NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH 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
Bachelor of Technology

Computer Science and Engineering (Cyber Security)

Second Year

(Effective from the Session: 2026-27)

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA

(An Autonomous Institute)

Computer Science and Engineering (Cyber Security)

Evaluation Scheme

SEMESTER -III

Sl. 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	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	CCSCY0301	Cryptography	Mandatory	2	0	0	30	20	50		50		100	2
6	CCSE0304	Computer Architecture & Parallel Processing	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	CCSCY0351	Cryptography 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	CCSE359	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)	*MOOCs											
		TOTAL		**17**	**1**	**18**	**240**	**160**	**350**	**225**	**400**	**225**	**1200**	**25**

*** List of MOOCs (Infosys Springboard) Based Recommended Courses for Second Year B. Tech Students**

S. No.	Subject Code	Course Name	University/ Industry Partner Name	N. of Hours	Credits
1.	CMC0015	Data Structures and Algorithms using Python - Part 1	Infosys Wingspan (Infosys Springboard)	29h 27m	2
2.	CMC0030	Network Fundamentals	Infosys Wingspan (Infosys Springboard)	37h 57m	3
3.	CMC0029	Cyber Security Essential	NASSCOM	70h	4

PLEASE NOTE:

A 3-4 weeks 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 in 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)

Computer Science and Engineering (Cyber Security)

Evaluation Scheme

SEMESTER IV

Sl. 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	CCSCY0401	Computer Network and Network Programming	Mandatory	3	0	0	30	20	50		100		150	3
4	CCSE0402	Software Engineering	Mandatory	3	0	0	30	20	50		100		150	3
5		Departmental Elective-I	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	CCSCY0451	Computer Network and Network Programming Lab	Mandatory	0	0	2				25		25	50	1
9	CCSCY0455	Incident Response Management and Ethical Hacking	Mandatory	0	0	6				50		100	150	3
10	CCSE0459	Mini Project	Mandatory	0	0	2				50			50	1
11	CCSCC0452	Problem Solving Approaches	Mandatory	0	0	2				50			50	1
12	CNC0402/ CNC0401	Environmental Science / Artificial Intelligence and Cyber Ethics	Compulsory Audit	2	0	0			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 (Infosys Springboard) Based Recommended Courses for Second year B. Tech Students**

S. No.	Subject Code	Course Name	University/ Industry Partner Name	N. of Hours	Credits
1.	CMC0052	Database and SQL	Infosys Wingspan (Infosys Springboard)	31h 11m	2.5
2.	CMC0053	Programming Using Java	Infosys Wingspan (Infosys Springboard)	113h 2m	4
3.	CMC0054	Networking Basics	NASSCOM	22 h	1.5

PLEASE NOTE:
A 3-4 weeks Internship shall be conducted during summer break after semester-II and will be assessed during semester-III
Compulsory Audit (CA) Courses (Non-Credit - CNC0402/CNC0401)

- ➢ All Compulsory Audit Courses (a qualifying exam) do not require any credit.
- ➢ The total and obtained marks are not added in 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)

List of Department Electives I: -

Sr. No	Subject code	Subject Name	Types of Subjects	Bucket Name	Branch	Sem.
1	CCS0411	Introduction to Cloud Computing	DE-I	Cloud Computing	CSE(CYS)	IV
2	CCSAI0411	Data Analytics	DE-I	Data Science	CSE(CYS)	IV
3	CCSE0411	Python Web Development with Django	DE-I	Full stack Development	CSE(CYS)	IV
4	CSCY0412	Cyber Threat Intelligence	DE-I	Cyber Security	CSE(CYS)	IV

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: B.Tech III 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 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	PSO3
CO1	3	3	-	2	3	1	-	3	2	2	2	1	1	2
CO2	3	3	3	3	3	2	1	3	2	2	2	2	2	2
CO3	3	3	2	1	-	-	-	1	2	2	2	3	2	2
CO4	-	-	-	-	1	2	-	3	2	2	-	-	-	-
CO5	-	2	1	-	1	2	-	3	2	2	1	-	-	-

Course Contents / Syllabus

Section-A

Module 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.

Module 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

Module 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

Module-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.		
Module-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: 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

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: B. Tech. Second Year: Semester-III/IV AI/AIML/AI (TWIN)/AIML(TWIN)/CYS/DS/CS/CSE/CSE(TWIN)/CSE R/IT/ IT- Twin/ M. Tech. (Int.)/MCT		**3**	**1**	**0**	**4**

Pre-requisite: B. Tech. First Year Syllabus

Course Objectives:

1. The objective of this course is to familiarize the students with concepts of Probability and statistical techniques.
2. It aims to equip the students with adequate Knowledge of statistics that will enable them in formulating Problems 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	PSO3
CO1	3	2	2	3	1	1	-	1	2	2	2	3	1	1
CO2	3	2	2	3	1	1	-	1	2	2	2	3	2	2
CO3	3	2	1	2	-	-	-	-	1	2	2	2	1	1
CO4	3	2	2	3	1	1	-	1	2	2	2	3	2	2
CO5	3	2	2	3	1	1	-	1	2	2	2	3	2	2

Course Contents / Syllabus

Module 1	Statistical Techniques-I	**9 hours**

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

Module 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.

Module 3	Probability and Random Variable	**9 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)

Module 4	Expectation and Probability Distributions	**9 hours**

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

Module 5	Hypothesis Testing and Control Charts	**9 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	**45hours**

Textbook:

1. Textbook of Engineering Mathematics- IV Bali, N.P.
2. Advanced engineering mathematics Jain, R.K.
3. Higher engineering mathematics Grewal, B.S.
4. Statistical methods Gupta, S.P.
5. Advanced engineering mathematics ZILL, DENNIS G.

Reference Books:

1. Introduction to Probability Models Ross, Sheldon M
2. Probability, Random Variables and Stochastic Processes Papoulis, Athanasios
3. Advanced engineering mathematics Kreyszig, E

Self-Study Content Links:		Duration (in Hrs)
MOOCs	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
Web Reference	https://www.geeksforgeeks.org/maths/probability-and-statistics/	3 hrs
	https://www.geeksforgeeks.org/machine-learning/statistics-for-machine-learning/	2 hrs

Course Code: CCSE0303A	**Course Name: Operating Systems**	**L**	**T**	**P**	**C**
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(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

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

Course Contents / Syllabus

Unit 1	**Fundamentals & Shell scripting**	**8 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**	**8 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, Belady'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	**32 hours**

Textbook:

S.No	**Book Title**	**Author**
1.	" Operating System Concepts Essentials" , Willey Publication,10th Edition,2018.	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne

2.	"A practical guide to Linux: Commands, Editors and Shell Programming", CreateSpace Independent Publishing Platform, 4th Edition,2017.	Marks G. Sobell
3.	"LINUX for beginners", 1stEdtion,2014.	Jason Cannon
Reference Books:		
S.No	**Book Title**	**Author**
1.	"Operating Systems: Internals and Design Principles", Pearson Education , 9th Edition, 2019.	William Stallings
2.	"Operating System: A Design-oriented Approach" , McGraw Hill Education,2017.	Charles Patrick Crowley
3.	"Learning Linux Shell Scripting", Packt Publishing ,2nd Edition 2018.	Ganesh Naik

Self-Study Contents Links:		**Duration (in hours)**
MOOCs	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013843258047987712344443_shared/overview	**29 hours**
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=pPAKs7tT8sw https://www.youtube.com/watch?v=hBPYP0ZEvS8 https://www.youtube.com/watch?v=UeVQxzlgd10 https://www.youtube.com/watch?v=9cuwgiHQyoI https://www.youtube.com/watch?v=-NONm-Jq34Y&t=25s https://www.youtube.com/watch?v=5uyGn147rr0 https://www.youtube.com/watch?v=utYC2tKDdoQ https://www.youtube.com/watch?v=Cuo8GTklgoE https://www.youtube.com/watch?v=YHQxp-XduS0 https://www.youtube.com/watch?v=yB8sm-pwvZo 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 hours**
Web References	https://www.geeksforgeeks.org/operating-systems/operating-systems/ https://studyhubzone.com/operating-systems-notes/	**4 hours**

Course Code: CCSE0301	Course Name: DATA STRUCTURES AND ALGORITHMS-II	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(MCT)		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

CO PO	Course Outcome	Bloom's Level
CO 1	Apply tree structures to solve specific problems using various tree operations and algorithms.	K3
CO 2	Analyse the graph data structure and evaluate the efficiency of its operations for problem solving.	K4
CO 3	Evaluate dynamic programming solutions for efficient problem-solving across diverse contexts.	K4
CO 4	Apply efficient backtracking and branch –and –bound techniques across diverse problem-solving scenarios.	K3
CO 5	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	PSO3
CO1	3	3	2	1	1	2	1	1	2	1	-	2	1	2
CO2	3	3	2	1	1	2	1	1	2	1	-	2	1	2
CO3	3	3	2	2	2	2	1	1	2	2	-	2	1	2
CO4	3	3	3	2	2	2	1	1	2	1	-	2	1	2
CO5	3	3	3	2	2	2	1	1	2	2	-	2	1	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: Graphs	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

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 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:

S.No.	Book Details	Author
1	"Data Structures and Algorithms in Python: An Indian Adaptation", 1st Edition, 2021	Michael T. Goodrich, Roberto Tamassia

2	"Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd, 2nd Edition, 2017.	Lipschutz
3	"Introduction to Algorithms", Printice Hall of India, 4th Edition, 2022	Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest,

Reference Books:

S.No.	**Book Details**	**Author**
1	"Data Structure Using C", Oxford University Press, 2nd Edition, 2014.	Reema Thareja
2	"Data Structure Using C", Pearson Education India, 2nd Edition,2011.	AK Sharma
3	"C and Data structure", Wiley Dreamtech Publication, 1st Edition, 2004.	P. S. Deshpandey

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013842032404848640104 70_shared/overview https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013094439914094592258 4_shared/overview	**18 hours** **3 hours**
Video Links	https://www.youtube.com/watch?v=zWg7U0OEAoE&list=PLBF3763AF2E1C572F https://www.youtube.com/watch?v=LQx9E2--p5c&pp=ygUMYXJyYXlzIG5wdGVs https://www.youtube.com/watch?v=K7VIKlUdo20&pp=ygUPbGluayBsaXN0IG5wdGVs https://www.youtube.com/watch?v=g1USSZVWDsY&list=PLB3CD0BBB95C1BF09&index=2&pp=iAQB https://www.youtube.com/watch?v=THMyk2_p530&pp=ygUccXVldWUgZGF0YSBzdHJ1Y3R1cmUgICBucHRlbA%3D%3D https://www.youtube.com/watch?v=_VV9v41FIq0&pp=ygUZZGl2aWRlIGFuZCBjb25xdWVyICBucHRlbA%3D%3D https://www.youtube.com/watch?v=ARvQcqJ_-NY&list=PLfFeAJ-vQopt_S5XlayyvDFL_mi2pGJE3	**40 Hours**
Web References	https://www.msajce-edu.in/academics/it/LectureNote/CD3291-LN.pdf	-

Course Code: CCSCY0301	Course Name: Cryptography	L	T	P	C
Course Offered in: CSE(CYS)		2	0	0	2

Prerequisite: Basic knowledge of Computer Fundamental

Course Objectives:

The course introduces the fundamental principles, mathematical foundations, and security goals of cryptography. It enables students to understand classical and modern encryption techniques, cryptanalysis methods, and symmetric key cryptographic systems. The course also provides knowledge of DES, cipher modes of operation, and the transition from symmetric to public key cryptography for secure communication systems..

