S	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013842032404848640104 70_shared/overview	18 hrs
	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013094439914094592258 4_shared/overview	3 hrs

VIDEO LECTURES	https://www.youtube.com/watch?v=zWg7U0OEAoE&list=PLBF376 3AF2E1C572F https://www.youtube.com/watch?v=LQx9E2--5c&pp=ygUMYXJyYXlzIG5wdGVs https://www.youtube.com/watch?v=K7VIKlUdo20&pp=ygUPbGluay BsaXN0IG5wdGVs https://www.youtube.com/watch?v=g1USSZVWDsY&list=PLB3CD 0BBB95C1BF09&index=2&pp=iAQB https://www.youtube.com/watch?v=THMyk2_p530&pp=ygUccXVld WUgZGF0YSBzdHJ1Y3R1cmUgICBucHRlbA%3D%3D https://www.youtube.com/watch?v=_VV9v41FIq0&pp=ygUZZGl2a WRlIGFuZCBjb25xdWVyICBucHRlbA%3D%3D https://www.youtube.com/watch?v=ARvQcqJ_-NY&list=PLfFeAJ-vQopt_S5XlayyvDFL_mi2pGJE3	40 hrs
WEB REFERENCES	https://www.msajce-edu.in/academics/it/LectureNote/CD3291-LN.pdf	

Course Code: CCSML0301	Course Name: Machine Learning	L	T	P	C
Course Offered in: CSE(AI)/CSE(AIML)		2	0	0	2

Pre-requisite: Python

Course Objectives:

This course aims to provide students with a strong foundation in the fundamental concepts and techniques of Machine Learning. It enables learners to understand and apply data preprocessing and feature engineering methods for preparing datasets for analysis and model development. The course focuses on the implementation of regression and classification algorithms for predictive analytics and decision-making tasks. Students will also explore clustering techniques, ensemble learning methods, and other advanced machine learning approaches for discovering patterns and improving model performance.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Explain fundamental concepts of Machine Learning, types of learning, and data representation techniques	K2
CO2	Apply data preprocessing, feature engineering, and model evaluation techniques on datasets	K3
CO3	Analyze regression and classification models using appropriate algorithms	K4
CO4	Analyze clustering, probabilistic models, ensemble methods, and anomaly detection techniques for real-world problems	K4

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

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	2	2	3	-	-	-	-	-	-	3	3	
CO2	3	3	3	2	3	-	-	-	-	-	1	3	3	
CO3	3	3	3	3	3	-	-	-	1	1	-	3	3	
CO4	3	3	3	3	3	-	-	-	1	1	1	3	3	

Course Contents / Syllabus

UNIT 1	Introduction, Data Handling & Feature Engineering	07 hours

Introduction to Machine Learning: Motivation, definition, applications; **Types of ML:** Supervised, Unsupervised, Reinforcement Learning; **Datasets and Features:** Types of features, **Data Preprocessing:** Handling missing data, Categorical feature encoding, Feature scaling, Outlier detection and removal; Feature Selection techniques, **Dimensionality Reduction**: Curse of Dimensionality, Principal Component Analysis (PCA) – intuition & working

UNIT 2	Regression, Optimization & Regularization	07 hours

Regression Problem formulation, Linear Regression: Simple, Multiple, Polynomial, **Error Metrics:** MSE, RMSE, MAE, **Cost Function:** Intuition & visualization, **Gradient Descent:** Algorithm, learning rate, Application to linear regression, **Model Evaluation:** R^2 and Adjusted R^2, **Regularization:** Bias–Variance tradeoff, Overfitting & Underfitting, L1 (Lasso) & L2 (Ridge), Regularized cost function.

UNIT 3	Classification Algorithms & Performance Evaluation	09 hours

Classification Problem: Binary & Multiclass, Decision Boundary concept, **Logistic Regression:** Model, cost function, Regularized logistic regression, One-vs-All strategy, **Performance Metrics:** Accuracy, Precision, Recall, F1-score, Confusion Matrix, ROC-AUC Curve, **Classification Algorithms:** Decision Trees (ID3, C4.5, CART), Support Vector Machines (SVM),k-Nearest Neighbors (k-NN), **Probabilistic Models:** Bayesian Learning, Naïve Bayes Classifier, Bayesian Belief Networks.

UNIT 4	Unsupervised & Ensemble Learning	07 hours

Clustering Algorithms: Clustering problem & distance measures, K-Means (algorithm, objective, initialization, K selection), K-Medoids, K-Modes, Hierarchical Clustering (AGNES, DIANA, linkage methods), DBSCAN (density-based clustering), **Ensemble Learning:** Bagging vs Boosting, Random Forest, AdaBoost, Gradient Boosting, XGBoost (overview), **Anomaly Detection:** Concept & applications, Gaussian Distribution & GMM, Anomaly detection algorithm

	Total Lecture Hours	30 hours

Textbook:

S.No	Book Title	Authors:
1.	Machine Learning, McGraw-Hill Education,1997	Mitchell, T. M.
2.	Introduction to Machine Learning, 3rd Edition, MIT Press, 2014	Alpaydin, E.

Reference Books:		
S.no	**Book Title**	**Authors:**
1.	Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, 2nd Edition, O'Reilly Media, 2019	Géron, A
2.	Python Machine Learning, 3rd Edition, Packt Publishing, 2019	Raschka, S., and Mirjalili, V

Self-Study Content Links:		Duration (in Hrs)
MOOCs	https://www.coursera.org/learn/machine-learning	60 Hrs
Video Lectures	https://www.youtube.com/watch?v=_WcSO1BE7To	10 Hrs 51M
Web References	https://www.tensorflow.org/learn	10Hrs

Course Code: CCSAI0302	Course Name: High-Performance Computing	L	T	P	C
Course Offered in: CSE/CS/IT/CSIT/CSE(AI)/CSE(AIML)/ECE/EEE/ELCE/ME		3	0	0	3

Pre-requisite: Fundamentals of Cloud Computing

Course Objectives:

This course introduces the fundamentals of High-Performance Computing (HPC) and modern computing paradigms like cloud, grid, cluster, and quantum computing. It covers computing architecture, parallelism concepts, distributed memory models, and message-passing techniques. Students will also develop parallel applications using OpenMP and explore fault-tolerant mechanisms in cluster systems.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Identify the need and applications of High-Performance Computing.	**K2**
CO2	Analyze computing architectures and parallelism concepts to assess HPC system performance.	**K4**
CO3	Implement distributed memory techniques and message passing for parallel algorithm design.	**K3**
CO4	Evaluate clustering models and fault tolerance methods in cluster computing systems.	**K4**
CO5	High-Throughput Computing (HTC) in Cloud Environments	**K3**

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

CO-PO MAPPING	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	-	1	1	-	-	-	-	1	3	2
CO2	3	3	2	2	2	-	-	-	-	-	1	3	2
CO3	3	3	3	2	3	-	-	-	1	1	1	3	3
CO4	2	3	2	3	2	1	-	-	1	-	2	3	2
CO5	3	3	2	2	3	2	-	-	-	-	2	3	3

Course Contents / Syllabus

UNIT 1	INTRODUCTION TO HIGH-PERFORMANCE COMPUTING	9 HOURS

Computational Thinking, Computing, Why High-Performance Computing, Applications of High-Performance Computing, Parallel programming Software Platforms and its significance, Cloud computing, Grid computing, Cluster computing and Quantum Computing, Multi-core CPUs, Graphical Processing Units.

UNIT 2	COMPUTING ARCHITECTURES	9 HOURS

SISD, SIMD, MISD, MIMD, Memory Hierarchy, Data parallelism, Task parallelism, Bit-level parallelism, Instruction-level parallelism, Concurrency, Decomposition, Mapping, PRAM, NUMA, Multithreading vs Multiprocessing, shared memory model, Open MP, distributed memory model, Message passing interface. Performance measures: Speedup, efficiency and scalability.

UNIT 3	Distributed Memory	10 hours

Distributed memory and Message passing networks, Broadcast, Reduction, Parallel Prefix Sum, Scatter, Gather, Network topologies for parallel computing, Network optimization. Distributed BFS, Graphs and adjacency matrix, Matrix based BFS, CUDA programming, Parallel matrix operations, Sparse vs Dense matrices.

UNIT 4	Cluster Computing	9 hours

BLAS, LAPAC, Cluster Computing, Clustering Models, Clustering Architectures, Clustering Architectures key factors, types of clusters, Mission critical Vs Business Critical Applications, Fault Detection and Masking Algorithms, Check pointing, Heartbeats, Watchdog Timers, Fault recovery through Failover and Failback Concepts

UNIT 5	OpenMP	9 hours

OpenMP, Key concepts, Fork-join model, Goals of OpenMP, Supported platforms and compilers, OpenMP API components: directives, runtime routines, environment variables, General code structure, Core syntax and structured blocks, Compiling OpenMP programs, Parallel region construct, Creating and managing threads, Specifying the number of threads, Thread management and thread IDs, Applications of OpenMP

Total Lecture Hours	45 HOURS

Textbook:

S.No	Book Title	Author
1	The Sourcebook of Parallel Computing	Jack Dongarra, Geoffrey Fox, and Ken Kennedy
2	Petascale Computing: Algorithms and Applications	David A. Bader (Ed.)
3	Parallel and High Performance Computing	Robert Robey and Yuliana Zamora
4	High Performance Computing: Modern Systems and Practices	Thomas Sterling, Maciej Brodowicz, and Matthew Anderson

Reference Books:

S.No	Book Title	Author
1	High Performance Computing: Modern Systems and Practices	Thomas Sterling, Matthew Anderson, and Maciej Brodowicz
2	Using OpenMP: Portable Shared Memory Parallel Programming	Barbara Chapman, Gabriele Jost, and Ruud van der Pas

3	Introduction to Parallel Computing	Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar
4	The OpenMP Common Core: A Hands-on Exploration	Tim Mattson et al.
5	A Hands-on Introduction to OpenMP	Tim Mattson and Larry Meadows
6	OpenMP Tasking Explained	Ruud van der Pas
7	An Introduction to Parallel Programming	Peter Pacheco
Self-Study Content Links:		
VIDEO LECTURES	https://www.youtube.com/watch?v=zJ_D6AiGlV8 https://www.youtube.com/watch?v=FTXikoQr46U https://www.youtube.com/watch?v=gE8KgD9D69k https://www.youtube.com/watch?v=kHV6wmG35po	

LAB Course Code: CCSE0353A	LAB Course Name: Operating Systems Lab	L	T	P	C
Course Offered in: CSE/CSE-R/IT/CS/CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CYS)/MCT		0	0	4	2

Pre-requisite: Basic knowledge of computer fundamentals, C programming, Data structures and Computer organization.

Course Objectives:

This course introduces the fundamentals of High-Performance Computing (HPC) and modern computing paradigms like cloud, grid, cluster, and quantum computing. It covers computing architecture, parallelism concepts, distributed memory models, and message-passing techniques. Students will also develop parallel applications using OpenMP and explore fault-tolerant mechanisms in cluster systems.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Execute basic Linux commands and shell scripts to automate file management and system administration tasks.	**K3**
CO2	Implement and compare various CPU scheduling algorithms, process synchronization solutions using semaphores and deadlock handling algorithms.	**K4**
CO3	Simulate memory allocation techniques and page replacement algorithms, disk management strategies and explore modern OS features including virtualization and distributed computing.	**K5**

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

CO-PO	PO1	PO2	PO3	PO 4	PO5	PO6	PO 7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	2	1	1	1	3	-	-	2	2	-	3	2	1	
CO2	3	3	3	2	2	-	-	2	2	-	2	2	2	
CO3	3	2	3	2	3	-	-	2	2	-	3	2	3	

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

S.No	Experiment	CO Mapping
1	Installation of Linux Operating system.	**CO1**
2	**Implementation of Linux Commands** i. Display system information using uname, hostname, and date etc. ii. File operations using cat, touch, cp, mv, rm, and chmod ,umask etc. iii. Create, view, and navigate directories using mkdir, rmdir, cd, pwd, ls etc. iv. Disk Commands df,du,mount,unmount,mkfs,fsck etc.	**CO1**
3	**Implementation of Linux Commands (continued)** i. Use redirection and piping in commands. ii. File compression and archiving using tar, gzip, zip, unzip etc. iii. Process commands ps,kill, killall,nice, pgrep, top,htop etc. iv. Network commands ifconfig, ping, netstat, host,ip route etc. Administrator Commands Adduser,Passwd, deluser, usermod, groupadd etc	**CO1**
4	**Shell Scripting Programming** i. Write a shell script to ask your name, program name and enrollment number and print it on the screen. ii. Write a shell script to find the sum, the average and the product of the four integers entered. iii. write shell script to find average of numbers given at command line iv. Write a shell program to exchange the values of two variables. v. Write a shell program to Print Numbers 1 to 10 using while & do while loop. Write a shell program to Print Numbers 1 to 10 using for loop.	**CO1**
5	**Shell Scripting Programming (continued)** i. Write a shell script to display the digits which are in odd position in a given 5-digit number. ii. Write a shell program to search for a given number from the list of numbers provided using binary search method. iii. Write a shell program to concatenate two strings and find the length of the resultant string. iv. Write a shell script to find the smallest of three numbers. Write a shell program to count number of words, characters, white spaces and special symbols in a given text.	**CO1**
6	Implement FCFS CPU Scheduling algorithm.	**CO2**
7	Implement the SJF CPU Scheduling algorithm (For both pre-emptive and non-pre-emptive).	**CO2**
8	Implement PRIORITY CPU Scheduling Algorithm (For both pre-emptive and non-pre-emptive).	**CO2**
9	Implement Round-Robin CPU Scheduling Algorithm.	**CO2**
10	Implement Multi-Level Queue CPU Scheduling algorithm.	**CO2**
11	Implement Multilevel Feedback Queue CPU Scheduling Algorithm.	**CO2**
12	Execute the RACE Condition of Process Synchronization.	**CO2**
13	Implement the Producer–consumer problem using semaphores.	**CO2**
14	Design a code and implement the Dinning Philosopher problem.	**CO2**
15	Implement Banker's algorithm of Deadlock Avoidance.	**CO2**
16	Execute an algorithm for Deadlock Detection.	**CO2**
17	Implement Contiguous memory variable size partition scheme.	**CO3**

