SHIV NADAR
UNIVERSITY
CHENNAI

SSN School of Engineering

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

B.TECH.
COMPUTER SCIENCE AND ENGINEERING
(CYBERSECURITY)

CURRICULUM & SYLLABUS

REGULATION 2025

I. PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Graduates will emerge as leaders in the cybersecurity field, capable of safeguarding organizations and critical infrastructures from evolving cyber threats.

PEO 2: Graduates will be adept at proactively identifying, assessing, and mitigating cybersecurity risks, developing robust strategies to defend against potential attacks and vulnerabilities.

PEO 3: Graduates will be able to design, implement, and manage secure systems and networks, applying industry best practices and adhering to relevant standards and regulations to ensure the confidentiality, integrity, and availability of information assets.

PEO 4: Graduates will demonstrate proficiency in digital forensics and incident response, effectively analyzing security breaches, minimizing damage, and contributing to the recovery and resilience of affected systems.

II. PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop to the solution of complex engineering problems. (WK1 to WK4)

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

III. PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Security Principles and Practices: Graduates will be able to analyze, apply, and evaluate fundamental cybersecurity principles, practices, and technologies to protect information systems and data assets.

PSO2: Threat Analysis and Risk Management: Graduates will be able to assess and manage cybersecurity risks by identifying threats, vulnerabilities, and potential impacts, and by implementing appropriate mitigation strategies.

SEMESTER I

Sl. No.	Course Title	Course Category	Contact Periods	L	T	P	C
1	Communicative English	AEC	3	2	1	0	3
2	Mathematics I	MD	4	3	1	0	4
3	Engineering Physics	MD	3	3	0	0	3
4	Engineering Physics LAB	MD	4	0	0	4	2
5	Problem Solving and Programming in C	MC	6	2	0	4	4
6	**FOUNDATIONS OF CYBERSECURITY + LAB**	MC	5	3	0	2	4
7	Digital Design + LAB	MC	4	2	0	2	3
8	Extra Academic Activity 1[#]	EAA	2	0	0	0	1
Total			**31**	**15**	**2**	**12**	**24**

SEMESTER II

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Technical English	AEC	3	2	1	0	3
2	Mathematics II	MC	3	3	0	0	3
3	Data Structures + LAB	MC	5	2	1	2	4
4	Computer Organization and Architecture	MC	3	3	0	0	3
5	Object Oriented Programming + LAB (C++ based)	MC	4	2	0	2	3
6	Discrete Mathematics	MC	3	3	0	0	3
7	SHELL SCRIPTING	SEC	3	0	1	2	2
8	Extra Academic Activity 2[#]	EAA	2	0	0	0	1
Total			**26**	**15**	**3**	**6**	**22**

[#]Note at a later page on Extra Academic Activity.

SEMESTER III

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Design and Analysis of Algorithms + LAB	MC	5	3	0	2	4
2	**MODERN CRYPTOGRAPHY + LAB**	MC	5	2	1	2	4
3	Theory of Computation	MS	3	3	0	0	3
4	Operating Systems + LAB	MC	4	2	0	2	3
5	Linear Algebra for Computer Technologists	MS	3	2	1	0	3
6	JAVA Programming	MS	2	2	0	0	2
7	Environmental Science and Engineering	VAC	2	2	0	0	2
8	Engineering Graphics	VAC	2	0	0	2	1
9	JAVA Programming LAB	SEC	3	0	1	2	2
Total			**29**	**16**	**3**	**10**	**24**

SEMESTER IV

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Computer Networks + LAB	MS	4	2	0	2	3
2	Database Management Systems + LAB	MC	4	2	0	2	3
3	**VULNERABILITY ASSESSMENT AND PENETRATION TESTING + LAB**	MC	5	2	1	2	4
4	**MACHINE LEARNING ALGORITHMS + LAB**	MC	4	2	0	2	3
5	Full Stack Web Technologies	MS	2	2	0	0	2
6	Full Stack Web Technologies LAB	SEC	4	0	0	4	2
7	Engineering Practices Workshop	VAC	2	0	0	2	1
8	SECURE SYSTEM DEVELOPMENT (MINI PROJECT)	SEC	3	0	1	2	2
Total			**28**	**10**	**2**	**16**	**20**

SEMESTER V

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	**NETWORK & INFORMATION SECURITY + LAB**	MC	5	2	1	2	4
2	**DISTRIBUTED SYSTEMS + LAB**	MC	5	2	1	2	4
3	**DIGITAL FORENSICS + LAB**	MC	4	2	0	2	3
4	MINOR STREAM COURSE 1	MS	3	3	0	0	3
5	MINOR STREAM COURSE 2	MS	3	3	0	0	3
6	Ability Enhancement Course Basket	AEC	2	2	0	0	2
7	TOOLS FOR ETHICAL HACKING	SEC	2	0	0	2	1
Total			**24**	**14**	**2**	**8**	**20**

SEMESTER VI

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	**MALWARE ANALYSIS AND REVERSE ENGINEERING + LAB**	MC	5	2	1	2	4
2	**EMBEDDED SYSTEMS AND SECURITY + LAB**	MC	5	2	1	2	4
3	Software Engineering	MC	3	3	0	0	3
4	MINOR STREAM COURSE 3	MS	3	3	0	0	3
5	MINOR STREAM BASKET 1	MS	3	3	0	0	3
6	Capstone Project (Phase 1 – Ideation and Literature Review)	PD	8	0	0	8	4
7	Industrial Training/Internship	SI	4	0	0	4	2
Total			**31**	**13**	**2**	**16**	**23**