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Explain the fundamentals of cryptography, security goals, attacks, and number theoretic concepts used in cryptographic systems	K2
CO2	Apply classical substitution cipher techniques for encryption, decryption, and basic cryptanalysis.	K3
CO3	Analyze transposition and stream ciphers using cryptographic and cryptanalytic methods.	K4
CO4	Demonstrate the working principles of symmetric key cryptography, Feistel structure, and DES concepts.	K3
CO5	Evaluate cryptographic modes, key management challenges, and the transition from symmetric to public key cryptography.	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	3	2	-	2	-	1	-	2	2	3	2	2
CO2	3	3	2	2	2	2	1	1	1	1	2	2	2	3
CO3	3	3	2	3	2	2	2	1	1	2	2	3	2	3
CO4	3	2	3	2	2	1	2	1	1	1	2	2	2	2
CO5	3	3	2	2	2	2	1	-	1	2	2	3	2	2

Course Contents / Syllabus

Unit 1	**Introduction to Cryptography & Number Theory**	**8 hours**

Definition, goals and terminology — confidentiality, integrity, authentication, non-repudiation. History of cryptography: ancient to modern era; Kerckhoffs' principle. Mathematical preliminaries: modular arithmetic- groups, rings, GCD, Euler's theorem, Fermat's little theorem Prime numbers and primality testing; the Chinese Remainder Theorem. Cryptosystem model: plaintext, ciphertext, key space, encryption/decryption functions. Types of attacks: ciphertext-only, known-plaintext, chosen-plaintext, chosen-ciphertext. Introduction to information theory: entropy, perfect secrecy (Shannon's theorem)

Unit 2	**Classical Substitution Ciphers**	**8 hours**

Caesar cipher — encryption, decryption, brute-force attack. Monoalphabetic ciphers: shift, affine, and keyword substitution ciphers. Cryptanalysis of monoalphabetic ciphers: letter frequency analysis. Vigenère cipher: polyalphabetic substitution, key repetition. Kasiski test and index of coincidence for breaking Vigenère cipher. Playfair cipher: key square construction, digraph encryption rules. Hill cipher: matrix-based encryption, decryption via matrix inverse. Nature of key stream: random number, TRNG, PRNG, CSPRNG, One-time pad: concept, perfect secrecy, practical limitations

Unit 3	**Classical Transposition & Stream Ciphers**	**7 hours**

Rail fence cipher: zigzag permutation, depth analysis. Columnar transposition cipher: single and double columnar transposition. Route cipher and scytale cipher. Cryptanalysis of transposition ciphers: anagramming and pattern recognition. Stream ciphers: concept, synchronous vs self-synchronizing; keystream generation

Unit 4	**Symmetric Key Principles & Bridge to Modern Cryptography**	**7 hours**

Block cipher design principles: confusion, diffusion (Shannon), substitution-permutation networks. Feistel cipher structure: rounds, sub-keys, encryption and decryption symmetry Introduction to DES: key scheduling, S-boxes, P-box, 16-round structure (conceptual). Modes of operation: ECB, CBC, CFB, OFB, CTR — use cases and weaknesses. Key management and distribution challenges in symmetric systems. Introduction to public key cryptography: limitations of symmetric systems

Total Lecture Hours	**30 hours**

Textbook:

1. Cryptography and Network Security: Principles and Practice by William Stallings, 7th Edition Pearson

Reference Books:

1. Cryptography: Theory and Practice by Douglas Stinson, 4th Edition, CRC Press, Taylor & Francis

NPTEL/ YouTube/ Faculty Video Link:

Self-Study Content Links:		Duration (in Hrs)
MOOCs	1. Fundamentals of *Cryptography* (2h18m) 2. *Cryptography* with Python (2h30m) 3. Python Case Study - *Cryptography* (5h47m) 4. Ethical Hacker: *Cryptography* Concepts (1h12m)	11h50m
Video Lectures	1. https://www.youtube.com/watch?v=Q-HugPvA7GQ&list=PL71FE85723FD414D7 2. https://onlinecourses.nptel.ac.in/noc22_cs90/preview 3. OpenSSL / AES Cryptography Library /GnuPG /	59 m
Web References	1. https://www.geeksforgeeks.org/computer-networks/cryptography-introduction/ 2. https://www.tutorialspoint.com/cryptography/index.htm	5 h

Course Code: CCSE0304	Course Name: Computer Architecture & Parallel Processing	L	T	P	C
Course Offered in: CSE/CSE-R/IT/CSE(DS)/CSE(CYS)/CSE(MCT)		3	0	0	3

Pre-requisite: Basic knowledge of computer systems, Logic gates and their operations.

Course Objectives:

This course aims to provide a comprehensive understanding of computer organization and architecture, covering processor design, memory systems, and control units. It explores advanced topics such as cache coherence, parallel architectures, and scalable shared memory systems. Students will gain insight into system performance through instruction execution, arithmetic operations, synchronization, and interconnect strategies.

Course Outcome: After completion of the course, the student will be able to		**Bloom's KnowledgeLevel (KL)**
CO1	Explain the basic structure and operation of a digital computer system.	K2
CO2	Analyze the design of arithmetic & logic unit and understand the fixed point and floating-point arithmetic operations.	K4
CO3	Implement control unit techniques and the concept of pipelining.	K3
CO4	Analyze parallel architectures and coherence protocols by exploring memory hierarchy, cache coherence mechanisms, and multiprocessor design techniques to ensure correctness and performance in parallel systems	K4
CO5	Analyze scalable shared memory systems by evaluating directory coherence protocols, memory consistency models, synchronization mechanisms, and interconnect strategies to ensure system.	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	PSO3
CO1	1	1	1	-	-	-	-	-	-	-	-	1	1	2
CO2	2	2	2	-	-	-	-	-	-	-	-	1	2	2
CO3	2	2	2	-	-	-	-	-	-	-	-	1	2	2
CO4	2	2	2	-	-	-	-	-	-	-	-	1	2	1
CO5	2	2	2	-	-	-	-	-	-	-	-	1	2	1

Course Contents / Syllabus

Unit 1	**Introduction**	**9 hours**

Computer Organization and Architecture, Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration and its types. Register, bus and memory transfer. Processor organization, general registers organization, stack organization and addressing modes.

Unit 2	**ALU Unit**	**9 hours**

Arithmetic and logic unit: Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic &logic unit design. IEEE Standard for Floating Point Numbers.

Unit 3	**Control Unit**	**9 hours**

Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc.), micro- operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Complex Instruction Set Computer, Pipelining. Hardwire and microprogrammed control, Concept of horizontal and vertical microprogramming, Flynn's classification.

Unit 4	**Introduction to Parallel Architectures**	**9 hours**

Introduction to Parallel Architectures, Parallel Programming models and Architectures, Memory Hierarchy-Cache and Virtual memory, Overview of Cache coherence, Coherence Protocols- Snooping, Directory based protocols, VI protocol, MSI, MESI, Dragon protocol and Correctness of coherence protocols- Types of cache misses, update vs invalidate protocol, Snoop based multiprocessor design, Single and multi-level cache with atomic and bus split transaction bus

Unit 5	**Parallel Systems**	**9 hours**

Scalable shared memory systems: Directory coherence protocols- Memory based, cache based, correctness, Case study: Origin- Architecture, protocol, correctness; Sequent NUMA Q- Architecture, protocol, correctness, Memory consistency models- Sequential, Relaxed consistency models, Synchronization- LL-SC, point to point, barrier synchronization, Interconnects- Introduction, Topologies, routing, flow control

	Total Lecture Hours	**45 hours**

SNO	Book Title	Author
1	Computer System Architecture", 3rd Edition, Pearson Publication, 2007.	M. Mano
2	Computer Architecture and Organization, Tata McGraw Hill, Third	John P. Hayes

	Edition, 1998.	
3	Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.	William Stallings
4	Parallel Computer Architecture. Morgan- Kaufmann publishers.	D. E. Culler and J. P. Singh with A. Gupta
5	Computer Architecture: A Quantitative Approach. Morgan-Kaufmann publishers.	J.L. Hennessy and D. A. Patterson
6	Parallel Computer Organization and Design. Cambridge University Press.	M. Dubois, M. Annavaram, Per Stenstrom,

Reference Books:

SNO	**Book Title**	**Author**
1	Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012	Carl Hamacher, ZvonkoVranesic, SafwatZaky
2	Advanced Microprocessors and Peripherals, TM	Ray A K, Bhurchandi K M
3	"Computer Architecture & Parallel Processing" Mcgraw Hill Education	Kai Hwang

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs168 https://nptel.ac.in/courses/106104024	12 weeks 12 weeks
Video Lectures	https://www.youtube.com/watch?v=WLgXUPOjKEc https://www.youtube.com/watch?v=BPhWlFIU1rc https://www.youtube.com/watch? v=6R7JDkpG1Wk&list=PLrjkTq l3jnm8HbdMwBYIMAd3UdstWChFH https://www.youtube.com/watch?v=nxryfWg5Hm4 https://www.youtube.com/watch?v=txAyA_UozmM https://www.youtube.com/watch?v=L9X7XXfHYdU&list=PLxCzCO Wd7aiHMonh3G6QNKq53C6oNXGrX	2 hours
Web References	https://media.geeksforgeeks.org/courses/syllabus/db29b4a6e537cb1a4c4a9119e1ca565f.pdf	-

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

Prerequisite: Basic knowledge of computer fundamentals, C programming, Data structures and Computer organization.