18	Simulate the First-Fit contiguous memory allocation technique.	**CO3**
19	Simulate the Best-Fit contiguous memory allocation technique.	**CO3**
20	Simulate the Worst-Fit contiguous memory allocation technique.	**CO3**
21	Implement the Non-contiguous Memory Allocation by using Paging.	**CO3**
22	Implement Contiguous memory variable size partition scheme.	**CO3**
23	Write a Program to simulate the FIFO page replacement algorithm.	**CO3**
24	Write a Program to simulate the LRU page replacement Algorithm.	**CO3**
25	Write a Program to simulate the Optimal page replacement Algorithm.	**CO3**
26	Write a program to simulate FCFS Disk Scheduling Algorithm.	**CO3**
27	Write a Program to simulate the SSTF Disk Scheduling Algorithm	**CO3**
28	Write a program to simulate SCAN Disk Scheduling Algorithm.	**CO3**
29	Write a Program to simulate the C SCAN Disk Scheduling Algorithm.	**CO3**
30	Write a Program to simulate the LOOK Disk Scheduling Algorithm.	**CO3**
	TEACHING HOURS	**60 HOURS**

LAB Course Code: CCSE0351	LAB Course Name: Data structure and Algorithms lab-II	L	T	P	C
Course Offered in: CSE/CS/CSR-R/M.TECH(INT) /IT//CSE(AI)/CSE(AIML)/CSE(DS)/CSE(IOT)/MCT		0	0	4	2

Pre-requisite: C, Python

Course Objectives:

Learn to implement linear data structures

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Implementation of tree data structures for basic operations like insertion, deletion, searching and traversal	**K3**
CO2	Implementation of algorithms based on graph data structures for solving real world problems.	**K3**
CO3	Implementing Dynamic Programming, Backtracking, Branch and Bound algorithms to solve complex data efficiently and effectively.	**K3**

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

CO-PO	PO1	PO2	PO3	PO 4	PO5	PO6	PO 7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	2	1	1	2	1	1	2	1	-	2	1	
CO2	3	3	2	1	1	2	1	1	2	1	-	2	1	
CO3	3	3	2	2	2	2	1	1	2	2	-	2	1	

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

S.No	Experiment	CO Mapping
1	Write a program to implement an in-order traversal of a binary tree and print the nodes.	**CO1**
2	Write a program to implement a pre-order traversal of a binary tree and print the nodes.	**CO1**
3	Write a program to implement a post-order traversal of a binary tree and print the nodes.	**CO1**
4	Write a program to count the number of nodes in a binary tree.	**CO1**
5	Write a program to find the height of the tree.	**CO1**
6	Write a program to check if the binary tree is balanced or not.	**CO1**
7	Write a program to search a number in a Binary Search Tree (BST).	**CO1**
8	Write a program to insert a node in a Binary Search Tree (BST).	**CO1**
9	Write a program to delete a node from a Binary Search Tree (BST).	**CO1**
10	Write a program to implement a max-heap and perform heap sort on an array of integers.	**CO1**
11	Write a program to implement a priority queue using a max heap.	**CO1**
12	Write a program to create a graph using an adjacency matrix.	**CO2**
13	Write a program to create a graph using an adjacency list.	**CO2**
14	Write a program to perform Depth-First Search (DFS) on a graph.	**CO2**
15	Write a program to perform Breadth-First Search (BFS) on a graph.	**CO2**
16	Write a program to check if there is a path between two nodes in a graph using DFS.	**CO2**
17	Write a program to find all the vertices reachable from a given vertex in a graph.	**CO2**
18	Write a program to detect a cycle in an undirected graph using DFS.	**CO2**
19	Write a program to detect a cycle in a directed graph using DFS.	**CO2**
20	Write a program to find the degree of each vertex in an undirected graph.	**CO2**
21	Write a program to count the number of connected components in an undirected graph.	**CO2**
22	Write a program to implement Dijkstra's Algorithm.	**CO3**
23	Write a program to implement Prim's Algorithm.	**CO3**
24	Write a program to implement Kruskal's Algorithm.	**CO3**
25	Write a program to implement Floyd–Warshall's all-pairs shortest path algorithm.	**CO3**
26	Write a program to implement Bellman–Ford Algorithm.	**CO3**
27	Write a program to implement Longest Common Subsequence (LCS) using Dynamic Programming.	**CO3**
28	Write a program to implement 0/1 Knapsack using Dynamic Programming.	**CO3**
29	Write a program for Matrix Chain Multiplication using Dynamic Programming.	**CO3**
30	Write a program to implement Sum of Subsets problem using backtracking.	**CO3**
	TEACHING HOURS	**60 HOURS**

LAB Course Code: CCSML0351	LAB Course Name Machine Learning Lab	L	T	P	C
Course Offered in: CSE(AI)/CSE(AIML)		0	0	2	1

Pre-requisite: Python Programming,

Course Objectives:

The course aims to develop practical skills in data preprocessing, feature engineering, supervised and unsupervised machine learning, model evaluation, anomaly detection, and ensemble techniques through hands-on implementation of real-world case studies using Python-based machine learning tools and algorithms.

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

CO1	Apply data preprocessing and feature engineering techniques on datasets	K3
CO2	Analyze and evaluate supervised learning models	K4
CO3	Investigate clustering, probabilistic models, and ensemble techniques for data analysis	K4

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

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	1	2	2	–	–	–	–	1	1	3	1	**2**
CO2	3	3	2	2	3	–	–	–	1	1	2	3	3	**3**
CO3	2	3	2	3	3	–	–	–	1	2	2	3	3	**3**

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

S. No	Experiment (Case Study Based Question)	CO Mapping
1	Load a dataset using Python and perform basic data exploration. Identify number of rows, columns, data types, and summary statistics.	CO1
2	Given a dataset with missing values, implement techniques to handle missing data.	CO1
3	Convert categorical features into numerical format using appropriate encoding techniques.	CO1
4	Apply normalization and standardization techniques on dataset features and compare the results.	CO1
5	Detect and remove outliers from a dataset using statistical methods and visualize the results.	CO1
6	Select important features from a dataset using suitable feature selection techniques.	CO1
7	Reduce dimensionality of a dataset using Principal Component Analysis (PCA) and analyze the variance retained.	CO1
8	Implement simple linear regression on a dataset and predict the output variable.	CO2
9	Build a multiple regression model using multiple input features and evaluate its performance.	CO2
10	Implement binary classification using logistic regression and evaluate model accuracy.	CO2
11	Evaluate the performance of a classification model using confusion matrix, precision, recall, and F1-score.	CO2
12	Implement k-Nearest Neighbors (k-NN) algorithm for classification and analyze the effect of different values of K.	CO2
13	Build and visualize a decision tree classifier for a given dataset.	CO2
14	Apply Naïve Bayes classifier on a dataset and evaluate its performance.	CO2
15	Implement Support Vector Machine (SVM) classifier and compare its performance with other classifiers.	CO2
16	Cluster a dataset into groups using K-Means algorithm and visualize the clusters.	CO3
17	Perform hierarchical clustering on a dataset and represent the results using a dendrogram.	CO3
18	Implement DBSCAN clustering algorithm and identify noise/outlier points.	CO3
19	Implement AdaBoost algorithm and analyze its performance improvement over base learners.	CO4

20	Detect anomalies in a dataset using Gaussian-based anomaly detection techniques.	CO4

Course Code: CCSE0355	Course Name: Object Oriented Techniques Using Java	L	T	P	C
Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech int/MCT		0	0	6	3

Pre-requisite: Basic Programming Knowledge, Knowledge of any programming language (e.g., C, C++, Python)

Course Objectives:

The objective of this course is to understand the object-oriented methodology, and its techniques to design stand alone and GUI applications using hands-on engaging activities.

	Course Outcome: After completion of the course, the student will be able to	**Bloom's Knowledge Level (KL)**
CO1	Understand the concepts of object-oriented programming and relationships among them needed in modeling.	**K2**
CO2	Demonstrate the Java programs using OOP principles and also implement the concepts of lambda expressions	**K3**
CO3	Analyze packages with different protection level resolving namespace collision and implement the error handling concepts for uninterrupted execution of Java program.	**K4**
CO4	Analyze Concurrency control, I/O Streams and Java Socket Programming Concepts.	**K4**
CO5	Design and develop the GUI based application, Generics and Collections in Java programming language to solve the real-world problem	**K6**

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	PSO1	PSO2
CO1	3	3	3	3	2	2	1	-	1	-	2	3	3	2
CO2	3	3	3	3	2	2	1	-	1	1	2	3	2	2
CO3	3	3	3	3	3	2	2	-	2	1	2	3	3	1
CO4	3	3	3	3	3	2	2	1	2	1	2	3	2	2
CO5	3	3	3	3	3	2	2	1	2	1	2	3	3	2

Course Contents / Syllabus

UNIT 1 — BASICS OF JAVA PROGRAMMING — 16 HOURS

Object Oriented Programming: Introduction and Pillars of OOP with real life example, jvm architecture and its components Modelling Concepts: Introduction, Class Diagram and Object Diagram, UML concepts: Association, Composition, aggregation, realization, and Generalization.
Control Statements: Decision Making, Looping and Branching, Argument Passing Mechanism: Command Line Argument, Console Input.
Class and Object: Object Reference, Constructor, Abstraction: Abstract Class, Interface and its uses, Defining Methods, Use of "this" and "super" keyword, Garbage Collection and finalize () Method etc.

UNIT 2 — OOPS FEATURES, ARRAYS AND LAMBDA EXPRESSIONS — 16 HOURS

Inheritance: Introduction and Types of Inheritance in Java, Access Modifiers, Constructors and super constructor in Inheritance. **Polymorphism:** Introduction and Types, Overloading and Overriding. **Lambda expression:** Introduction and Working with Lambda Variables. **Arrays:** Introduction and its Types. Jagged Array with example

UNIT 3 — PACKAGES, EXCEPTION HANDLING AND STRING HANDLING — 16 HOURS

Packages: Introduction and Types, Access Protection in Packages, Import and Execution of Packages.
Exception Handling, Assertions and Localizations: Introduction and Types, Exceptions vs. Errors, Handling of Exception. Finally, Throws and Throw keyword, Multiple Catch Block, Nested Try and Finally Block, Customized Exceptions, Tokenizer. Assertions and Localizations Concepts and its working.
String Handling: Introduction and Types, Operations, Immutable String, Method of String class, String Buffer and String Builder class.

UNIT 4 — CONCURRENCY IN JAVA AND I/O STREAM — 16 HOURS

Threads: Introduction and Types, Creating Threads, Thread Lifecycle, Thread Priorities, Daemon Thread, Runnable Class, Synchronizing Threads etc.
I/O Stream: Introduction and Types, Common I/O Stream Operations, Interaction with I/O Streams Classes. character and byte-oriented stream classes with example
Java Socket Programming: Introduction and types(TCP, UDP), java socket program with server-side and client-side by using connection.

UNIT 5 — GUI PROGRAMMING, GENERICS AND COLLECTIONS — 16 HOURS

GUI Programming: Introduction and Types, Swing, AWT, Components and Containers, Layout Managers and User-Defined Layout

and Event Handling.

Generics: Introduction to Generic Classes, types of generic defined in brief, bounded type parameter (Upper and Lower bound), Initializing a Generic Object, Classes, Methods and Interfaces Use enumerated type.

Collections: Introduction, main interfaces of collections (Collection, List Set, Map, Queue), classes of collections (ArrayList,Linked list, HashSet, HashMap and TreeSet) and methods (List, Set Map) Collection using Iterators.