Course Objectives: The course aims to provide hands-on experience with Linux and shell programming, while the lab focuses on implementing and analyzing key OS algorithms and simulating modern operating 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 Mapping	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	1	1	1	3	-	-	2	2	-	3	2	1	2
CO2	3	3	3	2	2	-	-	2	2	-	2	2	2	2
CO3	3	2	3	2	3	-	-	2	2	-	3	2	3	2

List Of Practical's (Indicative & Not Limited To)		**CO**
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)** v. Use redirection and piping in commands. vi. File compression and archiving using tar, gzip, zip, unzip etc. vii. Process commands ps,kill, killall,nice, pgrep, top,htop etc. viii. Network commands ifconfig, ping, netstat, host,ip route etc. ix. 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. vi. Write a shell program to Print Numbers 1 to 10 using for loop.	CO1
5	**Shell Scripting Programming (continued)** vii. Write a shell script to display the digits which are in odd position in a given 5-digit number. viii. Write a shell program to search for a given number from the list of numbers provided using binary search method. ix. Write a shell program to concatenate two strings and find the length of the resultant string. x. Write a shell script to find the smallest of three numbers. xi. 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.	CO3
13	Implement the Producer–consumer problem using semaphores.	CO3
14	Design a code and implement the Dinning Philosopher problem.	CO3
15	Implement Banker's algorithm of Deadlock Avoidance.	CO3
16	Execute an algorithm for Deadlock Detection.	CO3
17	Implement Contiguous memory variable size partition scheme.	CO3
18	Simulate the First-Fit contiguous memory allocation technique.	CO3
	Total Hours: 60 hrs.	

LAB Course Code: CCSE0351	**LAB Course Name:** Data structure and Algorithms lab-II	**L**	**T**	**P**	**C**
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(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

SNO	**Course Outcome**	**Bloom's Level**
1.	Implementation of tree data structures for basic operations like insertion, deletion, searching and traversal	K3
2	Implementation of algorithms based on graph data structures for solving real world problems.	K3
3	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 Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	1	1	2	1	1	2	1	-	2	1	2
CO2	3	3	2	1	1	2	1	1	2	1	-	2	1	2
CO3	3	3	2	2	2	2	1	1	2	2	-	2	1	2

List Of Practical's (Indicative & Not Limited To)	**Mapped CO**
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
Total Hours: 60 hrs.	

LAB Course Code: CCSCY0351	**LAB Course Name: Cryptography** Lab	**L**	**T**	P	**C**
Course Offered in: CSE (CYS)		0	**0**	2	**1**

Pre-requisite:

Course Objectives: The course provides practical knowledge of classical and modern cryptographic techniques through implementation of encryption, decryption, and cryptanalysis methods. It enables students to design and analyze secure communication systems using symmetric cryptography and authentication mechanisms.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Implement classical substitution and transposition of cryptographic algorithms along with basic cryptanalysis techniques.	K3
CO2	Demonstrate stream ciphers, one-time pad, Feistel structure, and DES-based symmetric encryption techniques through practical implementations.	K3
CO3	Design and analyze secure cryptographic applications using encryption, authentication, and security evaluation methods.	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	PSO3
CO1	3	3	2	2	3	-	-	1	1	1	2	3	2	3
CO2	3	2	3	2	3	-	-	1	1	1	2	2	2	3
CO3	2	3	3	3	2	1	-	2	2	2	3	3	2	2

List Of Practical's (Indicative & Not Limited To)	CO Mapping
1. Implement encrypt, decrypt, and brute-force key recovery using Caesar Cipher.	CO2
2. Implementation of modular arithmetic and cryptanalysis using frequency analysis with Affine Cipher.	CO2
3. Use Frequency Analysis tool to build a program for performing letter-frequency analysis on ciphertext and guessing monoalphabetic keys.	CO2
4. Implement Vigenère Cipher and breaking via Kasiski Test and Index of Coincidence.	CO2
5. Implement Playfair Cipher using 5×5 key square generation and digraph encryption/decryption.	CO2
6. Implement Hill Cipher using NumPy for matrix-based encryption and compute matrix inverse mod 26 for decryption.	CO2
7. Implement Monoalphabetic Substitution Cipher and break it using letter frequency analysis.	CO2
8. Implementation of Rail Fence and Columnar Transposition Ciphers and combined ciphers.	CO3
9. Simulate a 4-bit LFSR, generate keystream, encrypt plaintext, and compute linear complexity.	CO3
10. Implement One-Time Pad demonstrating perfect secrecy and practical limitations.	CO3
11. Design a 4-round custom Feistel Cipher and verify encrypt/decrypt symmetry.	CO3
12. Demonstrate ECB vs CBC mode encryption on the same plaintext image and compare output patterns.	CO3
13. Implementation of the Data Encryption Standard (DES) Algorithm for Encryption and Decryption.	CO3
14. Mini Project: Design a secure messaging simulation combining a classical cipher (Vigenère) with a hash-based message authentication code and present security analysis.	CO3
	Total Hours: 30 hrs.

Course Code: CCSE0355	Course Name: Object Oriented Techniques using Java	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(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	PSO1	PSO2	PSO3
CO1	3	3	3	3	2	2	1	-	1	-	2	2	2	2
CO2	3	3	3	3	2	2	1	-	1	1	2	2	1	2
CO3	3	3	3	3	3	2	2	-	2	1	2	3	1	1
CO4	3	3	3	3	3	2	2	1	2	1	2	2	2	2
CO5	3	3	3	3	3	2	2	1	2	1	2	2	2	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 Life-Cycle, 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 Managersand 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:

S.NO	Book Title	Author

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

Reference Books:

S.NO	Book Title	Author
1.	“Core Java Volume I – Fundamentals”, Prentice Hall	Cay S. Horstmann
2.	” Effective Java”, Addison Wesley	Joshua Bloch
3.	Java - The Complete Reference”, McGraw Hill Education 12th edition	Herbert Schildt

Self-Study Content Links:		Duration (in Hrs)
MOOCs	https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0125409722749255681063_shared/overview	46h 13m
Video Lectures	https://www.youtube.com/watch?v=iw1Xf_33YM0&list=PLQEaRBV9gAFujcBWJhBT2XXsuMlIfETBy	30 h
Web References	1. https://www.geeksforgeeks.org/dsa/introduction-of-object-oriented-programming/ 2. https://en.wikipedia.org/wiki/Object-oriented_programming	20 m 58 m

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 NumberPalindrome 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 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.	CO1
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: • 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
46	Java program to illustrate Multiple catch blocks	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 toanother 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	ava 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**

SYLLABUS- SEMESTER IV

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

C I O-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	-	2	3	1	-	3	2	2	2	-	-	-
CO2	3	3	3	3	3	2	1	3	2	2	2	-	-	-
CO3	3	3	2	1	-	-	-	1	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 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 I**	**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: 56 Hours

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

Self Study Content Link:		Duration(in hours)
MOOCs	nptel.ac.in/courses/109104107	8 weeks
Video Lectures	Employability skills - YouTube	33 m
Web Reference	Employability Skills Self-Management Guide | PDF | Goal | Self Esteem	-

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)/ **CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(MCT)**		**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: After completion of the course, the student will be able to		**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	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	3	3	2	1	1	1	2	1	2	3	1	2
CO2	3	2	3	3	2	2	1	2	1	2	2	3	2	2
CO3	3	3	2	3	3	2	-	2	1	-	2	3	3	2
CO4	3	2	2	2	2	2	1	-	1	1	1	3	3	2
CO5	2	2	2	2	3	2	1	-	1	2	2	3	3	2

Course Contents / Syllabus

Unit 1	**Introduction about the Database Conceptual Designing**	**8 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**	**8 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 Lecture Hours	**32 hours**

Textbook:

S.No	**Book Title**	**Author**
1.	Database Concepts, McGraw Hill, 7th Edition, 2020	Avi Silberschatz Henry F. Korth S. Sudarshan
2.	Fundamentals of Database Systems, Addision Wesley, 7th edition, 2016	R. Elmasri and S. B. Navathe

Reference Books:

S.No	Book Title	Author
1.	Database Systems: A practical Approach to Design, Implementation and Management. Pearson Education,3rd Edition, 2007.	Thomas Connolly , Carolyn Begg
2.	NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software, Ted Hills, 1st Edition,2016.	Ted Hills

Self-Study Contents Links:		**Duration (in Hours)**
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 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	30 hours
Web references	https://www.geeksforgeeks.org/dbms/dbms/ https://www.scribd.com/document/666843210/Database-Management-System-DBMS-HandWritten-Notes	2 hours

Course Code: CCSCY0401	Course Name: Computer Network and Network Programming	L	T	P	C
Course Offered in: CSE(CYS)		3	0	0	3

Prerequisite: The student should have a basic knowledge of data communication and programming.

Course Objectives:

The objective of the course is to present an introduction of TCP/IP protocol, packet switching and message switching, sliding window protocol, CDMA, network layer protocols (IPv4, ARP, RARP), routing, TCP and UDP, congestion control, quality of service, and network applications such as DNS, FTP, TELNET, and remote logging.

Course Outcome: After completion of this course students will be able to		Bloom's Knowledge Level (KL)
CO1	Build an understanding of the Layered Architecture of computer networking and the physical layer.	K2
CO2	Understand the properties of link and network layer and analyze the solutions for error control, flow control, and addressing, for networks.	K4
CO3	Understand the duties of the transport layer and the addressing and functions of sockets.	K2
CO4	Implement and analyze network connections using programming skills.	K4
CO5	Understand and analysed the different protocols used at application layer.	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	PSO3
CO1	3	2	2	2	2	1	1	-	-	1	2	2	1	1
CO2	3	3	2	3	3	1	2	-	-	1	2	3	2	1
CO3	3	3	3	3	3	2	2	-	1	2	2	3	3	2
CO4	3	3	3	2	3	2	2	-	2	2	1	3	3	2
CO5	3	2	2	3	2	2	1	1	1	2	2	2	3	2

Course Contents / Syllabus

Unit 1 Introduction to Networking — **10 hours**

Goals and applications of networks, Categories of networks, Organization of the Internet, ISP, The OSI reference model, TCP/IP protocol suite, Network devices and components, Mode of communications, Physical Layer: Network topology design, Types of connections, LAN, MAN and WAN, Transmission media, Signal transmission and encoding, Network performance and transmission impairments, multiplexing, IEEE standards.

Unit 2 Data Link and Network Layer — **9 hours**

Framing, Error Detection and Correction, Flow control (Elementary Data Link Protocols, Sliding Window protocols). Medium Access Control and Local Area Networks: Channel allocation, Multiple access protocols, LAN standards, Link layer switches & bridges. Network Layer: Point-to-point networks, Logical addressing, Basic internetworking (IP, CIDR, ARP, RARP, DHCP, ICMP), IPv4, IPv6, Routing, forwarding and delivery, Static and dynamic routing, Routing algorithms and protocols, Congestion control

Unit 3 Transport Layer — **9 hours**

Process-to-process delivery, Transport layer protocols (UDP and TCP), Connection management, Flow control and retransmission, Window management, TCP Congestion control, and Quality of service. Introduction to Socket Programming: Introduction to sockets, network programming concepts and applications, client-server architecture and communication model, socket address structures, socket APIs and system calls, TCP socket programming, UDP socket programming, unicast communication, broadcast communication and multicast communication using sockets, TCP three-way handshake and connection termination, concurrent client-server communication concepts, file transfer using TCP sockets, error handling in socket programming, and packet analysis using Wireshark.

Unit 4 Network Connection — **9 hours**

Network Programming Libraries and Frameworks, Byte ordering and Byte manipulation functions, socket presentation functions, socket options, and configuration. Establish connections and send and receive data using sockets, timeout mechanisms and socket configuration parameters. Advanced Socket Programming Concepts: Concurrent server design, Introduction to raw sockets, packet generation

Unit 5 Application Layer — **8 hours**

Domain Name System, World Wide Web and Hyper Text Transfer Protocol, Electronic mail, File Transfer Protocol, Remote

login, Network Management, Data compression, VPN, Firewalls.	
Total Lecture Hours	**45 hours**

Textbook:

1. Behrouz Forouzan, "Data Communication and Networking" Fourth Edition-2006, Tata McGraw Hill
2. UNIX Network Programming: The sockets networking API, Vol 1

Reference Books:

1. Data and Computer Communication", Eighth Edition-2008, Pearson. By William Stallings,
2. Computer Networks", Fifth Edition-2011, Prentice Hall. By Andrew Tanenbaum
3. Computer Networking- A Top-Down Approach", Eighth Edition-2021, Pearson. By Kurose and Ross

Self-Study Content Links:		Duration (in Hrs)
MOOCs	1. Java Network Programming Recipes - TOC | Infosys Springboard	5 Hrs 12 Min
Video Lectures	1. https://onlinecourses.nptel.ac.in/noc21_cs18/preview 2. https://nptel.ac.in/courses/106105081 3. https://elearn.nptel.ac.in/shop/nptel/computer-networks-and-internet-protocol/	8 Hrs 5 Hrs 2 Hrs
References	https://nptel.ac.in/courses/106105183?utm_source=chatgpt.com https://www.netacad.com/courses/packet-tracer?utm_source=chatgpt.com https://www.wireshark.org/docs/?utm_source=chatgpt.com	6Hrs 2 Hrs 2 Hrs

Course Code: CCSE0402	Course Name: SOFTWARE ENGINEERING	L	T	P	C
Course Offered in: CSE (CYS)		3	0	0	3

Prerequisite: Basic knowledge about software and its types. Basic knowledge of OOPs concepts.

Course Objectives: To teach the students all phases of the Software Development Life Cycle (SDLC) and their role in software development through theory as well as practice." Students will be able to apply scientific knowledge in a systematic way to create and build cost effective software solutions.

Course Outcome: After completion of this course students will be able to		Bloom's Knowledge Level (KL)
CO1	Explain software engineering fundamentals, software process models, Agile practices, and software development life cycle concepts.	K2
CO2	Analyze software requirements using requirement engineering techniques, modelling tools, and software quality standards.	K4
CO3	Evaluate software design approaches, architectural models, and object-oriented design principles for developing effective software systems.	K5
CO4	Apply software testing strategies and testing techniques for validating software quality and functionality.	K3
CO5	Evaluate software project management, estimation methods, risk analysis, and software maintenance practices using software engineering tools.	K5

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

Course Contents / Syllabus

Unit 1 Introduction — **9 hours**

Introduction: Evolving role of software, Software Characteristics, Software crisis, Silver bullet, Software myths, Software Engineering Phases, Team Software Process (TSP), emergence of software engineering, Software process, project and product, Software Process Models: Waterfall Model, Prototype Model, Spiral Model, Iterative Model, Incremental Model, Agile Methodology: Scrum Sprint, Scrum Team, Scrum Master, Product Owner.

Unit 2 Software Requirement — **9 Hours**

Software Requirement Specifications (SRS): Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modelling, Use Case Diagram, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Assurance (SQA): Quality concepts, SQA activities, Formal approaches to SQA; Statistical software quality assurance; CMM, The ISO standard.

Unit 3 Software Design — **9 Hours**

Software Design: Design principles, the design process; Design concepts: refinement, modularity: Cohesion, Coupling, Effective modular design: Functional independence, Design Heuristics for effective modularity, Software architecture: Function Oriented Design, Object Oriented Design: OOPs concepts-Abstraction, object, classification, inheritance, encapsulation, UML Diagrams-Class Diagram, Interaction diagram, Activity Diagram, control hierarchy: Top-Down and Bottom-Up Design, structural partitioning, software procedure.

Unit 4 Software Testing — **9 Hours**

Software Testing: Testing Objectives, 7 Principals of Testing, Levels of Testing: Unit Testing, System Testing, Integration Testing, User Acceptance Testing, Regression Testing, testing for Functionality and Testing for Performance, Top Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (WhiteBox Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Functional Testing (DAO, BO) Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walkthrough, Code Inspection, Compliance with Design and Coding Standards.

Unit 5 Project Maintenance and Management Concepts — **9 Hours**

Project management concepts, Planning the software project, Estimation: Software Measurement and Metrics, Various Size Oriented Measures-LOC based, FP based, Hallstead's Software Science, Cyclomatic Complexity Measures: Control Flow Graphs, Use-case based, empirical estimation COCOMO- A Heuristic estimation techniques, staffing level estimation, team structures, risk analysis and management. Configuration Management, Software reengineering: reverse engineering, restructuring: forward engineering, Clean Room

software engineering. Case Tools, Software Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Need of Maintenance.	
Total Lecture Hours	**45 hours**

Textbook:

1. KK Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers 3RDEdition (December 11, 2008)
2. RS Pressman, Software Engineering: A Practitioners Approach, McGraw Hill. 7thEdition. (14-Jan-2022)
3. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.4th Edition. (1 January 2014)

Reference Books:

1. Pankaj Jalote, Software Engineering, Wiley. (1 January 2010)
2. Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication. 2nd Edition. (1 January 2007)
3. Kassem Saleh, "Software Engineering", Cengage Learning. (2009)
4. Ian Summerville, Software Engineering, Addison Wesley. 9th Edition. (29 October 2017)

Self-Study Content Links:		Duration (in Hrs)
MOOCs	1. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01257046726843596878_shared/overview	10 Hrs
Video Lectures	1. https://www.youtube.com/playlist?list=PL9gSnSOLPFTAoJPbLSSdeXQE5cjP44Pki	15 Hrs
Web References	1. https://mrcet.com/downloads/digital_notes/CSE/III%20Year/Software%20Engineering.pdf 2. https://www.geeksforgeeks.org/software-engineering/software-engineering/	15 Hrs

Course Code: CCSDS0411	Course Name: Fundamentals of Cloud Computing	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CYS/IOT/MCT (Department Elective)		3	0	0	3

Pre-requisite: Knowledge of basic computing units

Course Objectives:

To introduce students to the core concepts, models, and technologies of cloud computing, enabling foundational understanding of cloud-based services and infrastructure.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Explain cloud computing fundamentals, business benefits, cloud service models, deployment models, and AWS global infrastructure components such as Regions, Availability Zones, and Edge Locations.	K2
CO2	Apply AWS pricing models, billing tools, IAM concepts, security mechanisms, and shared responsibility principles for secure and cost-effective cloud usage.	K3
CO3	Configure and implement AWS networking and compute services including VPC, EC2, Route 53, Lambda, and Elastic Beanstalk for cloud-based application deployment.	K3
CO4	Analyze and compare AWS storage and database services such as S3, EBS, EFS, RDS, DynamoDB, Aurora, and Redshift based on application requirements and use cases.	K4
CO5	Design scalable, highly available, and optimized cloud architectures using AWS Well-Architected Framework, Auto Scaling, Elastic Load Balancing, and CloudWatch monitoring services.	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	PSO3
CO1	3	2	1	–	2	–	–	–	1	–	2	2	2	1
CO2	3	2	2	1	3	–	2	–	1	2	2	3	3	1
CO3	3	3	3	2	3	–	1	2	1	2	2	3	3	2
CO4	3	2	2	3	3	–	1	–	2	1	2	3	3	2
CO5	3	3	3	2	3	1	2	2	2	2	3	3	3	3

Course Contents / Syllabus

Unit - 1 | Core Principles of Cloud Computing and AWS Fundamentals | 09 hours

Business case for cloud adoption, advantages of cloud computing, overview of cloud computing concepts, AWS global infrastructure including Regions, Availability Zones, and Edge Locations. Cloud service models (IaaS, PaaS, SaaS) and deployment models (Public, Private, Hybrid). Introduction to AWS services and navigation of AWS Management Console.

Lab Work: AWS account setup and exploration of AWS Management Console. (Introductory by Faculty)

Unit - 2 | Cloud Economics and Security | 09 hours

Fundamentals of AWS pricing models, total cost of ownership (TCO), AWS billing and cost management tools, AWS Organizations and support plans. AWS Shared Responsibility Model, Identity and Access Management (IAM), role-based access control, account security, data protection, and compliance fundamentals.

Lab Work:

- Introduction to AWS IAM
- IAM user, roles, and policy configuration
- Billing dashboard and cost monitoring exploration

Unit - 3 | Networking and Compute Services | 08 hours

Fundamentals of cloud networking, Amazon Virtual Private Cloud (VPC), subnets, routing, security groups, and network ACLs. Domain Name System using Route 53 and content delivery using CloudFront. Compute services including Amazon EC2 (instance types, pricing models), introduction to containers, AWS Lambda (serverless computing), and Elastic Beanstalk for application deployment.