Total Lecture Hours	**80 HOURS**

Textbook:

Book Details	**Authors Name**
” Java: A Beginner’s Guide”, McGraw-Hill Education 2nd edition	Herbert Schildt
“Programming with Java A Primer”, TMH, 4th edition.	E Balagurusamy,

Reference Books:

Book Details	**Authors Name**
“Core Java Volume I – Fundamentals”, Prentice Hall	Cay S. Horstmann
” Effective Java”, Addison Wesley	Joshua Bloch
Java - The Complete Reference”, McGraw Hill Education 12th edition	Herbert Schildt

Self-Study Content Links:		**Duration (in Hrs)**
MOOC’S	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01254097 22749255681063_shared/overview	46h 13m
VIDEO LECTURES	https://www.youtube.com/watch?v=iw1Xf_33YM0&list=PLQEaRBV9gAFu jcBWJhBT2XXsuMlIfETBy	30h
WEB REFERENCES	https://www.geeksforgeeks.org/dsa/introduction-of-object-oriented-programming/ https://en.wikipedia.org/wiki/Object-oriented_programming	20 m 58 m

LIST OF PRACTICAL

S.N.	List Of Practical’s (Indicative & Not Limited To)	CO
1	Understanding Text Editors to Write Programs Compile and run first java file Byte Code and class file	**CO1**
2	Sketch a class and object diagram describing the sales order system of restaurant	**CO1**
3	Sketch a class diagram describing the circle and rectangle class	**CO1**
4	Sketch a class diagram for a college platform including, classroom, playground, chair, table, smart board, teaching staff etc.	**CO1**
5	Sketch a class diagram containing class called Employee, which models an employee with an ID, name and salary. Add method raiseSalary(percent) that increases the salary by the given percentage.	**CO1**
6	Program to display default value of all Primitive data types	**CO1**
7	Implement the code using main() method to calculate and print the Total and Average marks scored by a student from the input given through the command line arguments. Assume that four command line arguments name, marks1, marks2, marks3 will be passed to the main() method in the below class with name TotalAndAvgMarks.	**CO1**
8	Write code which uses if-then-else statement to check if a given account balance is greater or lesser than the minimum balance. Write a class BalanceCheck with public method checkBalance that takes one parameter balance of type double. Use if-then-else statement and print Balance is low if balance is less than 1000. Otherwise, print Sufficient balance.	**CO1**
9	A class Number Palindrome with a public method isNumberPalindrome that takes one parameter number of type int. Write a code to check whether the given number is palindrome or not. For example, Cmd Args : 333 333 is a palindrome	**CO1**
10	Write a class FibonacciSeries with a main method. The method receives one command line argument. Write a program to display Fibonacci series i.e. 0 1 1 2 3 5 8 13 21	**CO1**
11	Write a Java Program to find the Factorial of a given number.	CO1
12	Java Program to create a class, methods and invoke them inside main method.	CO1
13	Write a Java program to illustrate the abstract class concept. Create an abstract class Shape, which contains an empty method numberOfSides(). Define three classes named Trapezoid, Triangle and Hexagon extends the class Shape, such that	CO1

	each one of the classes contains only the method numberOfSides(), that contains the number of sides in the given geometrical figure. Write a class AbstractExample with the main() method, declare an object to the class Shape, create instances of each class and call numberOfSides() methods of each class.	
14	Java program to illustrate the static field in the class.	CO1
15	Java Program to illustrate static class.	CO1
16	Write a java program to access the class members using super keyword	CO1
17	Java program to access the class members using this keyword	CO1
18	Implement an interface named MountainParts that has a constant named TERRAIN that will store the String value "off_road". The interface will define two methods that accept a String argument name newValue and two that will return the current value of an instance field. The methods are to be named: getSuspension, setSuspension, getType , setType.	CO1
19	Java program to demonstrate nested interface inside a interface.	CO1
20	Java program to demonstrate nested interface inside a class.	CO1
21	Java program to explicit implementation of garbage collection by using finalize() method	CO1
22	Java program to implement Single Inheritance	CO1
23	Java program to implement multi- level Inheritance	CO1
24	Java program to implement constructor and constructor overloading.	CO1
25	Java program implement method overloading.	CO1
26	Java program to implement method overriding.	CO1
27	Java program to implement lambda expression without parameter.	CO1
28	Java program to implement lambda expression with single parameter.	CO1
29	Java program to implement lambda expression with multi parameter.	CO1
30	Java program to implement lambda expression that iterate list of objects	CO2
31	Java program to define lambda expressions as method parameters	CO2
32	Write a class CountOfTwoNumbers with a **public** method compareCountOf that takes three parameters one is arr of type int[] and other two are arg1 and arg2 are of type int and returns true if count of arg1 is greater than arg2 in arr. The return type of compareCountOf should be boolean. Assummptions: 1. arr is never null arg1 and arg2 may be same	CO2
33	Java program to show the multiplication of two matrices using arrays.	CO2
34	Java Program to search an element using Linear Search	CO2
35	Java program to search an element using Binary Search	CO2
36	Java Program to sort element using Insertion Sort	CO2
37	Java Program to sort element using Selection Sort Largest element Method	CO2
38	Java program to Sort elements using Bubble Sort	CO2
39	Java program to create user defined package.	CO3
40	Java Program to create a sub- classing of package	CO3
41	Implement the following: 1. Import package.*; 2. import package.classname; Using fully qualified name.	CO3
42	Implement and demonstrate package names collision in java	CO3
43	Java program to handle and Arithmetic Exception Divided by zero	CO3
44	Java Program to implement User Defined Exception in Java	CO3
45	Java program to illustrate finally block	CO3

47	Java program for creation of illustrating throw in exception handling.	CO3
48	Implement the concept of Assertion in Java Programming Language	CO3
49	Implement the concept of Localization in Java Programming Language.	CO3
50	Java program to print the output by appending all the capital letters in the input string.	CO3
51	Java program that prints the duplicate characters from the string with its count.	CO3
52	Java program to check if two strings are anagrams of each other	CO3
53	Java Program to count the total number of characters in a string	CO3
54	Java Program to count the total number of punctuation characters exists in a String	CO3
55	Java Program to count the total number of vowels and consonants in a string	CO3
56	Java Program to show .equals method and == in java	CO3
57	Given a string, return a new string made of n copies of the first 2 chars of the original string where n is the length of the string. The string may be any length. If there are fewer than 2 chars, use whatever is there. If input is "Wipped" then output should be "WiWiWiWiWi".	CO3
58	Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If the inputs are "Hello" and "World", then the output is "HWeolrllod".	CO3
59	Java program to show the usage of string builder.	CO3
60	Java program to show the usage of string buffer.	CO3
61	Creating and Running a Thread	CO3
62	Implementing Runnable Interface	CO3
63	Synchronizing Threads with lock	CO3
64	Synchronizing Threads without lock	CO3
65	Java program to implement even and odd threads by using Thread class .	CO4
66	Java program to implement even and odd threads by using Runnable interface.	CO4
67	Java program to synchronize the threads by using Synchronize statements and Synchronize block.	CO4
68	Write a program where the client sends a message to the server, and the server prints it by using TCP	CO4
69	Implement a server that can handle multiple clients simultaneously using UDP	CO4
70	Write a client-server application where the client uploads a file and the server saves it by using TCP/UDP.	CO4
71	Java program to implement that read a character stream from input file and print it into output file.	CO4
72	Java program to implement that merge the content of two files (file1.txt, file2.txt) into file3.txt.	CO4
73	Write a Java program that reads the contents of one file and copies them to another file.	CO4
74	Write a Java program that reads a text file and counts the number of words in it.	CO4
75	Write a Java program that reads a text file and counts the frequency of each word in it.	CO4
76	Write a Java program that reads a text file and adds line numbers to each line. The program should create a new file with the line numbers added to the beginning of each line.	CO4
77	Write a Java program that reads two binary files and compares them byte by byte to determine if they are identical. Display a message indicating whether the files are the same or different.	CO4
78	Program to create a frame with three button in AWT and swing	CO4
79	Program to display message with radio buttons in swing	CO4
80	Program to display "All The Best" in 5 different colors on screen. (Using AWT/Swing)	CO5
81	Program to implement handling in a button “OK”	CO5
82	Java Program to implement BorderLayout	CO5
83	Java Program to implement GridLayout	CO5
84	Java Program to implement BoxLayout	CO5
85	Java Program to implement CardLayout	CO5

86	Java program to implement Generic class	CO5
87	Java program to illustrate Generic methods	CO5
88	Java program to implement wildcard in generics	CO5
89	Java program to implement of methods of HashSet	CO5
90	Java Program to implement methods available in HashMap class	CO5
91	Program to add, retrieve, and remove element from ArrayList	CO5
92	Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List.	CO5
93	Create a method which can create a HashSet containing values 1-10.The Set should be declared with the generic type Integer.The method should return the Set.	CO5
94	Java program to implement autoboxing	CO5
95	Java program to implement unboxing	CO5
96	Develop a java class with a method *storeEvenNumbers(int N)* using ArrayList to store even numbers from 2 to N, where N is a integer which is passed as a parameter to the method *storeEvenNumbers()*. The method should return the ArrayList (A1) created.	CO5
97	Create a method that accepts the names of five countries and loads them to an array list and returns the list.	CO5
98	Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List.	CO5

Course Code: CNC0302/CNC0402	Course Name: Environmental Science	L	T	P	C
Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech(Int) / MCT / CSE (CYS)		2	0	0	NA

Pre-requisite: Basic knowledge of biology, chemistry, ecology, geology, mathematics, and understanding of human impacts on natural systems.

Course Objectives:

To help the students in realizing the inter-relationship between man and environment and help the students in acquiring basic knowledge about environment.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Understand the basic principles of ecology and the environment. Ecosystem: Basic concepts, components of ecosystems, food chains and food webs. Ecological pyramids	**K2**
CO2	Understand the different types of natural recourses like food, forest, Minerals and energy and their conservation	**K2**
CO3	Understand the different types of pollution, pollutants, their sources, effects, and their control methods.	**K2**
CO4	Understand the basic concepts of sustainable development, Environmental Impact Assessment (EIA) and different acts related to environment	**K2**

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

CO-PO MAPPING	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	-	3	3	2	2	-	2	2	2
CO2	3	3	2	2	-	3	3	2	2	-	2	2	2
CO3	3	3	2	2	-	3	3	2	2	-	2	2	2
CO4	3	3	2	2	-	3	3	3	2	-	2	2	2

Course Contents / Syllabus

UNIT 1	**Basic Principle of Ecology and Biodiversity**	**5 HOURS**

Definition, Scope and basic principles of ecology and environment. Ecosystem: Basic concepts, components of ecosystem. Food chains and food. Webs. Ecological pyramids, Energy flow in ecological systems, Characteristics of different ecosystems. Biogeochemical Cycles: Importance, gaseous and sedimentary cycles. Carbon, Nitrogen, Phosphorus and Sulphur Cycles. Biodiversity and their importance, Threats to biodiversity, major causes, extinction's, vulnerability of species to extinction, IUCN threat categories, Red data book. Strategies for biodiversity conservation, principles of biodiversity conservation in-situ and ex-situ conservation strategies Mega diversity zones and Hot spots, concepts, distribution and importance.

UNIT 2	**Natural Resources and Ecological succession**	**5 HOURS**

Natural resources and associated problems. Forest resources: Use and over- exploitation, deforestation. Timber extraction, mining, dams and their effects on forest and tribal people. Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources. Food resources: World food problems, changes caused by agriculture and over- grazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, and salinity. Land resources: Land as a resource, land degradation, man induced landslides. Equitable use of resources for sustainable lifestyles. Non-Renewable Energy Resources: Fossil fuels and their reserves, Nuclear energy, types, uses and effects, Renewable Energy Resources: hydropower, Solar energy, geothermal, tidal and wind energy, Biomass energy, biogas and its advantages. Ecological succession-Types, stages, examples of ecological succession

UNIT 3	**Pollution and Waste Management**	**5 HOURS**

Air pollution: sources of air pollution, Primary and secondary air pollutants. Origin and effects of SOX, NOX, Cox, CFC, Hydrocarbon, control of air pollution. Water pollution: sources and types of water pollution, Effects of water pollution, Eutrophication, Soil pollution: Causes of soil pollution, Effects of soil pollution, Major sources of and effects of noise pollution on health, Radioactive and thermal pollution sources and their effects on surrounding environment. Solid waste disposal and its effects on surrounding environment, Introduction to E- Waste, Types and classification of E- Waste, Impacts of E- Waste on environment and human health,E-Waste management and recycling., Climate change, global warming, acid rain, ozone layer depletion.

UNIT 4	**Environmental Assessment and Legislation**	**5 HOURS**

Women education, Role of NGOs regarding environmental protection, Bio indicators and their role, Natural disasters and disasters management, Aims and objectives of Environmental Impact Assessment (EIA). Salient features of following Acts: Environmental Protection Act, 1986, Wildlife (Protection) Act, 1972.Water (Prevention and control of pollution) Act, 1974. Forest (Conserving) Act, 1980. Definition and concept of sustainability, impacted areas of sustainable development, Global initiative, and issues on sustainable development UN SDGs, System Thinking and Sustainability.

	Total Lecture Hours	**20 HOURS**

Textbook:

S.No	**Book Title with publication agency & year**	**Author**

1	Brady, N.C. 1990. The nature and properties of Soils, Tenth Edition. Mac Millan Publishing Co., New York	Brady, N.C
2	Sodhi G.S. 2005, Fundamentals of Environmental Chemistry: Narosa Publishing House, New Delhi.	Sodhi G.S
3	Dash, M.C. (1994), Fundamentals of Ecology, Tata Mc Graw Hill, New Delhi.	Dash, M.C
Reference Books:		
1	Rao M.N. and H.V.N. Rao, 1989 : Air Pollution, Tata McGraw Hill Publishing Co. Ltd., New Delhi	Rao M.N. and H.V.N. Rao
2	A Text Book of environmental Science By Shashi Chawla	Shashi Chawla
Self-Study Content Links:		
MOOC'S		
VIDEO LECTURES	https://www.youtube.com/watch?v=T21OO0sBBfc https://www.youtube.com/watch?v=qt8AMjKKPDo https://www.youtube.com/watch?v=brF0RWJyx9w https://www.youtube.com/watch?v=GK_vRtHJZu4	**7 min 42 sec** **15 min** **3 min 25 sec** **4 min 25 sec**
WEB REFERENCES	https://www.youtube.com/watch?v=yqev1G2iy20, https://www.youtube.com/watch?v=_74S3z3IO_I, https://www.youtube.com/watch?v=jXVw6M6m2g0	**28 min** **31 min 35 sec** **11 min**

Course Code: CASCC0401	Course Name: Employability Skill Development - II	L	T	P	C
Course Offered in: IV Semester Note: Section C - offered in B.Tech. III/IV SEM		2	0	0	2

Pre-requisite: Students should have successfully completed communication skills courses in the first year.