Lab Work:

- Build your VPC and Launch a Web Server (Module 5)
- Introduction to Amazon EC2 (Module 6)
- AWS Lambda Implementation (Module 6)
- Activity: Elastic Beanstalk Deployment (Module 6)

Unit - 4 | Storage and Database Services | 10 hours

Overview of storage types including object, block, and file storage. Amazon S3 (buckets, storage classes, lifecycle policies), Amazon EBS, EFS, and Glacier. Database services including Amazon RDS, DynamoDB, Aurora, and Redshift along with their use cases and

comparisons.
Lab Work:
- Working with EBS (Module 7)
- Build a Database Server (RDS) (Module 8)
- S3 bucket creation and storage management (console practice by Faculty)

Unit - 5	**Cloud Architecture, Monitoring, and Scaling**	**09 hours**

AWS Well-Architected Framework principles including operational excellence, security, reliability, performance efficiency, and cost optimization. Elastic Load Balancing, Auto Scaling, and high availability concepts. Monitoring and logging using Amazon CloudWatch, performance analysis, and AWS Trusted Advisor for optimization and best practices.
Lab Work:
- Scale and Load Balance your Architecture (Module 10)
- CloudWatch monitoring and performance analysis (console practice by Faculty)

Total Lecture Hours	**45 hours**

Textbooks:

Book Title	**Authors Name**
1. Fundamentals of Cloud Computing, Nitya Publication, 2020	Dr. Arun Singh Chouhan, Bipin Pandey, Vishwas Srivastava
2. Cloud Computing Basics: A Non-Technical Introduction, Apress, 2022	Anders Lisdorf

Reference Books:

Book Title	**Authors Name**
1. Cloud Computing revised and updated edition, 2023	Nayan B. Ruparelia

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	1. AWS Academy Cloud Foundations 2. AWS Cloud Practitioner Essentials	40 – 50 Hrs 6 – 8 Hrs
Video Lectures	1. https://nptel.ac.in/courses/106/104/106104182/ 2. https://www.youtube.com/watch?v=M988_fsOSWo&t=4s 3. https://www.youtube.com/watch?v=JYq1AQkMdhE 4. https://www.youtube.com/watch?v=iSG_72VNBVs&t=55s 5. https://nptel.ac.in/courses/106/105/106105167/ 6. https://youtu.be/FZR0rG3HKIk?si=i9Ol3TdIeWtC-UUJ 7. https://aws.amazon.com/ 8. https://www.youtube.com/watch?v=36zducUX16w 9. https://www.youtube.com/watch?v=3WIJ4axzFlU 10. https://www.youtube.com/watch?v=m8iz4CFVWK0 11. https://www.youtube.com/watch?v=IKxigcbhsGk 12. https://www.youtube.com/watch?v=NbkPRn1mqlU 13. https://youtube.com/playlist?list=PL1TLTEHdRxDbFyipEb0KENRuBTI9yUu26&si=Si2LGUG6 fu6v0Jr3	

Course Code: CCSIOT0411	Course Name: Sensors and Instrumentations	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(MCT) (Department Elective)		3	0	0	3

Pre-requisite: Basics of Electronics

Course Objectives:

To provide understanding of the fundamentals, classification, and operating principles of sensors and transducers used for measurement systems. To familiarize students with techniques for measurement of displacement, force, pressure, temperature, position, flow, and level using different sensors. To introduce the concepts of virtual instrumentation, graphical programming, and software-based instrumentation for industrial automation. To provide knowledge of data acquisition methods, analog and digital interfacing, ADC/DAC techniques, and networked communication. To acquaint students with smart sensors, their characteristics, selection criteria, and applications in automation and control systems.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Explain the principles, classification, characteristics, and applications of sensors and transducers used for displacement, force, pressure, and smart devices.	K2
CO2	Analyze suitable sensors for measuring physical parameters, such as temperature, position, flow, level, and vibration, in engineering applications.	K4
CO3	Examine virtual instrumentation applications using graphical programming concepts, loops, arrays, and clusters, as well as software-based industrial automation tools.	K4
CO4	Illustrate data acquisition methods, including Analog/Digital I/O, ADC, DAC, counters, timers, and networked communication techniques.	K3
CO5	Evaluate intelligent and smart sensors for industrial, robotics, and automobile control applications and select suitable sensors for practical systems.	K5

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

Course Contents / Syllabus

Unit 1 **Sensors & Transducers** **07 hours**

Sensors & Transducers: Definition, Classification of sensors, Measurement of displacement using Potentiometer, LVDT & Optical Encoder, Measurement of force using strain gauge, Measurement of pressure using LVDT based diaphragm & piezoelectric sensor, Introduction to sensors most widely used in computer and mobile phones: Temperature, Pressure, Humidity, Toxic Gas, pH sensor, sound sensor, magnetic field sensor, Motion (Infra-red) sensor, barometer, and Gyroscope.

Unit 2 **Measurement of physical parameters** **08 hours**

Measurement of temperature using Thermistors, Thermocouple & RTD, Concept of thermal imaging, Measurement of position using Hall effect sensors, Proximity sensors: Inductive & Capacitive, Use of proximity sensor as accelerometer and vibration sensor, Working Principles of Flow Sensors: Ultrasonic
& Laser, Level Sensors: Ultrasonic & Capacitive.

Unit 3 **Virtual Instrumentation** **10 hours**

Virtual Instrumentation: Graphical programming techniques, Data types, Advantage of Virtual Instrumentation techniques, Concept of WHILE & FOR loops, Arrays, Clusters & graphs, Structures: Case, Sequence & Formula nodes, Need of software based instruments for industrial automation.

Unit 4 **Data Acquisition Methods** **12 hours**

Data Acquisition Methods: Basic block diagram, Analog and Digital IO, Counters, Timers, Types of ADC: successive approximation and sigma-delta, Types of DAC: Weighted Resistor and R-2RLadder type, Use of Data Sockets for Networked Communication.

Unit 5 **Advanced Sensors** **08 hours**

Intelligent Sensors: General Structure of smart sensors & its components, Characteristic of smart sensors: Self calibration, Self-testing & self-communicating, Selection of Sensors for Practical Applications, Application of smart sensors: Automatic robot control & automobile engine control.

Total Lecture Hours **45 hours**

Textbook:

1. DVS Murthy, Transducers and Instrumentation, PHI 2nd Edition 2013
2. D Patrana bis, Sensors and Transducers, PHI 2nd Edition 2013.

Reference Books: 1. Arun K. Ghosh, Introduction to measurements and Instrumentation, PHI, 4th Edition 2012. 2. Hermann K.P. Neubert, "Instrument Transducers" 2nd Edition 2012, Oxford University Press.

Self-Study Content Links:		Duration (in Hrs)
MOOCs	**Sensors and Actuators** https://onlinecourses.nptel.ac.in/e-learning/preview/noc24_ee45	30 hours
Video Lectures	https://www.youtube.com/watch?v=9S6-fXxZruY&list=PL5vhyNmOQ5OSOSy63jf7tStFM2p78aUS4	08 hours
Web References	https://www.ni.com/en.html	08 hours

Course Code: CCSAI0411S	Course Name: DATA ANALYTICS	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI) /CSE(AIML) /CSE(DS) /CSE(CS) /CSE(CYS)/CSE(IOT)/CSE(MCT) (Department Elective)		3	0	0	3

Pre-requisite: Basic Knowledge of Statistics and Probability

Course Objectives:

The objective of this course is to understand the fundamental concepts of Data analytics and learn about various types of data formats and their manipulations. It helps students to learn exploratory data analysis and visualization techniques in addition to R/Python/Tableau programming language.

Course Outcome: After completion of this course students will be able to:		Bloom's Knowledge Level (KL)
CO1	Understand the fundamental concepts of data analytics in the areas that plays major role within the realm of data science.	K1
CO2	Explain and exemplify the most common forms of data and its representations.	K2
CO3	Understand and apply data pre-processing techniques.	K3
CO4	Analyze data using exploratory data analysis.	K4
CO5	Illustrate various visualization methods for different types of data sets and application scenarios.	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	-	-		-	2	-	3	1	2
CO2	2	3	-	1	2	-	-	1	-	1	-	2	1	1
CO3	3	3	2	2	3	-	-		-	2	-	3	2	2
CO4	2	3	2	3	3	-	-	1	-	2	1	3	2	2
CO5	2	2	2	2	3	-	-	2	-	1	1	3	2	2

Course Contents / Syllabus

Module 1	**Introduction To Data Science**	**08 hours**

Introduction to Data Science, Big Data, the 5 V's, Evolution of Data Science, Datafication, Skillsets needed, Data Science Lifecycle, types of Data Analysis, Data Science Tools and technologies, Need for Data Science, Analysis Vs Analytics Vs Reporting, Big Data Ecosystem, Future of Data Science, Applications of Data Science in various fields, Use cases of Data science-Facebook, Netflix, Amazon, Uber, AirBnB.

Module 2	**Data Handling**	**09 hours**

Types of Data: structured, semi-structured, unstructured data, Numeric, Categorical, Graphical, High Dimensional Data, Transactional Data, Spatial Data, Social Network Data, standard datasets, Data Classification, Sources of Data, Data manipulation in various formats, for example, CSV file, pdf file, XML file, HTML file, text file, JSON, image files etc. import and export data in R/Python

Module 3	**Data pre-processing**	**09 hours**

Form of Data Pre-processing, data Attribute and its types, understanding and extracting useful variables, KDD process, Data Cleaning: Missing Values, Noisy Data, Discretization and Concept hierarchy generation (Binning, Clustering, Histogram), Inconsistent Data, Data Integration and Transformation. Data Reduction: Data Cube Aggregation, Data Compression, Numerosity Reduction.

Module 4	**Exploratory Data Analysis**	**09 hours**

Handling Missing data, Removing Redundant variables, variable Selection, identifying outliers, Removing Outliers, Time series Analysis, Data transformation and dimensionality reduction techniques such as Principal Component Analysis (PCA), Factor Analysis (FA) and Linear Discriminant Analysis (LDA), Univariate and Multivariate Exploratory Data Analysis. Data Munging, Data Wrangling- APIs and other tools for scrapping data from the web/ internet using R/Python

Module 5	**Data Visualization**	**10 hours**

Introductions and overview, Debug and troubleshoot installation and configuration of the Tableau. Creating Your First visualization: Getting started with Tableau Software, Using Data file formats, connecting your Data to Tableau, creating basic charts (line, bar charts, Tree maps), Using the Show me panel.

Tableau Calculations: Overview of SUM, AVR, and Aggregate features Creating custom calculations and fields, Applying new data calculations to your visualization. Manipulating Data in Tableau: Cleaning-up the data with the Data Interpreter, structuring your data, Sorting, and filtering Tableau data, Pivoting Tableau data.

Advanced Visualization Tools: Using Filters, Using the Detail panel Using the Size panels, customizing filters, Using and Customizing tooltips, formatting your data with colours, Creating Dashboards & Stories, Distributing & Publishing Your Visualization

Total Lecture Hours	**45 hours**
Textbook:	

TEXTBOOK

BOOK TITLE	AUTHOR
Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining. 2nd Edition, John Wiley & Sons, 2014.	Myatt, G. J., & Johnson, W. P.
Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons, 2015.	Dietrich, D., Heller, B., & Yang, B

REFRENCE BOOKS:

BOOK TITLE	AUTHOR
An Introduction to Statistical Learning with Applications in Python. Springer, 2023.	James, G., Witten, D., Hastie, T., & Tibshirani, R.
R for Data Science. 2nd Edition, O'Reilly, 2023.	Wickham, H., & Grolemund, G.

SELF-STUDY CONTENT LINKS:		DURATION (IN HRS)
MOOCS	https://www.netacad.com/courses/dataanalyticsessentials?courseLang=en-US	**30Hrs**
VIDEO LECTURES	https://www.youtube.com/watch?v=r-uOLxNrNk8	**12Hrs**
WEB REFERENCES	https://jakevdp.github.io/PythonDataScienceHandbook/	**3Hrs**

Course Code: CCSE0411	Cloud Python Web Development with Django	L	T	P	C
Course Offered in: CSE (CYS)		3	0	0	3

Pre-requisite: Students should have good knowledge of Python Programming and Python coding experience.