Course Objectives:

Section A: Develop proficiency in C++ programming, data management, and project development by applying concepts, teamwork, and problem-solving to build practical applications.
Section B: Enhance quantitative aptitude skills through business mathematics concepts, enabling accurate problem-solving in financial calculations, ratios, and real-world numerical applications.
Section C: To develop the ability to communicate effectively and empathetically in workplace situations while strengthening active listening, adaptability, and cross-cultural awareness for navigating complex professional environments.

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Apply C++ programming concepts, file handling, and STL techniques to develop efficient programs for data storage and manipulation.	K3
CO2	Design and implement a C++ -based project by applying planning, teamwork, and problem-solving skills.	K6
CO3	Solve real-life business math problems involving percentages, profit and loss, discounts, interest, averages, mixtures, and ratios using appropriate mathematical methods.	K3
CO4	Apply spoken English skills to participate effectively in standard workplace routines, including daily stand-ups, reporting, and collaborative problem- solving.	K3
CO5	Evaluate and adapt communication styles to suit dynamic workplace communication situations and deliver context-sensitive, and logically structured verbal responses under time constraints.	K6

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	-	2	3	1	-	3	2	2	2	1	1	
CO2	3	3	3	3	3	2	1	3	2	2	2	2-	1	
CO3	3	3	2	1	-	-	-	1	2	2	2	1	1	
CO4	-	-	-	-	1	2	-	3	2	2	-	1	2	
CO5	-	2	1	-	1	2	-	3	2	2	1	2	2	

Course Contents / Syllabus

Section-A

UNIT 1	FOUNDATIONS OF C++ PROGRAMMING AND DATA MANAGEMENT TECHNIQUES	8 hours

Core Fundamentals of C++, Functions & Program Structure, Arrays & Strings, File Handling: File streams (ifstream, ofstream, fstream), Reading/writing text files, saving user data (scores, records, logs) STL Basics: vector, array, map, unordered_map set Algorithms (sort, find)

UNIT 2	PROJECT GAME DEVELOPMENT	8 hours

Project Planning & Development (Teams, roles, idea pitching and implement in C++), Mini Project, Project Demonstration and Review

Section-B

UNIT 3	BUSINESS MATH 1	16 hours

Percentage, Profit and Loss, Discount, Simple Interest and Compound Interest, Average, Ratio & Proportion, Partnership, Mixture & Allegation, Clock , Calendar.

Section-C

UNIT 4	WORKPLACE INTERACTION ESSENTIALS	12 hours

The Professional Shift: Navigating the transition from academic to professional tone. Using appropriate register when addressing peers vs. supervisors.
Active Listening & The Feedback Loop: Moving beyond passive hearing. Practicing the SBI (Situation-Behaviour-Impact) model for peer feedback and clarifying multi-step technical instructions.
Structured Micro-Speaking: The art of the concise update. Delivering 1-2 minute "Stand-up" reports (What I did, What I will do, Blockers) and explaining basic technical processes to laymen.

Guided Interpersonal Simulations: Role-playing everyday scenarios: politely interrupting a meeting, seeking assistance on a delayed task, and confirming understanding of complex briefs using paraphrasing techniques. Digital & Hybrid Nuances: Verbal signalling in virtual meetings, managing "overtalk," and translating concise written chat messages into spoken professional updates.		
UNIT 5	**WORKPLACE COMMUNICATION IN ACTION**	**12 hours**
Dynamic Problem-Based Communication: Simulating the "War Room." Explaining sudden technical failures, proposing immediate solutions, and negotiating revised deadlines with stakeholders. Listening Under Pressure & Global Adaptability: Interpreting fast, accented, or incomplete instructions (incorporating CEFR-level listening benchmarks). Decoding tone, urgency, and indirect cultural communication styles. Collaborative Conflict Resolution: Managing disagreements in team settings using "I-statements" and the "Yes, and…" technique to build consensus without escalating tension. The Persuasive Professional: Delivering evidence-based arguments using the PREP (Point, Reason, Example, Point) framework to justify a technical choice or resource allocation. Real-Time "Crisis" Simulation (Capstone): A comprehensive role-play where student groups are given a sudden project change or client issue. They must conduct an emergency meeting, delegate tasks verbally, and present a united response to the "client."		
	Total Lecture Hours	**56 HOURS**

A list of problem statements will be provided by the mentor; you may choose anyone.
Note: Students may choose an alternative problem statement with prior approval from the mentor.

REFERENCE BOOKS:

S. No	**BOOK TITLE**	**AUTHOR**
1	A Project-Based Introduction to Programming	Access Point Publishing
2	Programming: Principles and Practice Using C++	Bjarne Stroustrup
3	Effective Modern C++	Scott Meyers
4	Quantitative Aptitude	R S Aggarwal
5	Verbal & Non-Verbal Reasoning	R S Aggarwal
6	*Crucial Conversations Tools*	Kerry Patterson et al.
7	HBR Guide to Persuasive Presentations	Nancy Duarte
S. NO.	**SELF-STUDY CONTENT LINKS:**	**DURATION (IN HRS)**
MOOCs	nptel.ac.in/courses/109104107	**8 Weeks**
Video Lectures	Employability Skills Training Session - YouTube	**1 hour**
Web References	https://www.scribd.com/document/690580893/EMPLOYABILITY-SKILLS-NOTES-full-1	-

Course Code: CCSE0401	Course Name: Database Management Systems	L	T	P	C
Course Offered in: CSE/CSE-R/IT/M.Tech Int./CSE (Twin)/IT (Twin)/CSE(Prof)/IT (Prof)/M&C/AI/AI (TWIN)/AIML/AIML (TWIN)/CS/CYS/DS/IOT		2	0	0	2

Pre-requisite: Basic understanding of computer fundamentals such as architecture, storage and hardware. In addition, familiarity with data structures, algorithms, and basic programming concepts will be beneficial.

Course Objectives: The objective of the course is to introduce about database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information in relational & non-relational databases.

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Apply ER model for conceptual design of the database.	K3
CO2	Execute SQL and apply the normalization to improve the database design.	K3
CO3	Implement complex queries in database with different applications.	K3
CO4	Understand transaction and concurrency control and execute the concept of PL/SQL.	K3
CO5	Analyze and implement effective Relational and Non-Relational database for real-world problems.	K4

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	3	3	2	1	1	1	2	1	2	3	1
CO2	3	2	3	3	2	2	1	2	1	2	2	3	2
CO3	3	3	2	3	3	2	-	2	1	-	2	3	3
CO4	3	2	2	2	2	2	1	-	1	1	1	3	3
CO5	2	2	2	2	3	2	1	-	1	2	2	3	3

Course Contents / Syllabus

UNIT 1	INTRODUCTION ABOUT THE DATABASE CONCEPTUAL DESIGNING	7 hours

Basic Concept: Database system concept, architecture, History of Database, Data Independence, Database system Vs File system, Data models & Types of Data Models, schema and instances.
Data Modelling using the Entity Relationship Model: ER model concepts, Degree of relationship, Notation for ER diagram, mapping constraints, Generalization, Aggregation, Reduction of an ER diagrams to tables. Extended ER Diagram & reduction of EER. Codd Rules.
Introduction on SQL: Types of SQL commands: -DDL, DML, DCL, TCL. Basic of Relation Algebra & Operations, Query Optimization.

UNIT 2	BASIC OF SQL & NORMALIZATION	7 hours

Keys & Types of Keys: Super key, Candidate Key, Primary Key, Alternative Key, Foreign Key, unique. Constraints and Types of Constraints.
Use of Functions and Clause: Aggregate Functions, Scalar Functions, Where Group by, Having and Order by.
Use of Predicates: SQL Operators. Like, Between, Aliases, distinct, limit.
Functional Dependencies, attribute Closure, Normalization & Types of Normalization, Candidate Key, FD Preservation, Lossy and Lossless decomposition, Canonical Cover of FD's, Denormalization.

UNIT 3	TRANSACTION AND CONCURRENCY CONTROL	8 hours

Transaction system: Life cycle of transaction, ACID Properties Schedule & Types of Schedule, Serializability, Recoverability, Deadlock Handling.
Concurrency Control Techniques: Concurrency Control, Concurrency control Techniques: Locking Techniques, Timestamping, Validation Based Protocol, Transaction & Data Control: -Grant, Revoke, commit & Rollback.

UNIT 4	INTRODUCTION OF NOSQL WITH MONGODB	8 hours

Introduction to NoSQL: NoSQL Data Models, Overview of NoSQL Databases with their Types, Uses & Features of NoSQL Document Databases, CAP theorem, BASE Vs ACID, Comparison of relational databases to NoSQL stores, uses and deployment: MongoDB, Cassandra, HBASE, Neo4j, Redis and Riak. **Introduction to MongoDB:** Introduction and Features of MongoDB, MongoDB Operators, MongoDB Collection & Document, MongoDB Shell & their commands, CRUD operations.
Introduction of Cloud Database. MongoDB Cloud product : Stitch, Atlas & Cloud Manager.

		Total: 30 Hours
TEXTBOOK		
BOOK TITLE		**AUTHOR**
Database Concepts, McGraw Hill, 7th Edition, 2020		Avi Silberschatz Henry F. Korth S. Sudarshan
Fundamentals of Database Systems, Addision Wesley, 7th edition, 2016		R. Elmasri and S. B. Navathe
REFRENCE BOOKS:		
BOOK TITLE		**AUTHOR**
Database Systems: A practical Approach to Design, Implementation and Management. Pearson Education,3rd Edition, 2007.		Thomas Connolly , Carolyn Begg
NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software, Ted Hills, 1st Edition,2016.		Ted Hills
SELF-STUDY CONTENT LINKS:		**DURATION (IN HRS)**
MOOCS	nptel.ac.in/courses/106105175	**16 HOURS**
VIDEO LECTURES	DBMS L1 Inauguration & Introduction DBMS L2 Introduction to Relational Model DBMS L8C Entity Relationship Model DBMS L8D Entity Relationship Model (Problem Solving and Discussion) DBMS L4A Joins, Set Operations and Aggregate Functions DBMS L4B Joins, Set Operations and Aggregate Functions DBMS L9B Relational Database Design DBMS L9C Relational Database Design DBMS L5A Nested Subqueris DBMS L15 Transactions DBMS L16A Concurrency Control	**30 HOURS**
	DBMS L16B Concurrency Control DBMS L16C Concurrency Control L64: Non-relational database management systems | nosql & cap theorem DBMS L19 Distributed Data Stores and NoSQL Databases	
WEB REFERENCES	https://www.geeksforgeeks.org/dbms/dbms/ https://www.scribd.com/document/666843210/Database-Management-System-DBMS-HandWritten-Notes	**2 HOURS**

Course Code: CCSML0452	Course Name: Deep Learning	L	T	P	C
Course Offered in: CSE(AI)/CSE(AIML)		3	0	0	3

Pre-requisite: Python, Machine Learning

Course Objectives:

This course aims to provide students with a comprehensive understanding of Artificial Neural Networks (ANNs) and Deep Learning concepts, forming the foundation for modern intelligent systems. It focuses on the development of Convolutional Neural Network (CNN) models for image classification and recognition tasks, while also enabling students to implement object detection techniques and Recurrent Neural Network (RNN) models for sequential data analysis. The course further explores Autoencoders and feature extraction methods for efficient data representation and learning.

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Analyze ANN models and evaluate their performance metrics.	K4
CO2	Examine CNN models for multi-class image classification.	K4
CO3	Apply deep learning techniques for object detection and recognition.	K3
CO4	Analyze RNN models for time series forecasting and sequence analysis.	K4
CO5	Apply dimensionality reduction techniques for feature extraction and data representation.	K3

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	1	1	1	–	2	1	2	3	-
CO2	3	3	2	2	3	–	–	–	1	1	1	3	3	1
CO3	3	3	3	2	3	–	–	–	2	1	1	3	3	1
CO4	2	3	2	2	2	–	–	1	2	1	1	3	3	1
CO5	2	2	3	2	3	–	–	–	1	1	1	3	3	2

Course Contents / Syllabus

UNIT-1	INTRODUCTION	10 hours

Artificial Neural Network: Neuron, Nerve structure and synapse, Artificial Neuron and its model, activation functions, Neural network architecture: Single layer and Multilayer feed forward networks, Various learning techniques; Perception and Convergence rule, Hebb Learning. Perceptron's, Multilayer perceptron, Gradient descent and the Delta rule, Multilayer networks, Derivation of Backpropagation Algorithm, Introduction to Deep Learning. Regularization for Deep Learning -Parameter Norm Penalties, Dataset Augmentation, Noise Robustness, Early Stopping, Dropout, Batch Normalization.

UNIT-2	CONVOLUTION NEURAL NETWORK	07 hours

Introduction to Computer Vision and CNN, Train a simple convolutional neural net, Explore the design space for convolutional nets, Pooling layer motivation in CNN, Padding and Edge Detection, Strided Convolution, Design a convolutional layered application, Understanding and visualizing a CNN, Transfer learning and fine-tuning CNN, Image classification, Text classification, Image classification and hyper-parameter tuning.

UNIT-3	DETECTION & RECOGNITION	10 hours

Object Detection Fundamentals: Bounding Box and Localization, IoU; Evaluation metrics: Precision, Recall, mAP; Real-Time Detection: YOLO, SSD; Advanced Object Detection Models: YOLOv8, RetinaNet; Object Tracking: Tracking by Detection, Algorithm: SORT, Deep SORT

UNIT-4	RECURRENT NEURAL NETWORKS	10 hours

Sequential Data, Sequence models, Recurrent Neural Network Model, Notation, Back-propagation through time (BTT), Different types of RNNs, Language model and sequence generation, Sampling novel sequences, Vanishing gradients with RNNs, Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Bidirectional RNN, Deep RNNs

UNIT-5	AUTO ENCODERS IN DEEP LEARNING	08 hours

Definition and Motivation, Architecture of Autoencoder, Loss Functions in Autoencoders, Type of Autoencoder, Variational Autoencoder, Stacked auto-encoders, Sequence Autoencoder Application of Autoencoder.

Total: 45 Hours

TEXTBOOK

BOOK TITLE	AUTHOR

Deep Learning: An MIT Press Book	Ian Goodfellow and Yoshua Bengio Aaron Courville.
Neural Networks and Deep Learning, Determination Press,2015.	Michael Nielson
Neural networks: A classroom Approach, Tata McGraw-Hill Education, 2004.	Satish Kumar

REFRENCE BOOKS:

BOOK TITLE	AUTHOR
Deep Learning with Python, Manning publications, 2018	Francois Chollet
Advanced Deep Learning with Keras, PACKT Publications, 2018	Rowel Atienza

SELF-STUDY CONTENT LINKS:		DURATION (IN HRS)
MOOCS	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01274814254931148859_shared/overview	34h 51m
VIDEO LECTURES	YOLOv8 Full Tutorial Playlist | Object Detection, Segmentation & Classification Explained - YouTube:	10h 30m
WEB REFERENCES	https://www.datacamp.com/blog/yolo-object-detection-explained	2h

Course Code: CCSE0403	Course Name: Theory of Computation	L	T	P	C
Course Offered in: CSE/IT/CSE R/CSE(Twin)/IT(Twin)/CSE(AI)/CSE(AI Twin)/CSE(AIML)/CSE(AIML TWIN)		3	0	0	3

Pre-requisite: Mathematical Foundations , Fundamentals of Computer System

Course Objectives:

The primary objective of this course is to provide a foundational understanding of Automata Theory and its role in the language, processing systems, also explores their application in fields like Natural Language Processing (NLP), speech recognition

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Identify the fundamental concepts of automata theory, formal languages and compiler components.	K2
CO2	Understand the role of finite automata, regular expressions, and grammar rules in language processing.	K2
CO3	Demonstrate context-free grammars, pushdown automata, and syntax-directed translation to construct intermediate code for language processors.	K3
CO4	Analyze various parsing strategies, code translation methods, and intermediate representations in compiler phases.	K4
CO5	Analyze the functioning of Turing machine models and optimization techniques in code generation for performance improvement.	K4

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	2	–	–	–	–	–	–	2	3	2
CO2	3	3	3	2	1	–	–	–	–	–	2	3	2
CO3	3	3	3	3	1	–	–	–	–	–	2	3	2
CO4	3	3	3	3	1	–	–	–	–	–	2	3	2
CO5	3	3	3	3	1	–	–	–	–	–	2	3	2

Course Contents / Syllabus

UNIT-1	INTRODUCTION TO FINITE AUTOMATA AND COMPILER	10 hours

Introduction: Role of Automata and Formal languages, Alphabet, String, Grammar, Language, Chomsky Hierarchy of languages. **Introduction to Compiler:** Translators, Language Processing System, Phases and passes of compilation **Use Case 1 :** Role of Finite Automata in NLP and Speech Recognition.

UNIT-2	REGULAR EXPRESSION AND TOKENIZATION	9 hours

Regular Expression: Regular Expression, Regular Sets, Properties of Regular Expression, Finite Automata and Regular Expression, Arden's Theorem, Regular Grammars-Right Linear and Left Linear grammars. **Lexical Analyzer**: Role of lexical Analyzer, Specifications and Recognition of tokens, Lex,

UNIT-3	CONTEXT FREE GRAMMAR AND PUSH DOWN AUTOMATA	09 hours

Context Free Grammar (CFG): Definition and Language, Derivations, Parse Trees and Ambiguity, Simplification of CFG, **Push Down Automata:** Definition of the Pushdown Automata, Languages of PDA

UNIT-4	PARSER AND INTERMEDIATE REPRESENTATION	10 hours

Parser: Role of parser, Top down Parsing-LL (1) parser, Bottom up parsing- shift reduce parser and LR (0), SLR parser, Introduction to Syntax directed Translation, **Intermediate-Code Generation:** Three-Address Code- Quadruples, Triples, Indirect triples **Use Case-2**:Role of CFG and parsing in Voice Assistant

UNIT-5	TURING MACHINE AND OPTIMIZATION	10 hours

Turing Machine: Basic Concept of Turing Machine, Variants of Turing Machine, Universal Turing Machine **Code optimization and generation:** Basic Block, Flow graph, DAG, Optimization Techniques

Total: 45 Hours

TEXTBOOK

BOOK TITLE	AUTHOR
Theory of Computer Science-Automata, Languages and Computation	K.L.P. Mishra, and N. Chandrasekharan ,
Compilers: Principles, Techniques and Tools	Alfred V. Aho, Jeffrey D Ullman

REFRENCE BOOKS:

BOOK TITLE	AUTHOR
Introduction to languages and the theory of computation	J Martin
Compiler Design in C	Allen I. Holub

SELF-STUDY CONTENT LINKS:	DURATION (IN HRS)
https://archive.nptel.ac.in/courses/106/106/106106049/	18 hours

MOOCS	https://archive.nptel.ac.in/courses/106/108/106108113/	
	https://archive.nptel.ac.in/courses/106/108/106108113/	
	https://archive.nptel.ac.in/courses/106/106/106106049/	
	https://archive.nptel.ac.in/courses/106/106/106106049/	
	https://archive.nptel.ac.in/courses/106/108/106108113/	
VIDEO LECTURES	https://www.youtube.com/watch?v=539Bk9fFOyo	**42Hrs**
	https://www.youtube.com/watch?v=6b40kKe2SFg	
	https://www.youtube.com/watch?v=1qOMlqE6LhU	
	https://www.youtube.com/watch?v=1qOMlqE6LhU	
	https://www.youtube.com/watch?v=BR6fHjKFqa0	
WEB REFERENCES	https://archive.nptel.ac.in/courses/106/108/106108113/	**3Hrs**

Course code: CCSDS0401	PROGRAMMING FOR DATA ANALYTICS	L	T	P	C
Course Offered in: CSE(AIML-TWIN)/CSE(AI-TWIN)		3	0	0	3

Pre-requisites: Basic Knowledge of Python and R

Course Objectives:

Demonstrate knowledge of statistical data analysis techniques utilized in business decision making. Apply principles of Data Science to the analysis of business problems. Use data mining software to solve real-world problems. Employ cutting edge tools and technologies to analyze Big Data.

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Install, Code and Use Python & R Programming Language in R Studio IDE to perform basic tasks on Vectors, Matrices and Data frames.	K1
CO2	Implement the concept of the R packages.	K3
CO3	Understand the basic concept of the MongoDB.	K2
CO4	Understand and apply the concept of the RNN and tensorflow.	K4
CO5	Understand and evaluate the concept of the keras in deep learning.	K4

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	**3**	2	2	1	**3**	–	–	–	–	–	1	2	1	
CO2	2	2	**3**	2	**3**	–	–	–	–	–	1	2	1	
CO3	2	2	2	1	**3**	–	–	–	–	–	1	1	1	
CO4	**3**	**3**	**3**	2	**3**	–	–	–	–	–	1	3	2	
CO5	**3**	**3**	2	**3**	**3**	–	–	–	–	–	1	3	2	

Course Contents / Syllabus

Unit 1	BASIC DATA ANALYSIS USING PYTHON/R	08 hours

Pandas data structures – Series and Data Frame, Data wrangling using pandas, Statistics with Pandas, Mathematical Computing Using NumPy, Data visualization with Python Descriptive and Inferential Statistics, Introduction to Model Building, Probability and Hypothesis Testing, Sensitivity Analysis, Regular expression: RE packages.

Unit 2	R GRAPHICAL USER INTERFACES	08 hours

Built-in functions, Data Objects-Data Types & Data Structure, Structure of Data Items, Manipulating and Processing Data in R using Dplyr package & Stringr package, Building R Packages, Running and Manipulating Packages, data import and export, attribute and data types, descriptive statistics, exploratory data analysis, Flexdashboard and R-shiny.

Unit 3	DATA ENGINEERING FOUNDATION	10 hours

Connecting to a database (sqlite) using Python, Sending DML and DDL queries and processing the result from a Python Program, Handling error, NOSQL query using MongoDB, MongoDB Compass.

Unit 4	INTRODUCTION TO TENSOR FLOW AND AI	12 hours

Introduction, Using TensorFlow for AI Systems, Up and Running with TensorFlow, Understanding TensorFlow Basics, Convolutional Neural Networks, Working with Text and Sequences, and TensorBoard Visualization, Word Vectors, Advanced RNN, and Embedding Visualization. TensorFlow Abstractions and Simplifications, Queues, Threads, and Reading Data, Distributed TensorFlow, Exporting and Serving Models with TensorFlow.

Unit 5	DEEP LEARNING WITH KERAS	08 hours

Introducing Advanced Deep Learning with Keras, Deep Neural Networks, Autoencoders, Generative Adversarial Networks (GANs), Improved GANs, Disentangled Representation GANs, Cross-Domain GANs, Variational Autoencoders (VAEs), Deep Reinforcement Learning, Policy Gradient Methods.

Total: 45 Hours

TEXTBOOK

BOOK TITLE	AUTHOR
Making sense of Data: A practical Guide to Exploratory Data Analysis and Data Mining, John Wiley Publishers, 2007.	Glenn J. Myatt,
Learning TensorFlow by Tom Hope, Itay Lieder O'Reilly Media, Inc.	Yehezkel S. Resheff,
Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep R, unsupervised learning, object detection and segmentation, and more, 2nd Edition.	
, Making sense of Data: A practical Guide to Exploratory Data Analysis and Data Mining, John Wiley Publishers, 2007.	Glenn J. Myatt

REFRENCE BOOKS:

BOOK TITLE		AUTHOR
"Professional Hadoop Solutions", 1 st Edition, Wrox, 2013.		Boris lublinsky, Kevin t. Smith, Alexey Yakubovich,
"Understanding Big data", Indian Edition, McGraw Hill, 2015.		Chris Eaton, Dirk Deroos,
"HADOOP: The definitive Guide", 3 rd Edition, O Reilly, 2012		Tom White
SELF-STUDY CONTENT LINKS:		**DURATION (IN HRS)**
MOOCs	Exploratory data analysis using R (11h) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_17171750833801646000_shared/overview	20 hours
	Learn MongoDB (8h 55m) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0140053548969000966513/overview	
Video Lectures	https://www.youtube.com/watch?v=V9Gi-DJF8Ao	19hours
	https://www.youtube.com/watch?v=C5R5SdYzQBI	
	https://www.youtube.com/watch?v=Y6vY-fJbwUA&list=PL9ooVrP1hQOH-yEvwMdMmXbZT1CFx5Bv_	
	https://www.youtube.com/watch?v=IjEZmH7byZQ	
	https://www.youtube.com/watch?v=pWp3PhYI-OU	
Web References	https://www.ibm.com/cloud/blog/python-vs-r https://hevodata.com/learn/data-engineering-and-data-engineers/	

Course Code: CASL0401	Course Name: TECHNICAL COMMUNICATION	L	T	P	C
Course Offered in: B. Tech. All branches (except CSBS)		2	0	0	2

Pre-requisite: Intermediate level (CEFR) and above

Course Objectives:

Demonstrate effective verbal and non-verbal communication skills in diverse professional settings, including meetings, presentations, and interpersonal interactions. Develop and apply clear, concise, and audience-appropriate written communication, such as emails, letters, memos, resume', using correct grammar, tone, and format. Adapt communication style based on cultural, organizational, and situational contexts to foster inclusive and respectful professional relationships. Employ digital communication tools and platforms (e.g., video conferencing, business messaging apps) responsibly and effectively in remote or hybrid work environments.