Course Objectives:

This course focuses on how to design and build static as well as dynamic webpages and interactive web based applications.These courses mainly focus how Python operates within web development using the increasingly popular Django framework.

Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Apply the knowledge of python programing that are vital in understanding Django application and analyze the concepts, principles and methods in current client-side technology to implement Django application over the web.	K3
CO2	Demonstrate web application framework i.e. Django to design and implement typical dynamic web pages and interactive web based applications.	K3
CO3	Implementing and analyzing the concept of Integrating Accounts & Authentication on Django.	K4
CO4	Understand the impact of web designing by database connectivity with SQLite in the current market place where everyone uses to prefer electronic medium for shoping, commerce, and even social life also.	K2
CO5	Analyzing and creating a functional website in Django and deploy Django Web Application on Cloud.	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	PSO3
CO1	2	2	3	1	3	-	-	1	3	-	3	3	2	2
CO2	2	2	3	1	3	-	-	1	-	-	2	3	2	2
CO3	2	2	2	2	2	-	-	-	2	-	2	2	2	2
CO4	2	1	2	1	2	-	-	-	1	1	2	2	2	1
CO5	2	1	3	2	3	-	-	-	3	2	3	3	2	1

Course Contents / Syllabus

Unit-1	**Introduction to Nodejs**	**9 hours**

Collections-Container datatypes, Tkinter-GUI applications, Requests-HTTP requests, BeautifulSoup4-web scraping, Scrapy, Zappa, Dash, CherryPy, Turbo Gears, Flask, Web2Py, Bottle, Falcon, Cubic Web, Quixote, Pyramid.

Unit-2	**Introduction to Django Framework**	**9 hours**

Understanding Django environment, Features of Django and Django architecture, MVC and MTV, Urls and Views, Mapping the views to URLs, Django Template, Template inheritance Django Models, Creating model for site, Converting the model into a table, Fields in Models, Integrating Bootstrap into Django, Creating tables, Creating grids, Creating carousels.

Unit-3	**Integrating Accounts & Authentication on Django**	**9 hours**

Introduction to Django Authentication System, Security Problem & Solution with Django Creating Registration Form using Django, Adding Email Field in Forms, Configuring email settings, Sending emails with Django, Adding Grid Layout On Registration Page, Adding Page Restrictions, Login Functionality Test and Logout.

Unit-4	**Connecting SQLite with Django**	**9 hours**

Database Migrations, Fetch Data From Database, Displaying Data On Templates, Adding Condition On Data, Sending data from url to view, Sending data from view to template, Saving objects into database, Sorting objects, Filtering objects, Deleting objects, Difference between session and cookie, Creating sessions and cookies in Django.

Unit-5	**Deploying Django Web Application on Cloud**	**9 hours**

Creating a functional website in Django, Four Important Pillars to Deploy, registering on Heroku and GitHub, Push project from Local System to GitHub, Working with Django Heroku, Working with Static Root, Handling WSGI with gunicorn, Setting up Database & adding users.

Total Lecture Hours	**45 hours**

Textbooks:

	Authors Name
1. Python: The Complete Reference Paperback", 4th Edition 2018, McGraw Hill Education	Martin C. Brown
2. Python Programming: Using Problem Solving Approach", 3rd Edition 2017, Oxford University Press Publication.	Reema Thareja,
3. Beginning Django Web Application Development and Deployment with Python", 2 Edition 2017, Apress Publication.	Daniel Rubio, Apress
4. Python Django Web Development: The Ultimate Django web framework guide for Beginners", 2nd Edition 2019, Kindle Edition..	William Jordon,

Reference Books:

	Authors Name
1. Building Django 2.0 Web Applications: Create enterprise-grade, scalable Python web applications easily with Django 2.0", 2nd Edition 2018, and Packt Publishing.	Tom Aratyn
2. Build a website with Django", 1 Edition 2019, GNW Independent Publishing Edition.	Nigel George
3. Django in 8 Hours: For Beginners, Learn Coding Fast! 2nd Edition 2020, independently published Edition.	Ray Yao
4. Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript", 2nd Edition 2019, Kindle Edition.	Harry Percival

Self-Study Content Links:		**Duration (in Hrs)**
MOOCs	**Web Development with Python-Django**	**4 weeks**
Video Lectures	1. https://www.youtube.com/watch?v=rHux0gMZ3Eg 2. https://www.youtube.com/watch?v=D584Rm9VLLc	1Hrs 3Min 4Hrs 48Min
Web References	**Python Web Development With Django - GeeksforGeeks**	

Course Code: CSCY0412	Course Name: Cyber Threat Intelligence	L	T	P	C
Course Offered in: CSE(CYS)		3	0	0	3

Prerequisites: Basic computer literacy, fundamental networking, introduction to cybersecurity, problem-solving, logical analysis, and pattern recognition.

Course Objectives: Understand Threat Landscapes and emerging risks to master intelligence threat analysis skills and enhance defensive strategies to improve decision-making for future trends.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Define Cyber Threat Intelligence (CTI) and its role in proactive cybersecurity and apply CTI analysis. to a real-world threat scenario.	K2
CO2	Understand and apply the intrusions model to demonstrate threat sharing and frameworks.	K3
CO3	Categories and evaluate threat intelligence sources, select optimal collection methods, and process raw. data into actionable intelligence.	K4
CO4	Differentiate between IOCs (Indicators of Compromise) and IOAs (Indicators of Attack) and Analyze adversary TTPs (tactics, techniques, and procedures) using frameworks.	K5
CO5	Integrate CTI into security operations (SOC/IR/Threat Hunting) and evaluate intelligence sharing. platforms and measure threat intel ROI (Return on Investment).	K5

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

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

Course Contents / Syllabus

Unit 1	Introductions to Cyber Threat Intelligence (CTI)	8 hours

Definition and importance of CTI, Differences between tactical, operational, and strategic intelligence, The intelligence lifecycle (Planning, Collection, Processing, Analysis, Dissemination, Feedback); Key stakeholders (SOC teams, executives, law enforcement)

Unit 2	Threat Intelligence Frameworks & Models	10 hours

Definition and importance of CTI, Differences between tactical, operational, and strategic intelligence, The intelligence lifecycle (Planning, Collection, Processing, Analysis, Dissemination, Feedback), Key stakeholders (SOC teams, executives, law enforcement)

Unit 3	Collection & Processing of Threat Data	9 hours

sources of threat intelligence (Open-Source, Closed/Private, Technical, Human), Data collection methods (Logs, Threat Feeds, dark web monitoring, and Honeypots), Processing raw data into actionable intelligence, Tools for automation (MISP, ThreatConnect, Recorded Future)

Unit 4	Threat Analysis & Attribution	9 hours

Analyzing IOCs (Indicators of Compromise) and IOAs (Indicators of Attack), Behavioral analysis (TTPs of threat actors) Threat actor profiling (APT groups, cybercriminals, hacktivists), Case studies (e.g., SolarWinds, WannaCry, Emoted)

Unit 5	Operationalizing Threat Intelligence	9 hours

Integrating CTI into SOC, IR, and threat hunting, sharing intelligence (ISACs, ISAOs, and vendor platforms), Measuring effectiveness (ROI of threat intelligence), Future trends (AI in CTI, automation, deception technologies).

	Total Lecture Hours	45 hours

Textbook:

1. Cyber Threat Intelligence, Martin Lee, Wiley, 1st Edition.

Reference Books:

1. Threat Intelligence: From Data to Action, Robert M. Lee and others (Editors), Recorded Future, 1st Edition.

Self-Study Content Links:		Duration (in Hrs)
MOOCs	https://www.futureskillsprime.in/iDH/fsp/Catalog/search_text/Cyber%20threat%20intelligence/order_by/relevance/pageNo/3	16 Hours
Video Lectures	1. https://www.youtube.com/watch?v=BzXICnDoFSU&ab_channel=Archer(26 minutes 33 Seconds) 2. https://youtu.be/75GCYd5pHjg?si=0-stamvzZLtZiUzU (1 hour 21 minutes) 3. https://www.youtube.com/watch?v=p7by_uWzdEg&ab_channel=PrabhNair (30 Minutes 13 seconds)	(26 Min 33 Sec) (1h 21 Min) (30 Min 13Sec)
Web References	https://nptel.ac.in/courses/106105217?utm_source=chatgpt.com https://www.netacad.com/courses/cybersecurity/introduction-cybersecurity?utm_source=chatgpt.com https://skillsbuild.org/students/course-catalog/cybersecurity-fundamentals?utm_source=chatgpt.com	5 Hours 4 Hours 4 Hours

Course Code: CASL0401	Course Name: Technical Communication	L	T	P	C
Course Offered in: B. Tech. All Computing 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: After completion of the course, the student will be able to		**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	PSO3
CO1	1	1	1	1	1	2	1	2	3	1	1	-	-	-
CO2	1	1	1	1	1	1	1	2	3	1	1	1	-	1
CO3	1	1	1	1	1	1	1	2	3	1	1	2	1	2
CO4	1	1	1	1	1	1	1	2	3	1	1	1	1	2
CO5	1	1	1	1	1	1	1	2	3	1	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 Lecture Hours	**24 hours**

Textbook:

SNO	Books Title	Author
1	Technical Communication – Principles and Practices, 4th Edition , Oxford Univ. Press, 2022, New Delhi.	Meenakshi Raman & Sangeeta Sharma

Reference Books:

S.NO	Books Title	Author
1	Technical Communication, 15th Edition , Pearson, 2021	John M. Lannon & Laura J. Gurak

2	Spoken English- A Manual of Speech and Phonetics (5th Edition) by , 2024, New Delhi.	R K Bansal & J B Harrison, Orient Blackswan
3	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
4	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
Video Lectures	https://www.youtube.com/watch?v=kOJlwMJxEG0&t=8s https://www.youtube.com/watch?v=Sg7Q_dC_fWU&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb https://www.youtube.com/watch?v=ymLFJDpjgCk&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb&index=6	2.5 hours
Web References	Technical Communication And Soft Skills For Engineers	

LAB Course Code: CCSE0451	LAB Course Name: Database Management Systems Lab	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(MCT)		0	0	4	2

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

Course Objectives: To familiarize the students with 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)

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	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)		CO
S.No	Experiment	CO Mapping
1	Understand and implement 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, and 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	CO2

	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 the 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).	
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 and implement 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 III. 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 will 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
		Total Hours: 48 hrs.

LAB Course Code: CCSCY0451	LAB Course Name: Computer Network and Network Programming Lab	L	T	P	C
Course Offered in: CSE(CYS)		0	0	2	1

Pre-requisite: The student should have a basic knowledge of data communication and programming.

Course Objectives:

The course objectives for the practical's are to provide students with hands-on experience and skills in various aspects of cybersecurity using the mentioned tools. The practical aim is to equip students with the knowledge and skills necessary to address cybersecurity challenges and contribute effectively to the industry.