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Comprehend the principles and functions of technical communication.	K2
CO2	Write for specific audience and purpose to fulfil the provided brief	K3
CO3	Recognize and produce different kinds of technical documents.	K3
CO4	Apply effective speaking skills to efficiently carry out official discourses.	K3
CO5	Demonstrate their understanding of communication through digital media.	K3

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

CO-PO Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	1	1	1	1	1	2	1	2	3	1	1	1	2
CO2	1	1	1	1	1	1	1	2	3	1	1	1	2
CO3	1	1	1	1	1	1	1	2	3	1	1	2	1
CO4	1	1	1	1	1	1	1	2	3	1	1	1	1
CO5	1	1	1	1	1	1	1	2	3	1	1	1	1

Course Contents / Syllabus

UNIT 1	INTRODUCTION TO TECHNICAL COMMUNICATION	4 hours

Technical Communication: Definition, Process, Types, Levels, and Flow; **Barriers to Technical Communication:** emphasis on gender neutral language and cultural sensitivity; **Significance of audience in technical communication**

UNIT 2	TECHNICAL WRITING 1	5 hours

Technical writing skill: characteristics, examples; **Business letters/emails**: Content organization, Tone and intent; **Agenda & Minutes of Meetings**

UNIT 3	TECHNICAL WRITING 2	5 hours

Job application, **Resume'**; **Report, proposal**; **Technical paper**: Abstract; **Ethical Writing:** Copy Editing, Referencing and Plagiarism

UNIT 4	PUBLIC SPEAKING	6 hours

Components of effective speaking: Simplicity, order, balance in arranging ideas. Importance of *KOPPACT;* **Appearing for a job interview:** FAQs; **Telephonic & Online Interviews**

UNIT 5	VIRTUAL/REMOTE COMMUNICATION	4 hours

Remote work: online platforms; **Video conferencing**; **Virtual etiquette**: email ids, usernames; **Writing Blogs & creating Vlogs**

Total: 24 Hours

TEXTBOOK

BOOK TITLE	AUTHOR
Technical Communication – Principles and Practices, 4th Edition , Oxford Univ. Press, 2022, New Delhi.	Meenakshi Raman & Sangeeta Sharma

REFRENCE BOOKS:

BOOK TITLE	AUTHOR
Technical Communication, 15th Edition , Pearson, 2021	John M. Lannon & Laura J. Gurak
Spoken English- A Manual of Speech and Phonetics (5th Edition) by , 2024, New Delhi.	R K Bansal & J B Harrison, Orient Blackswan
Business Correspondence and Report Writing (6 Edition), Tata McGraw Hill & Co. Ltd., 2020, New Delhi.	Prof. R C Sharma, Krishna Mohan, and Virendra Singh Nirban
Intercultural Communication in Virtual Exchange , Cambridge Univ. Press, 2024.	Francesca Helm

SELF-STUDY CONTENT LINKS:		DURATION (IN HRS)
MOOCS	https://onlinecourses.nptel.ac.in/noc24_ge37/preview https://archive.nptel.ac.in/courses/109/106/109106094/	4 WEEKS 4 WEEKS
	https://www.youtube.com/watch?v=kOJlwMJxEG0&t=8s	

VIDEO LECTURES	https://www.youtube.com/watch?v=Sg7Q_dC_fW U&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb https://www.youtube.com/watch?v=ymLFJDpjgCk&list=PL PuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb&index=6	**2.5 HRS**
WEB REFERENCES	Technical Communication And Soft Skills For Engineers	**2.5 HRS**

LAB Course Code: CCSE0451	Lab Course Name: Database Management Systems Lab	L	T	P	C
Course Offered in: CSE/CSE-R/IT/M.Tech Int./CSE (Twin)/IT (Twin)/M&C/AI/AI (TWIN)/AIML/AIML (TWIN)/CS/CYS/DS/IOT		0	0	4	2

Pre-requisite: Basic knowledge of computer fundamentals, programming, data structures, relational database concepts.

Course Objectives:

To familiarize the students to the basics of database design and implementation.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Design and implement ER/EER models to solve real-world problems and transform them into relational schemas using appropriate database tools.	K3
CO2	Apply SQL and PL/SQL to create complex data queries, and procedural operations including triggers and functions, along with database connectivity.	K3
CO3	Analyze and ensure database integrity using constraints and implement unstructured databases using MongoDB with appropriate query operations.	K4

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

CO-PO	PO1	PO2	PO3	PO 4	PO5	PO6	PO 7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	2	1	-	1	2	1	2	3	1	2
CO2	3	3	3	3	2	2	-	2	1	2	2	3	3	2
CO3	2	2	2	2	3	2	-	-	1	2	2	3	3	2

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

S.NO	EXPERIMENT	CO MAPPING
1	Understand and implement the different ER diagram notation with their relationship and cardinalities.	**CO1**
2	Choose a case study on different domains I. E-commerce Platform II. Inventory Management III. Railway System IV. Hospital Data Management V. Company System VI. Banking System	**CO1**
3	Create an ER diagram for the above chosen case-study (Program-2). Identify the suitable entities and find the relationship and cardinalities between the entities with their relevant attributes.	**CO1**
4	Implement DDL, DML, DCL & TCL commands.	**CO2**
5	Implementation of I/O Constraints: Primary Key, Composite primary Key, Foreign Key, Unique Key.	**CO2**
6	Implementation of Business Constraints: Null, Not Null, Default, Check.	**CO2**
7	Practicing Queries using Like, Between, Aliases, distinct Operator & Predicate and Implement Aggregate functions.	**CO2**
8	Implementation of Queries using Where, Group by, Having and Order by Clause.	**CO2**
9	Create a table EMPLOYEE with following schema:-(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation, Salary) Write SQL statements for the following query. i. List the E_no, E name, Salary of all employees working for MANAGER. ii. Display all the details of the employee whose salary is more than the Sal of any IT PROFF. iii. List the employees in the ascending order of Designations of those joined after 1981. iv. List the employees along with their Experience and daily. v. List the employee who are either 'CLERK' or 'ANALYST'. vi. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81. vii. List the e_name those are starting with 'S'. viii. Display total salary spent for each job category. ix. Display lowest paid employee details under each manager. x. Display number of employees working in each department and their department name. xi. Display the details of employees sorting the salary in increasing order. xii. Show the record of employee earning salary greater than 16000 in each department. xiii. Add constraints to check, while entering the empno value (i.e) empno> 100. xiv. Define the field DEPTNO as unique.Create a primary key constraint for the colum(EMPNO).	**CO2**

10	Implementation of Queries using set theory operators UNION, INTERSECT, MINUS.	**CO2**
11	Implementation of Queries using Inner Join and Outer Join along with their types.	**CO2**
12	Implementation of Queries, nested Queries or Sub Queries: IN, NOT IN, Exists, Not Exists, All and Any.	**CO2**
13	Apply the set theory operators, joins and nested queries on the chosen above case-study(Program-2)	**CO2**
14	Understand & implement the Database Connectivity with any programming language.	**CO2**
15	Implementation of Indexing, Views and sequence.	**CO2**
16	I. Write a PL/SQL Program to Add Two Numbers II. Write PL/SQL Program for Fibonacci Series Write PL/SQL Program to Find Greatest of Three Numbers	**CO2**
17	Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named Areas, consisting of two columns Radius and Area.	**CO2**
18	Write a PL/SQL code block that will accept an account number from the user, check if the users balance is less than the minimum balance, only then deduct Rs.100/- from the balance.	**CO2**
19	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values.	**CO2**
20	Implementation of commit and rollback statement with amount transfer example.	**CO2**
21	Study of Open Source NOSQL Database and installation of MongoDB.	**CO3**
22	Implementation of the MongoDB Shell commands.	**CO3**
23	Implementation of the CRUD Operations in MongoDB.	**CO3**
24	Implementation of Aggregate Functions in MongoDB.	**CO3**
	TEACHING HOURS	**60 HOURS**

LAB Course Code: CCSML0452	**LAB Course Name: Deep Learning Lab**	**L**	**T**	**P**	**C**
Course Offered in: CSE(AI)/CSE(AIML)		**0**	**0**	**2**	**1**

Pre-requisite: Python Programming, Machine Learning

Course Objectives:

The course aims to develop practical skills in designing, training, and evaluating deep learning models using ANN, CNN, RNN, Autoencoders, and object detection techniques for solving real-world intelligent system and pattern recognition problems.

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

CO1	Apply ANN models, learning rules, optimization, and regularization techniques for intelligent systems.	K3
CO2	Develop and evaluate CNN models for image classification and object detection tasks.	K4
CO3	Implement RNNs and Autoencoders for sequence learning, prediction, and feature representation.	K3

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

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	1	2	2	–	–	–	–	1	1	3	1	
CO2	2	3	1	3	1	–	–	–	1	2	1	3	3	
CO3	2	3	2	3	3	–	–	–	2	2	2	2	1	

List Of Practical's (Indicative & Not Limited To)		**CO**
S.No.	**Experiment (Case Study Based Question)**	**CO Mapping**
1	An electronics company wants to design a system to predict whether a circuit will turn ON or OFF. Implement an artificial neuron model using Python to simulate this binary decision-making system.	CO1
2	A biomedical system needs nonlinear transformations for classification. Implement and compare activation functions: Sigmoid, ReLU, and Tanh.	CO1
3	A digital automation system must perform logical operations. Design a Perceptron model to simulate AND and OR gates.	CO1
4	A training system must ensure learning convergence. Implement the Perceptron learning rule and verify convergence.	CO1
5	A neural system adapts based on signal correlation. Implement Hebbian Learning rule.	CO1
6	An AI model requires optimization to minimize error. Implement Gradient Descent and Delta Rule.	CO1
7	A company wants to classify customer data. Implement a Multilayer Perceptron (MLP).	CO1
8	A prediction system needs error correction. Implement Backpropagation algorithm.	CO1
9	A model suffers from overfitting. Apply L1 and L2 Regularization techniques.	CO1
10	A model must generalize better. Implement Dropout and Early Stopping techniques.	CO1
11	A surveillance system processes images. Perform image preprocessing using OpenCV.	CO2
12	A digit recognition system is required. Train a CNN model using MNIST dataset.	CO2
13	An engineer wants to understand convolution. Implement convolution operation manually.	CO2
14	An image system must preserve edges. Implement padding and stride operations.	CO2
15	A CNN must reduce dimensions efficiently. Implement Max and Average pooling.	CO2
16	A startup wants to classify product images. Design a CNN for a custom dataset.	CO2
17	A research team wants to visualize CNN learning. Visualize feature maps.	CO2
18	A company wants faster deployment. Implement transfer learning using VGG/ResNet.	CO2

19	A model underperforms. Perform hyperparameter tuning in CNN.	CO2
20	A smart city system needs real-time detection. Implement object detection using YOLO.	CO2
21	A model must be evaluated. Calculate IoU, Precision, Recall.	CO2
22	A traffic system needs detection. Implement SSD object detection.	CO2
23	A surveillance system tracks objects. Implement SORT tracking algorithm.	CO2
24	A weather system uses time-series data. Implement basic RNN.	CO3
25	A sequence model needs temporal learning. Implement BPTT.	CO3
26	A chatbot needs sequence prediction. Implement LSTM.	CO3
27	A stock system needs efficient modeling. Implement GRU.	CO3
28	A compression system needs feature reduction. Implement Autoencoder.	CO3
29	A noisy system needs reconstruction. Implement Denoising Autoencoder.	CO3
30	A probabilistic system is needed. Implement Variational Autoencoder.	CO3
31	A deep feature system is required. Implement Stacked Autoencoder.	CO3
32	A sequence compression system is needed. Implement Sequence Autoencoder.	CO3
	TEACHING HOURS	**30 HOURS**

Course Code: CCSE0455	Course Name: WEB TECHNOLOGIES	L	T	P	C
Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech int		0	0	6	3

Pre-requisite: Basic Programming Knowledge, Knowledge of any programming language (e.g., C, C++, Python/Java), Familiarity with basic concepts of Internet.

Course Objectives:

This course covers different aspect of web technology such as HTML, CSS, Java Script and provide fundamental concepts of Internet, Web Technology and Web Programming. Students will be able to build a proper responsive website.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Identify the basic facts and explaining the basic ideas of Web technology and internet.	**K2**
CO2	Applying and creating various HTML5 semantic elements and application with working on HTML forms for user input.	**K6**
CO3	Understanding and applying the concepts of Creating Style Sheet CSS3 and bootstrap.	**K3**
CO4	Analysing and implementing concept of JavaScript and its applications.	**K6**
CO5	Creating and evaluating dynamic web pages using the concept of PHP.	**K6**

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	PSO1	PSO2
CO1	3	3	2	-	2	-	-	-	2	2	-	3	1	1
CO2	3	2	3	2	3	-	-	-	2	1	2	1	2	3
CO3	3	2	3	-	3	-	-	-	2	2	2	2	3	2
CO4	3	3	3	2	3	-	1	-	2	2	2	3	2	3
CO5	3	3	3	2	3	-	-	-	2	2	2	2	1	2

Course Contents / Syllabus

UNIT 1	BASICS OF WEB TECHNOLOGY & TESTING	16 HOURS

Introduction: Introduction to Web Technology, History of Web and Internet, Connecting to Internet, Introduction to Internet services and tools, Client-Server Computing, Protocols Governing Web, Basic principles involved in developing a web site, Planning process, Types of Websites, Web Standards and W3C recommendations.

Web Hosting: Web Hosting Basics, Types of Hosting Packages, registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel, Using FTP Client, Maintaining a website.

UNIT 2	INTRODUCTION TO HTML & XML	16 HOURS

HTML: What is HTML, DOM- Introduction to Document Object Model, Basic structure of an HTML document, Mark up Tags, Heading-Paragraphs, Line Breaks, Understand the structure of HTML tables. Lists, Working with Hyperlinks, Image Handling, Understanding Frames and their needs, HTML forms for User inputs. New form Elements- date, number, range, email, search and data list, Understanding audio, video and article tags.

XML: Introduction, Tree, Syntax, Elements, Attributes, Namespaces, Display, HTTP request, Parser, DOM, XPath, XSLT, XQuerry, XLink, Validator, DTD, Schema, Server.