1. To be familiarized with the use of network commands.
2. To learn socket programming.
3. To implement and analyses various network protocols.

To use simulation tools to analyses the performance of various network protocols.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply networking concepts, IP addressing, subnetting, routing protocols, and network simulator tools to configure and establish computer networks.	K3
CO2	Analyse and implement various communication techniques, congestion control methods, routing algorithms, and packet transmission using socket programming.	K4
CO3	Evaluate client-server applications, packet capture mechanisms, and network protocol analysis using TCP/UDP sockets and network monitoring tools.	K5

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	3	3	3	3	2	2	2	2	2	1	3	3	2
CO2	3	3	2	3	3	1	1	1	2	2	1	3	3	2
CO3	3	3	3	3	3	2	2	1	2	2	1	3	3	3

List Of Practical's (Indicative & Not Limited To)	CO
1. Basic Networking commands.	CO1
2. Familiarization of network simulator software - Setting up a small network, Configure interfaces, IP addresses and routing Protocols.	CO1
3. Build a small network using Cisco Packet Tracer.	CO1
4. Configuration and simulation of static and dynamic routing protocols using Cisco Packet Tracer.	CO1
5. Study and implementation of DNS and DHCP server configuration in a LAN environment	CO1
6. Implementation of Checksum	CO2
7. Implementation of error detection and correction techniques using CRC and Hamming Code.	CO2
8. Implementation of congestion control algorithms.	CO2
9. Socket Programming for TCP Client-Server Communication	CO2
10. Socket Programming for UDP Client-Server Communication	CO2
11. Implementation of Unicast, Broadcast, and Multicast Communication using Socket Programming	CO2
12. Study and Analysis of TCP Three-Way Handshake using Wireshark.	CO3
13. Development of Client-Server File Transfer Application using TCP Sockets.	CO3
14. Development of Concurrent Client-Server Communication System.	CO3
15. Implementation of a simple Network Intrusion Detection System (NIDS) using packet sniffing and traffic analysis tools.	CO3
	Total Hours: 30 hrs.

Course Code: CCSCY0455	Course Name: Incident Response Management and Ethical Hacking	L	T	P	C
Course Offered in: CSE(CYS)		0	0	6	3

Pre-requisite: Computer Networks, Operating System Fundamentals, Security Fundamentals, and Web Application Concepts.

Course Objectives:

This course aims to develop knowledge and practical skills in incident response and ethical hacking by enabling learners to understand incident handling frameworks, perform evidence collection and analysis, identify and investigate security incidents, apply ethical hacking techniques to discover vulnerabilities, and design effective reporting and remediation strategies aligned with industry standards.

Course Outcome: After completion of the course, the student will be able to		Bloom's Knowledge Level (KL)
CO1	Apply incident response frameworks and lifecycle processes to effectively manage and mitigate security incidents.	K3
CO2	Perform evidence collection, log analysis, and incident investigation using appropriate tools and techniques.	K4
CO3	Analyze security threats using SIEM, threat intelligence, and attack frameworks like MITRE ATT&CK.	K4
CO4	Apply ethical hacking techniques to discover, analyze, and validate system and application vulnerabilities.	K3
CO5	Prepare professional incident and security assessment reports with appropriate remediation and mitigation strategies.	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	PSO3
CO1	3	2	1	1	1	-	1	-	1	-	2	2	2	3
CO2	3	3	2	2	2	-	2	-	2	-	2	3	3	3
CO3	3	3	3	2	3	-	3	2	2	1	3	3	3	3
CO4	2	2	2	1	2	1	3	2	2	-	3	3	3	3
CO5	2	2	2	2	1	1	3	2	2	1	3	2	2	3

Course Contents / Syllabus

Unit1	**Incident Response Frameworks & Lifecycle Management**	**12 hours**

Introduction to incident response concepts and frameworks including NIST, SANS, and ISO 27035. Phases of incident response lifecycle: preparation, identification, containment, eradication, recovery, and lessons learned. Roles and responsibilities of incident response teams (SOC/CSIRT). Incident categorization, severity levels, SLAs, and response planning. Communication, escalation procedures, and stakeholder coordination. Legal, regulatory, and ethical considerations in incident handling

Unit2	**Evidence Handling & Forensic Readiness**	**14 hours**

Roles of first responders in incident handling. Types of digital evidence including volatile and non-volatile data. Evidence collection, preservation, and chain of custody procedures. Disk imaging and memory acquisition concepts. Integrity validation using hashing. Legal admissibility and ethical considerations. Introduction to forensic tools and forensic readiness planning. Basic timeline analysis and documentation.

Unit3	**Threat Identification & Incident Analysis**	**18 hours**

Identification of Indicators of Compromise (IOCs) and suspicious activities. Log analysis and correlation across systems. Integration of threat intelligence feeds. Malware indicators and behavioural analysis basics. Incident triage, root cause analysis, and attack vector identification. Mapping incidents to MITRE ATT&CK framework. Introduction to SIEM concepts and basic threat hunting techniques.

Unit4	**Ethical Hacking, Reconnaissance & Exploitation**	**18 hours**

Foot printing and reconnaissance techniques using OSINT. Network scanning, enumeration, and vulnerability identification. System hacking concepts including password attacks and privilege escalation (controlled environment). Network attacks such as sniffing, spoofing, MITM, and DoS/DDoS (conceptual). Web application security based on OWASP Top 10 including SQL injection, XSS, authentication flaws, and misconfigurations. Use of tools such as Burp Suite/OWASP ZAP. Defensive strategies and secure configuration practices.

Unit5	**Incident Reporting, Remediation & Security Improvement**	**13 hours**

Incident documentation and reporting standards. Preparation of technical reports and executive summaries. Root cause analysis and lessons learned. Post-incident remediation and recovery strategies. Continuous monitoring and security improvement. Compliance, audit considerations, and introduction to modern security practices such as EDR/XDR and Zero Trust.

	Total Lecture Hours	**75 hours**

Textbook:

	Author

1. The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy 2nd Edition Elsevier / Syngress	Patrick Engebretson.
2. CEH Certified Ethical Hacker All-in-One Exam Guide 2nd Edition, McGraw Hill	Matt Walker.
3. Web Application Hacker's Handbook: Finding and Exploiting Security Flaws 2nd Edition, Wiley Publishing	Dafydd Stuttard and Marcus Pinto

Reference Books:

	Author
1. Penetration Testing: A Hands-On Introduction to Hacking, 1st Edition, No Starch Press	Georgia Weidman

Self-Study Content Links:		Duration (in Hrs)
MOOCs	1. Mastering Ethical Hacking - TOC | Infosys Springboard	6 Hrs 46 Min
Video Lectures	1. https://www.youtube.com/watch?v=aFpI0VIkRKg 2. https://nptel.ac.in/courses/106105217	10 Hrs. 37 min
Web References	• OWASP Foundation • Kali Linux Documentation • Nmap Official Documentation • Wireshark Documentation • Metasploit Framework • Burp Suite Documentation • OWASP Web Security Testing Guide (WSTG) • National Vulnerability Database (NVD) • MITRE ATT&CK Framework • CISA Cybersecurity Resources	14 Hrs

Note: Self-study material may include assignments, seminars, micro projects, industrial visits, any other student activities, etc

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

Lab No.	Unit	Topic	Program Logic Building	CO Mapping
1.1	Unit1	Incident Response Frameworks & Lifecycle	• Study and document incident response lifecycle based on NIST framework.	CO1
1.2	Unit1	Incident Response Frameworks & Lifecycle	• Classify different types of security incidents based on severity.	CO1
1.3	Unit1	Incident Response Frameworks & Lifecycle	• Perform incident categorization using real-world scenarios.	CO1
1.4	Unit1	Incident Response Frameworks & Lifecycle	• Design an incident response plan for a sample organization.	CO1
1.5	Unit1	Incident Response Frameworks & Lifecycle	• Simulate incident escalation and communication workflow.	CO1
1.6	Unit1	Incident Response Frameworks & Lifecycle	• Analyze roles and responsibilities in SOC/CSIRT teams.	CO1
1.7	Unit1	Incident Response Frameworks & Lifecycle	• Document incident response workflow for phishing attack scenarios.	CO1
1.8	Unit1	Incident Response Frameworks & Lifecycle	• Analyze case study of a real-world incident response failure/success.	CO1
2.1	Unit2	Evidence Handling & Forensic Readiness	• Identify and classify digital evidence (volatile/non-volatile).	CO2
2.2	Unit2	Evidence Handling & Forensic Readiness	• Perform disk imaging using forensic tools (basic).	CO2
2.3	Unit2	Evidence Handling & Forensic Readiness	• Acquire memory data from a live system (conceptual/lab).	CO2
2.4	Unit2	Evidence Handling & Forensic Readiness	• Generate and verify hashes for evidence of integrity.	CO2

2.5	Unit2	Evidence Handling & Forensic Readiness	• Document chain of custody for a forensic case.	CO2
2.6	Unit2	Evidence Handling & Forensic Readiness	• Identify forensic artifacts from a compromised system.	CO2
2.7	Unit2	Evidence Handling & Forensic Readiness	• Prepare evidence handling and acquisition report.	CO2
3.1	Unit3	Threat Identification & Incident Analysis	• Identify Indicators of Compromise (IOCs) from sample logs.	CO3
3.2	Unit3	Threat Identification & Incident Analysis	• Perform log analysis using Windows Event Viewer/Linux logs.	CO3
3.3	Unit3	Threat Identification & Incident Analysis	• Correlate logs from multiple sources for incident detection.	CO3
3.4	Unit3	Threat Identification & Incident Analysis	• Perform IOC enrichment using threat intelligence tools.	CO3
3.5	Unit3	Threat Identification & Incident Analysis	• Analyze malware indicators from sample datasets.	CO3
3.6	Unit3	Threat Identification & Incident Analysis	• Conduct incident triage and classify severity.	CO3
3.7	Unit3	Threat Identification & Incident Analysis	• Perform root cause analysis of a simulated attack.	CO3
3.8	Unit3	Threat Identification & Incident Analysis	• Map attack techniques to MITRE ATT&CK framework.	CO3
3.9	Unit3	Threat Identification & Incident Analysis	• Perform basic threat hunting using log datasets.	CO3
3.10	Unit3	Threat Identification & Incident Analysis	• Generate incident investigation report based on analysis.	CO3
4.1	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Perform OSINT-based reconnaissance on a target domain.	CO4
4.2	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Conduct network scanning using tools like Nmap.	CO4
4.3	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Perform service enumeration on target systems.	CO4
4.4	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Identify Network vulnerabilities using scanning tools.	CO4
4.5	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Identify web vulnerabilities using web scanning tools	CO4
4.6	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Conduct password attack simulation (controlled environment).	CO4
4.7	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Analyze network attacks (sniffing using Wireshark).	CO4
4.8	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Simulate MITM / DoS / DDoS attacks in controlled lab setup.	CO4
4.9	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Perform privilege escalation using PowerShell Empire	CO4
4.10	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Perform persistence and lateral movement using PowerShell Empire	CO4

4.11	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Analyze OWASP Top 10 vulnerabilities in test applications.	CO4
4.12	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Perform API security testing (basic endpoint analysis).	CO4
4.13	Unit4	Ethical Hacking, Reconnaissance & Exploitation	• Document vulnerabilities and suggest remediation.	CO4
5.1	Unit5	Reporting, Remediation & Security Improvement	• Prepare detailed incident response report (technical).	CO5
5.2	Unit5	Reporting, Remediation & Security Improvement	• Create an executive summary for business stakeholders.	CO5
5.3	Unit5	Reporting, Remediation & Security Improvement	• Perform root cause and impact analysis of incidents.	CO5
5.4	Unit5	Reporting, Remediation & Security Improvement	• Recommend remediation strategies for identifying issues.	CO5
5.5	Unit5	Reporting, Remediation & Security Improvement	• Simulate patching and validation of vulnerabilities.	CO5
5.6	Unit5	Reporting, Remediation & Security Improvement	• Map findings for MITRE ATT&CK techniques.	CO5
5.7	Unit5	Reporting, Remediation & Security Improvement	• Analyze compliance and audit requirements for incidents.	CO5
5.8	Unit5	Reporting, Remediation & Security Improvement	• Present a complete case study including detection, response, and mitigation.	CO5
				Total Hours: 75 hrs.