UNIT 3	CONCEPT OF CSS3 AND BOOTSTRAP	16 HOURS

Concept of CSS 3: Creating Style Sheet, CSS Properties , CSS Styling(Background, Text Format, Controlling Fonts) , Working with block elements and objects , Working with Lists and Tables , CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties) CSS Advanced(Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector) , CSS Color, Creating page Layout and Site.

Bootstrap: Introduction, Bootstrap grid system, Bootstrap Components.

UNIT 4	JAVASCRIPT AND ES6	16 HOURS

JavaScript Essentials: Introduction to Java Script , JavaScript Types , Var, Let and Const Keywords, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects ,JS Functions , Using Java Script in Real time , Validation of Forms, Arrow functions and default arguments, Template Strings, Strings methods, Callback functions, Object de-structuring, Spread and Rest.

Operator, Typescript fundamentals, Typescript OOPs- Classes, Interfaces, Constructor etc. Decorator and Spread Operator Difference == & ===, Asynchronous Programming in ES6, Promise Constructor, Promise with Chain, Promise Race.

UNIT 5	INTRODUCTION TO PHP	16 HOURS

Introduction to PHP, Basic Syntax, Variables & Constants, Data Type, Operator & Expressions, Control flow and Decision making statements, Functions, Strings, Arrays.

Working with files and directories: Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading.

Session & Cookies: Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session.	
Total Lecture Hours	**80 HOURS**
Textbook:	
Book Details	**Authors Name**
"Web Technology and Design", 1st Edition 2003, New Age International.	C Xavier
"Internet and Web Technologies", 2nd Edition 2017,Mc Graw Hill Education.	Raj Kamal
"Beginning PHP Laravel",2nd Edition 2020, kindle Publication.	Oluwafemi Alofe,
Reference Books:	
Book Details	**Authors Name**
"Collaborative Web Development" 5th Edition 1999, Addison Wesley Publication.	Burdman, Jessica
"Fundamentals of Web Development",3rd Edition 2016	Randy Connolly
"HTML, DHTML, Java Script, Perl & CGI", 4th Edition 2010 BPB Publication	Ivan Bayross
Self-Study Content Links:	**Duration (in Hrs)**
MOOC'S https://onlinecourses.swayam2.ac.in/e-learning/preview/nou24_cs09	12 WEEKS
VIDEO LECTURES 1. https://youtu.be/96xF9phMsWA 2. https://youtu.be/Zopo5C79m2k 3. https://youtu.be/ZliIs7jHi1s 4. https://youtu.be/htbY9-yggB0 5. https://youtu.be/vHmUVQKXlVo 6. https://youtu.be/qz0aGYrrlhU 7. https://youtu.be/BsDoLVMnmZs 8. https://youtu.be/vpAJ0s5S2t0	**52 Hours**
WEB REFERENCES **Web Development Technologies - GeeksforGeeks**	

LIST OF PRACTICAL

S.N.	List Of Practical's (Indicative & Not Limited To)	CO
1	A.Overview and Installation of various code editors.	**CO1**
2	B. Overview and Installation of various servers	**CO1**
3	Implementing HTML program that represent in the document as a start tag, which gives the name and attributes	**CO2**
4	Implementing HTML program that represents a document	**CO2**
5	Implementing HTML program to display your simple CV	**CO2**
6	Creating html document that represents document object model	**CO2**
7	To Create a table to show your class time table.	**CO2**
8	Apply various colors to suitably distinguish keywords , also apply font styling like italics, underline and two other fonts to words you find appropriate , also use header tags.	**CO2**
9	Create a webpage with HTML describing your department use paragraph and list tags	**CO2**
10	Implementing HTML program that for Heading	**CO2**
11	Implementing program that implement paragraph and line-break	**CO2**
12	Use tables to provide layout to your HTML page describing your college infrastructure.	**CO2**
13	Use and tags to provide a layout to the above page instead of a table layout	**CO2**
14	Create links on the words e.g. —Wi-Fi and —LAN to link them to Wikipedia pages	**CO2**
15	Insert an image and create a link such that clicking on image takes user to other page	**CO2**
16	Change the background color of the page; At the bottom create a link to take user to the top of the page.	**CO2**
17	Creating HTML program to implement three articles with independent, self-contained content.	**CO2**
18	Creating a XML document that defines the self-descriptive tags	**CO2**
19	Designing XML document that store various book data such as: book category, title, author, year and price	**CO2**
20	To Describe the various types of XML key components	**CO2**
21	Design XML DTD to define the structure and legal element and attribute of XML document	**CO2**
22	To implement internal and external DTD	**CO2**
23	Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks.	**CO2**
24	Design a HTML registration form that takes user name, user password and mobile number with submit button control	**CO2**
25	Design a HTML5 document that implement of date, number, range, email, search and data list.	**CO2**

26	Implementation in HTML5 that include native audio and video support without the need for Flash.	**CO2**
27	Create a simple form to submit user input like his name, age, address and favourite subject, movie and singer.	**CO2**
28	Add few form elements such as radio buttons, check boxes and password field. Add a submit button at last.	**CO2**
29	Design a XML Schema that describes the structure of an XML document.	**CO2**
30	Design a XML document that describe the well-formed XML document	**CO2**
31	Design a XML document of CD Catalog through each <CD> element, and displays the values of the <ARTIST> and the <TITLE> elements in an HTML table	**CO2**
32	Create a XSL document for and taken xml document by you.	**CO2**
33	Create a XSLT document for and taken xml document by you with all steps	**CO2**
34	Design a web page by applying CSS grouping and dimensions property.	**CO3**
35	Demonstrating the CSS Box model with consists of: borders, padding, margins, and the actual content.	**CO3**
36	Add CSS property assign a style or behavior to an HTML element such as: color, border, margin and font-style.	**CO3**
37	Add To Style Text Elements with Font, Size, and Color in CSS	**CO3**
38	Applying a block element in CSS acquires up the full width available for that content.	**CO3**
39	Design a web page by applying CSS Display and Positioning property.	**CO3**
40	Design a web page by applying CSS Display and Positioning property .	**CO3**
41	Design a web page by applying CSS pseudo classes.	**CO3**
42	Creating a Java Script code to implement all data types.	**CO4**
43	Design a basic structure of Bootstrap Grid system.	**CO4**
44	Design All Bootstrap Components with example.	**CO4**
45	Implementing a program in Java script to implement augmented function.	**CO4**
46	Implementing a program to implement calculator application as real time.	**CO4**
47	Design a HTML form validation using Java Script.	**CO4**
48	Write a program to implement Arrow function with default argument in ES6	**CO4**
49	Implementing a program in ES6 to implement Template string concepts	**CO4**
50	Implementing a program in ES6 to implement all string methods.	**CO4**
51	Creating a Java Script program to implement Dialog, Confirm and Message Popup Boxes.	**CO4**
52	Implementing a Java Script program to implement onClick and onSubmit event	**CO4**
53	Creating a java script code to implement ‘let’ keyword	**CO4**
54	Creating a java script code to implement ‘const’ keyword	**CO4**
55	Implementing a program to implement call back functions in ES6.	**CO4**
56	Implementing a program for de-structuring of an array in ES6	**CO4**
57	Javascript code to implement object and class concepts in Typescript.	**CO4**
58	Write a Typescript program that implement interface and constructor	**CO4**
59	Write a code in typescript that implement decorator and spread operator	**CO4**
60	Create a constant by using define() function with its proper syntax	**CO4**
61	Creating PHP script that return any data types whatever you use.	**CO4**
62	Implementing a code in Java Script to implement Spread and rest operator	**CO4**
63	Javascript code that should compile by Typescript compiler as’tsc’	**CO4**
64	Write a code in typescript that implement Asynchronous Programming concepts.	**CO4**
65	Write a program in Typescript that implement promise constructor	**CO4**
66	Implementing promise and chain concepts in Typescript	**CO4**
67	Write a code in typescript that implement Promise.race() static method.	**CO4**
68	Crating a program that implement control flow and decision making statement.	**CO4**
69	Creating PHP to implements parameterized function	**CO4**
70	Creating program in PHP to store multiple string and concatenate these string and print it.	**CO5**
71	Write a PHP script to create and delete directory structure	**CO5**
72	Program to upload and download a file in PHP	**CO5**
73	Implements single dimension array in PHP	**CO5**
74	Write a PHP code to open and close a file in a proper manner	**CO5**
75	Write a PHP script to copying, renaming and deleting a file.	**CO5**
76	PHP program to create and destroy a session.	**CO5**
77	PHP program to set and delete a cookie.	**CO5**
78	PHP program to manually register the session variable	**CO5**
79	PHP program to manually destroy the session variable	**CO5**
80	PHP program to store the session data on one page and would be available on second page.	**CO5**

Course Code: CCSCC0452	Course Name: PROBLEM SOLVING APPROACHES	L	T	P	C
Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech int		0	0	2	1

Pre-requisite: Programming Language C/C++ or Java or Python

Course Objectives:

Problem-solving in computer programming involves a structured approach to identifying, analyzing, and resolving coding challenges. The process typically includes thoroughly understanding the problem, decomposing it into smaller, manageable parts, designing an appropriate algorithm, implementing the solution through code, and performing testing and debugging to ensure correctness and efficiency.

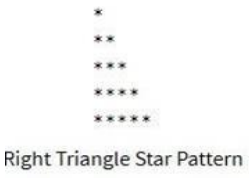
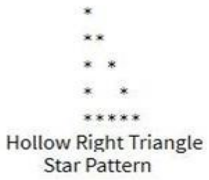
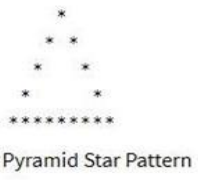
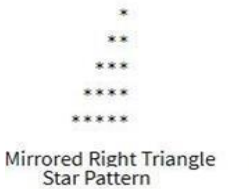
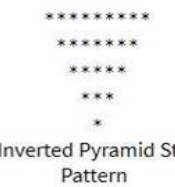
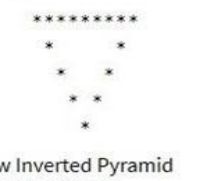
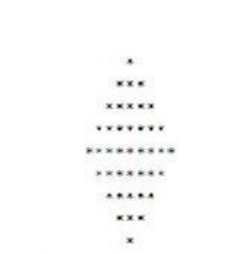
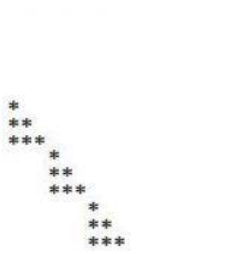
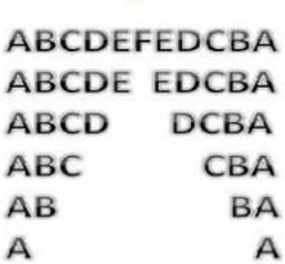
Course Outcome		Bloom's Knowledge Level (KL)
CO1	Develop logic-based solutions using control statements, recursion and bit manipulation to solve basic and intermediate computational problems.	K6
CO2	Implement and manipulate arrays and strings using fundamental and advanced searching sorting techniques.	K3
CO3	Analyze and debug code for logical errors and improve the efficiency of the solution using appropriate data structures and algorithmic patterns.	K4

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

CO-PO Mapping	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
CO1	3	3	3	2	2	1	2	2	-	-	2	3	3	
CO2	3	3	2	2	2	-	2	-	-	-	2	2	2	
CO3	3	3	2	2	3	1	2	2	-	-	3	3	3	

LIST OF PRACTICAL'S

S.NO	PRACTICAL	CO
1	**Secure Password Generator** A company wants to create a secure password generator for their employees. The password must be based on specific numeric properties to enhance its complexity and security. Write a program to validate and generate a secure password according to the following rules: ***1. Prime Number Validation:*** The user must input a 3-digit number. The program should first check if the number is a prime number. • If it is not a prime number, the user should be prompted to enter another number until a valid prime number is provided. ***2. Sum of Digits Check:*** • Once a valid prime number is entered, calculate the sum of its digits. If the sum of the digits is not divisible by 3, ask the user to enter another prime number until a valid one is found. ***3. Armstrong Number Check:*** • Check entered prime number is Armstrong or not? If Armstrong are found, prompt the user to enter another prime number and repeat the process. **Password Generation:** Concatenate the 1 if entered prime number is Armstrong otherwise 2 with the sum of the digits of the valid prime number to form the secure password. **Example Scenario:** ***Sample Input*** • Enter a 3-digit prime number: 153 Sum of digits of 153 = 9	
2	Write a function to input electricity unit charges and calculate total electricity bill according to the given condition: For first 50 units Rs. 0.50/unit For next 100 units Rs. 0.75/unit For next 100 units Rs. 1.20/unit For unit above 250 Rs. 1.50/unit	