LAB Course Code: CCSE0459	LAB Course Name: Applied Research with Mini Project	L	T	P	C
Course Offered in: CSE/ CSE(R)/ CSE(Twin)/ CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CSE(CYS)/CSE(IOT)/CSE(MCT)		0	0	2	1

Pre-requisite: Basic knowledge of programming, data structures, databases, and fundamental concepts of software development. Familiarity with tools and technologies for system development is desirable.

Course Objectives:

This course aims to develop the ability to design and implement solutions for real-world problems using appropriate tools and technology. It emphasizes system integration, performance evaluation, and effective documentation, along with promoting research skills through the preparation of a structured research paper.

Course Outcome: After completion of the course, the student will be able to		Bloom's level
CO1	Identify and define a domain-specific real-world problem through literature review, objective formulation, and structured project planning suitable for research-based implementation.	K4
CO2	Design, develop, and validate a functional prototype/model/system using appropriate tools, technologies, or methodologies while considering performance, ethics, safety, and reliability.	K6
CO3	Document, analyze, and present project outcomes through a technical report and a structured research paper in IEEE/Scopus/UGC-approved format, demonstrating research communication and publication readiness.	K4

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

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

Module	Common Applied Research-Based Mini Project Activities (All Branches)	Expected Outcomes
Module 1: Research Foundation & Project Planning	1. Identify the domain-specific problem statement relevant to the branch. 2. Define objectives, hypothesis, and expected outcomes 3. Conduct literature review 4. Select suitable tools, technologies, software, datasets, laboratory methods, or platforms. 5. Design project methodology, workflow, or system/process architecture	Clear research problem formulation, defined objectives, foundational knowledge, and structured project roadmap
Module 2: Development, Design & Implementation	6. Develop core project components, models, experiments, prototypes, or processes. 7. Integrate relevant modules such as software, hardware, simulation, experimentation, analysis, or field implementation. 8. Apply discipline-specific techniques such as coding, fabrication, experimentation, surveys, analytics, optimization, or modelling. 9. Incorporate safety, ethics, reliability, sustainability, and security considerations where applicable	Functional prototype/model/system/process with interdisciplinary implementation and practical technical skills
Module 3: Testing, Validation & Optimization	10. Perform testing, experimentation, or validation of developed work. 11. Analyze performance, efficiency, feasibility, or effectiveness. 12. Apply analytical, computational, statistical, or intelligent techniques where relevant. 13. Optimize design, methodology, process, or implementation for improved outcomes	Validated and optimized project demonstrating performance improvement, technical analysis, and practical feasibility
Module 4: Documentation, Research Publication & Presentation	14. Document project design, methodology, implementation, and results. 15. Prepare technical report/project dissertation. 16. Draft a research paper in standard format (Abstract, Introduction, Methodology, Results, Conclusion, References) 17. Format manuscript according to IEEE/Scopus/UGC-approved guidelines. 18. Include proper figures, tables, graphs, citations, and references 19. Present project findings through seminar/poster/presentation. 20. Demonstrate developed system/model/prototype/process. 21. Submit final project report and research paper for evaluation/publication consideration	Complete technical documentation, publishable-quality research paper, professional presentation skills, and final project submission

LAB Course Code: CCSCC0452	LAB Course Name: Problem Solving Approaches Lab	L	T	P	C
Course Offered in: All Computing Branches		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

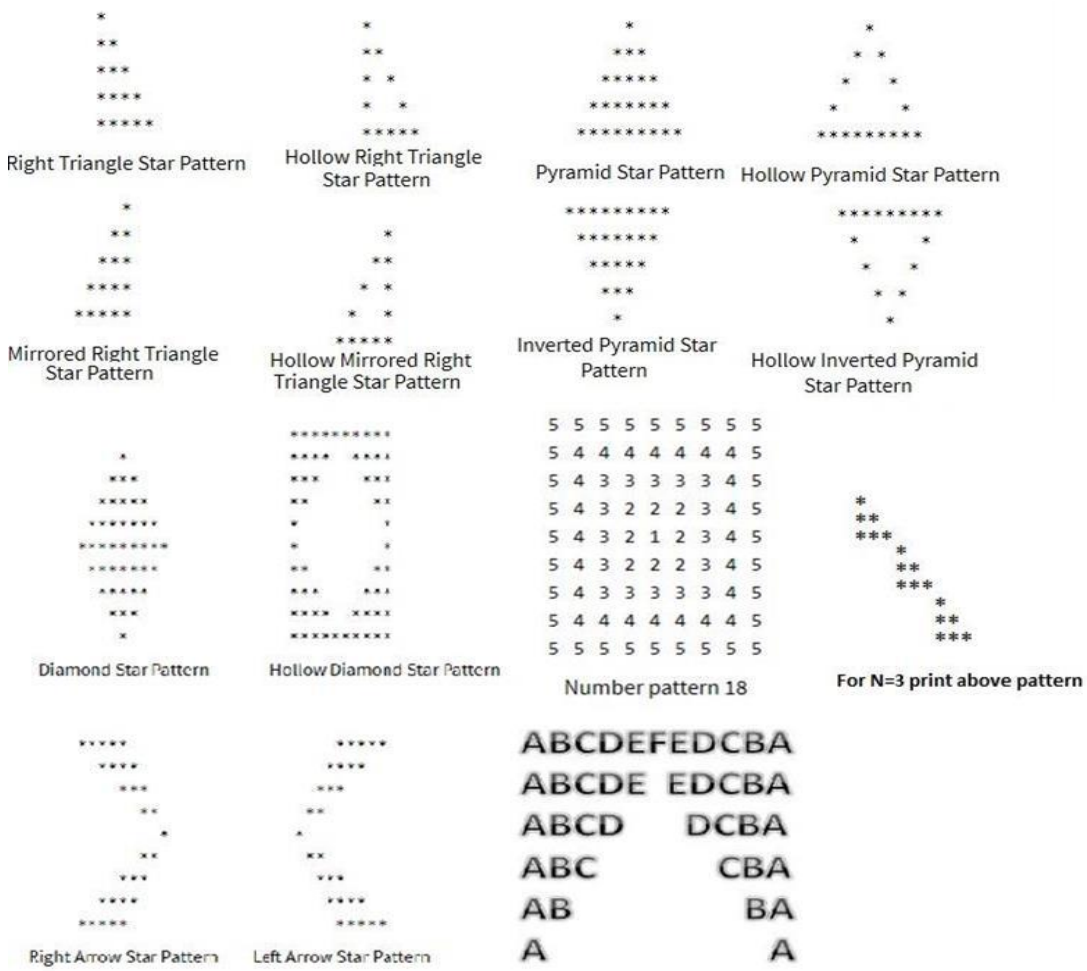
Course Outcome: After completion of the course, the student will be able to		**Bloom's Knowledge Level (KL)**
CO1	Develop logic-based solutions using control statements, recursion and bit manipulation to solve basic and intermediate computational problems.	K6
CO2	Implement and manipulate arrays and strings 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	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	1	2	2	-	-	2	3	3	2
CO2	3	3	2	2	2	-	2	-	-	-	2	2	2	2
CO3	3	3	2	2	3	1	2	2	-	-	3	3	3	2

List Of Practical's

S. No.	Name of Experiments/ Programs	CO
1. 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. 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. 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. 4	Draw the following Patterns for N=5  	
5. 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. 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. 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. 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. 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. 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. 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. 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.	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.	You 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.	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"	
Total Hours: 30 hrs.		

Course Code: CNC0401	Course Name: Artificial Intelligence and Cyber Ethics	L	T	P	C
Course Offered in: All Computing branches		2	0	0	2

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

Course Contents / Syllabus

Module 1	**An Overview to AI Ethics**	**6 hours**

DefinitionofAIEthical principles in AI, Sources ofAI data, Legal implications of AI Security Breaches, Privacy and AI Regulations, KeyPrinciplesofResponsibleAI, Transparency and Accountability,Dual-Use Dilemma, Human-Centric Design, Introduction to Cyber Laws and Ethics, Historical Development ofCyberlaws,Legal frameworks.

Module 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 GenerativeAI.

Module 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 bestpractices in engineering projects.

Module 4	**Cybercrimes, IPR and Legal Measures**	**8 hours**

Types of Cybercrimes and their Impact, Legal measures for Cybercrime Prevention andProsecution, IPR: Copyrights, Trademarks, Patents, and Trade Secrets, Ethical Implications of Intellectual Property, Cyber Securityand 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:

S.NO	Book Title	Author
1.	Artificial Intelligence: A Guide for Thinking Humans , Penguin Books, 2019.	Melanie Mitchell
2.	Cyber Ethics: Morality and Law in Cyberspace ,7th Edition (2023).	Richard Spinello, Jones & Bartlett Learning

Reference Books:

S.NO	Book Title	Author
1.	Artificial Intelligence and Ethics , BPB Publications, 2023.	S. B. Kishor, Debajit Biswas
2.	Cyber Security and Cyber Laws ,Cengage India, 2022.	Alfred Basta, Nadine Basta, Sattwik Panda,

Self-Study Content Links:		Duration (in Hrs)
MOOCs	https://swayam-plus.swayam2.ac.in/courses/course-details?id=P_INTEL_02	64 hours
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	-