	An additional surcharge of 20% is added to the bill	
3	Write a method to generate a secure code which the sum of all possible palindrome numbers between given two numbers. For Example: Input: 10, 80 Output: 308 Explanation: All palindrome numbers between 10 & 80 are: 11,22,33,44,55,66,77 Password= 11+22+33+44+55+66+77 = 308	
4	Draw the following Patterns for N=5  Right Triangle Star Pattern  Hollow Right Triangle Star Pattern  Pyramid Star Pattern  Hollow Pyramid Star Pattern  Mirrored Right Triangle Star Pattern  Hollow Mirrored Right Triangle Star Pattern  Inverted Pyramid Star Pattern  Hollow Inverted Pyramid Star Pattern  Diamond Star Pattern  Hollow Diamond Star Pattern  Number pattern 18  For N=3 print above pattern  Right Arrow Star Pattern  Left Arrow Star Pattern  ABCDEFEDCBA ABCDE EDCBA ABCD DCBA ABC CBA AB BA A A
5	Write a program that takes an integer n as input and prints the multiplication table of n from n * 1 to n * 10. The output should clearly show each multiplication step.	
6	Write a program to calculate the sum of all integers from 1 to a given number N. The program should take N as input and output the total sum using iteration or recursion.	
7	**Find the GCD of Two Numbers Using Recursion:** Write a recursive function to calculate the Greatest Common Divisor (GCD) of two numbers using Euclid's algorithm. The function should take two integers as input and return their GCD.	
8	**Find the LCM of Two Numbers Using Recursion:** Write a program to compute the Least Common Multiple (LCM) of two numbers using recursion. You may use the relationship LCM(a, b) = |a* b| / GCD(a, b) and a recursive function for GCD.	
9	Write a program to count the number of set bits (1s) in the binary representation of a given integer. The program should efficiently use bitwise operations to perform the task without converting the number to	

	a string.	
10	Write a program that takes a number and a bit position as input and checks whether the bit at that position is set (1) or clear (0). Use bitwise operators to perform the check	
11	Given a number and a position, write a program to toggle (invert) the bit at the given position using bitwise operations. The result should reflect the updated value of the number after flipping the bit.	
12	Write a program to compute the XOR of all numbers from 1 to n using a mathematical pattern (not a loop). Use bitwise XOR properties to achieve an efficient solution.	
13	**Given an array of size n-1 containing unique elements from 1 to n, find the missing number using bit manipulation (preferably XOR approach) without sorting or using extra space.**	
14	**Given an array where all elements repeat twice except two elements that appear only once, write a program to find the two non-repeating elements using bitwise operations in linear time and constant space.**	
15	**Write a program to check if a given number is a power of two using bit manipulation. A number is a power of two if it has exactly one set bit in its binary representation.**	
16	**Given two integers A and B, write a program to count how many bits need to be flipped to convert A to B. Use XOR to find differing bits and count the number of set bits.**	
17	**Write an efficient program to count the total number of set bits in binary representations of all numbers from 1 to n. Optimize the approach using bitwise logic and recursion.**	
18	**Write a program to calculate the square of a number using only bitwise operations and addition. Do not use multiplication, division, or any power functions.**	
19	Write a function to add two integers using bitwise operations only. Avoid using the + or - operators. Implement logic using XOR and AND operations for binary addition.	
20	**Write a program to generate the power set (all subsets) of a given set using bitwise representation. Each subset can be represented by a binary number where each bit indicates inclusion of the corresponding element.**	
21	Sarah is assisting the "MathMinds Club" in creating passwords for their online platform. They have a list of numbers, some stable and some unstable. Define a function that can help Sarah calculate the password according to the given scenario. Scenario: • There are N numbers provided. • A number is stable if each digit appears the same number of times. • A number is unstable if the frequency of its digits is not the same. • The password is computed as the sum of all stable numbers minus the sum of all unstable numbers. • Consider only those numbers in the list that have more than equal to three digits. For example: Input: N=5 List: 12, 1313, 122, 678, 898 Output: Password: 971	
22	Given an array of integers, including possible negative values, you are allowed to modify at most one element by doubling its value. The goal is to find the maximum possible sum of any subarray after making this modification. **Input**: arr = [-2, 1, -3, 4, -1, 2, 1, -5, 4] Expected Output: • Original Maximum Subarray Sum: 6 (achieved from [4, -1, 2, 1]) Maximum Sum After Modification: 10(achieved from [8, -1, 2, 1], where the value 4 is doubled to 8).	

23	**For a given string, generate a pattern based on the following rules:** **Input: A string of characters (e.g., "HAT").** **Output: Generate patterns by replacing characters with the numeric value 1 and process the patterns as described below:** **1. Replace one character at a time with 1:** **o For each character in the string, replace it with 1, keeping the other characters unchanged.** **o Example for "HAT": 1AT, H1T, HA1** **2. Replace two characters at a time with 1:** **o Replace every combination of two characters with 1, keeping the remaining character unchanged.** **o If 1s are consecutive, replace them with their sum (e.g., 11T becomes 2T).** **o Example for "HAT":** **11T → 2T, H11 → H2, 1A1** **3. Replace all characters with 1:** **o Replace all characters in the string with 1.** **o If there are consecutive 1s, sum them up (e.g., 111 becomes 3).** **o Example for "HAT": 111 → 3** **Final Output** **For the string "HAT", the output should be: 1AT, H1T, HA1, 2T, H2, 1A1, 3.**	
24	Given a sorted array arr [] and a target value, the task is to count triplets (i, j, k) of valid indices, such that arr[i] + arr[j] + arr[k] = target and i < j < k. **Examples**: Input: arr[] = [-3, -1, -1, 0, 1, 2], target = -2 Output: 4	
25	**You are given an array prices[] where prices[i] represents the price of a given stock on day i. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Write a program to return the maximum profit you can achieve from this transaction. If no profit is possible, return 0.**	
26	**Find the "Kth" max and min element of an array:** **Given k, find the k-th smallest and k-th largest element in the array. Input: arr = [7, 10, 4, 3, 20, 15], k = 3** **Output: Kth Smallest: 7, Kth Largest: 10**	

27	**7. Sort a binary array** **Input: [0, 2, 1, 2, 0]** **Output: [0, 0, 1, 2, 2] with values 0, 1, and 2 using constant space and one pass (Dutch National Flag algorithm).**	
28	**Find longest consecutive subsequence:** **Return the length of the longest consecutive elements sequence. Input: [1, 9, 3, 10, 4, 20, 2]** **Output: 4 (Sequence: 1, 2, 3, 4)**	
29	Given a number of bits and a number K. In one flip you can toggle exactly K consecutive bits. With only this flip operation available, convert the string into all 1. Input String: 0000110000 and K=3 Following are four flip operations by using which all bits converted into 1's. Flip1- 1110110000 Flip2- 1110110111 Flip3-1111000111 Flip4- 1111111111 If it is not possible to convert all bits into one's then print "IMPOSSIBLE".	
30	**Given a list of non-negative integers, arrange them in such a way that they form the largest possible number. Since the result can be very large, return it as a string in O(N log N) time complexity.** **Example-1** **Input:** **N = 5** **Arr[] = {3, 30, 34, 5, 9}** **Output: 9534330** **Example-2** **Input:** **N = 4** **Arr[] = {54, 546, 548, 60}** **Output: 6054854654**	
31	**11.** Given an array arr[] of size n containing distinct integers within the range [1, n+2], find the two missing numbers from the first n+2 natural numbers. **Constraints:** • The solution must run in **O(N) time** and use **O(1) extra space**. • The array does not contain duplicate values. **Examples:** **Input:** arr[] = [1, 2, 4, 6, 3, 8], n = 6 **Output:** 5, 7	
32	Given a string str of lowercase alphabets and a number k, the task is to print the minimum value of the string after removal of k characters. The value of a string is defined as the sum of squares of the count of each distinct character present in the string. Return the minimum possible required value. **Examples:** **Input:** str = "abccc", k = 1 **Output:** 6	

	Input: str = “aabcbcbcabcc”, k = 3 **Output**: 27 **Expected Time Complexity:** O(n+klog(p)) **Note**: Here n is the length of string and p is number of distinct alphabets and k number of alphabets to be removed.	
33	**13.** Given a non-negative integer **S** represented as a string, remove **K** digits from the number so that the new number is the smallest possible. **Note :** The given *num* does not contain any leading zero. **Expected Time Complexity:** O(|S|). **Example 1:** **Input:** **S = "149811", K = 3** **Output:** **111** **Example 2:** **Input:** **S = "1002991", K = 3** **Output:** **21**	
34	u are given a two-dimensional grid board[][] of size n * m consisting of English letters and a string target. Your task is to determine whether the target word can be formed by sequentially connecting letters from the grid. You may move to adjacent cells **horizontally or vertically** (not diagonally), and **a cell may not be reused** once it is part of the current path. **Examples:** **Input:** **board[][] = [['C', 'A', 'T'], ['R', 'A', 'K'], ['T', 'O', 'N']],** **target = "CART" Output: true Explanation:** **You can trace the word "CART" through the path: C → A → R → T (moving horizontally and vertically, without repeating cells).**	
35	**15.** Given an encoded string **s**, the task is to decode it. The encoding rule is: • **k[encodedString],** where the **encodedString** inside the square brackets is being repeated exactly **k** times. Note that **k** is guaranteed to be a positive integer, and encodedString contains only lowercase english alphabets. **Note:** The test cases are generated so that the length of the output string will never exceed 10^5. **Examples:** **Input:** s = "1[b]" **Output:** "b" **Input:** s = "3[b2[ca]]" **Output:** "bcacabcacabcaca"	
	TEACHING HOURS	**30 HRS**

Course Code: CNC0401	Course Name: ARTIFICAL INTELLIGENCE AND CYBER ETHICS	L	T	P	C
Course Offered in: All B.Tech Branches		2	0	0	NA

Pre-requisite: Basic understanding of AI, Cybercrime, Computer System and Ethics

Course Objectives:

The course aims to foster critical thinking about ethical issues, promote responsible use of technology, and ensure students can identify, analyze, and address ethical dilemmas in Artificial Intelligence and cyber domains.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Learn key principles of AI ethics, summarizing ethical considerations and applications in AI development and deployment.	K2
CO2	Apply policies and framework for Fairness in AI and Machine Learning.	K3
CO3	Apply privacy and security concepts, risk management and regulatory compliance in the field of AI and Cyber Security.	K3
CO4	Understand the nature of cybercrimes, the principles of intellectual property rights (IPR), and the legal measures necessary to address and prevent these issues.	K2

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	PSO1	PSO2	PSO3
CO1	-	1	-	-		1	2	-	-	-	2	1	1	1	-
CO2	2	3	2	-	2	1	2	-	2	1	2	2	1	1	2
CO3	2	3	2	1	2	3	3	-	2	2	2	2	2	1	2
CO4	2	2	-	-	1	3	3	-	2	2	2	2	1	1	2

Course Contents / Syllabus

UNIT 1	AN OVERVIEW TO AI ETHICS	6 HOURS

Definition of AI Ethical principles in AI, Sources of AI data, Legal implications of AI Security Breaches, Privacy and AI Regulations, Key Principles of Responsible AI, Transparency and Accountability, Dual-Use Dilemma, Human-Centric Design, Introduction to Cyber Laws and Ethics, Historical Development of Cyberlaws, Legal frameworks.

UNIT 2	FAIRNESS AND FAVORITISM IN MACHINE LEARNING	8 HOURS

Introduction to Fairness and Bias in AI, Types of Fairness and Bias, Impact of Bias and Fairness in AI, Techniques for Measuring Fairness and Bias, Techniques for Mitigating Bias, Current Policies and Frameworks for Fairness in AI, Bias in Data Collection, Fairness in Data Processing, Generative AI, Types of Bias in Generative AI.

UNIT 3	AI ETHICS AND CYBERSECURITY PRINCIPLES	8 HOURS

Importance of Privacy and Security in AI, AI specific Security Tools and Software, Privacy-Preserving Machine Learning (PPML) and Privacy-Preserving Data Mining (PPDM),Risk Management: Risk Assessment and Incident Response, Regulatory Compliance: GDPR, HIPAA, Case Studies: Implementation of AI Ethics guidelines and best practices in engineering projects.

UNIT 4	CYBERCRIMES, IPR AND LEGAL MEASURES	8 HOURS

Types of Cybercrimes and their Impact, Legal measures for Cybercrime Prevention and Prosecution, IPR: Copyrights, Trademarks, Patents, and Trade Secrets, Ethical Implications of Intellectual Property, Cyber Security and Privacy Issues, Cyber Crime Investigations and Digital Evidence Handling, Overview of Indian Cyber Laws (IT Act 2000 and Amendments), Comparative Overview: Indian vs Global Cyber Laws, Case Study: The ATM Heist – Cosmos Bank Cyber Attack (India, 2018).

Total Lecture Hours	30 HOURS

Textbook:

Book Details	Authors Name
Artificial Intelligence: A Guide for Thinking Humans , Penguin Books, 2019.	Melanie Mitchell
Cyber Ethics: Morality and Law in Cyberspace ,7th Edition (2023).	Richard Spinello, Jones & Bartlett Learning

Reference Books:

Book Details	Authors Name
Artificial Intelligence and Ethics , BPB Publications, 2023.	S. B. Kishor, Debajit Biswas
Cyber Security and Cyber Laws ,Cengage India, 2022.	Alfred Basta, Nadine Basta, Sattwik Panda,

Self-Study Content Links:		Duration (in Hrs)
MOOC'S	https://swayam-plus.swayam2.ac.in/courses/course-details?id=P_INTEL_02	64 HRS
VIDEO LECTURES	https://www.youtube.com/watch?v=hVJqHgqF59A https://www.youtube.com/watch?v=VqFqWIqOB1g https://www.youtube.com/watch?v=O5RX_T4Tg24 https://www.youtube.com/watch?v=RJZ0pxcZsSQ	5.2m 6.52m 45m 5.02m
WEB REFERENCES	CCS345 Ethics and AI Lecture Notes | PDF | Artificial Intelligence | Intelligence (AI) & Semantics	