SEMESTER VII

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	**CLOUD COMPUTING AND SECURITY + LAB**	MC	5	2	1	2	4
2	**BLOCKCHAIN TECHNOLOGY**	MC	3	3	0	0	3
3	Compiler Design	MC	3	3	0	0	3
4	MINOR STREAM BASKET 2	MS	3	3	0	0	3
5	MINOR STREAM BASKET 3	MS	2	2	0	0	2
6	Capstone Project (Phase 2 – Prototype)	PD	12	0	0	12	6
Total			**28**	**13**	**1**	**14**	**21**

SEMESTER VIII

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Professional Ethics and Human Values	VAC	2	2	0	0	2
2	Value Added Course Basket	VAC	2	2	0	0	2
3	MINOR STREAM BASKET 4	MS	2	2	0	0	2
4	Capstone Project (Phase 3 – Report Writing)	PD	4	0	0	4	2
Total			**10**	**6**	**0**	**4**	**8**

COURSE SUMMARY

The listed courses are broadly classified as per the recommendations from the UGC.

Sl. No	Broad Category of Course	Minimum Credit Requirement
1	Major Core (MC)	80
2	Minor Stream (MS)	32
3	Multidisciplinary (MD)	09
4	Ability Enhancement Course (AEC)	08
5	Skill Enhancement Course (SEC)	09
6	Value Added Courses (VAC)	06-08
7	Summer Internship (SI)	02-04
8	Project Dissertation (PD)	12
Total		**160**

The semester wise credit breakup of the Curriculum based on the above credit breakup proposed by the UGC is as follows:

Semester\Category	MC	MS	MD	AEC	SEC	VAC	SI	PD	EAA#	TOTAL
1	11	-	9	3	-	-	-	-	1	**23 (+1)**
2	16	-	-	3	2	-	-	-	1	**21 (+1)**
3	11	8	-	-	2	3	-	-	-	**24**
4	10	5	-	-	4	1	-	-	-	**20**
5	11	6	-	2	1	-	-	-	-	**20**
6	11	6	-	-	-	-	2	4	-	**23**
7	10	5	-	-	-	-	-	6	-	**21**
8	-	2	-	-	-	4	-	2	-	**8**
Total	**80**	**32**	**9**	**8**	**9**	**8**	**2**	**12**	2	**160 (+2)**

The credits earned through the Extra Academic Activities (EAA) are not included in the calculation of Cumulative Grade Point Average (CGPA).

Note on Course Baskets

The following is a tentative list of courses that may be offered under the ***Value-Added Course Basket***:

1. Entrepreneurship Essentials
2. Green Computing
3. Water Resource Management

Note on Extra Academic Activity

At the induction programme, students will have the option to choose from three youth programmes: the National Service Scheme (NSS), the Youth Red Cross Society (YRCS), and the National Sports Organisation (NSO). The grades earned by the students will be included in the mark statement; however, these grades will not contribute to the CGPA calculation.

MINOR STREAMS

The Department of Computer Science and Engineering offers its students the following list of Minor Streams.

1. Foundations of AI and Data Science
2. Foundations of Internet of Things
3. Foundations of Cybersecurity
4. Machine Intelligence (Advanced AI and DS)
5. Cyber Physical Systems (Advanced IoT)
6. Advanced Cybersecurity

Each Minor Stream has seven courses worth 19 credits. The student must enrol in any one of the above six Minor Streams. Minor Streams tagged with Foundations must not be enrolled by students whose Major Core courses overlap with the listed courses in the Minor Stream.

MINOR STREAM 1 - Foundations of AI and Data Science

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Data Analysis through Statistical Inference*	5	4	2	0	2	3
2	Deep Learning*	5	6	2	1	0	3
3	Data Mining and Data Warehousing + LAB*	6	5	2	1	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	3	3	0	0	3
6	MINOR STREAM BASKET 3	7	3	3	0	0	2
7	MINOR STREAM BASKET 4	8	4	2	0	0	2

*Mandatory Course for the Minor Stream

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Natural Language Processing[#]	Data Preprocessing[#]	Machine Learning Operations (MLOps)	Speech Analytics
Reinforcement Learning[#]	Predictive Analytics[#]	Multi-Agent Systems	Video Analytics
GPU Programming for AI	Graph Machine Learning[#]	Edge Computing	Financial Data Analytics
Recommendation Systems	AI Ethics, Safety, and Explainability[#]		
Optimization Techniques for Machine Learning	Soft Computing		

[#]These courses are part of the Major Core courses for B.Tech. Artificial Intelligence and Data Science.

Note: Students from B.Tech. AI and DS must not opt for this Minor Stream.

MINOR STREAM 2 - Foundations of Internet of Things

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Fundamentals of Embedded Systems*	5	3	2	1	0	3
2	Fundamentals of Internet of Things*	5	3	2	1	0	3
3	IoT Architecture & Programming*	6	3	2	1	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	3	3	0	0	3
6	MINOR STREAM BASKET 3	7	2	2	0	0	2
7	MINOR STREAM BASKET 4	8	2	2	0	0	2

*Mandatory courses for the Minor Stream

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Foundations of Robotics[#]	Wireless Communication Techniques[#]	Wheeled Mobile Robots	Industry IoT 4.0
Cloud Analytics for IoT[#]	Signal Processing for IoT[#]	Smart Power Grids	Micro and Smart systems
Programming Edge Accelerators	Edge Intelligence[#]	Wireless Ad Hoc and Sensor Network	Affective Computing
Embedded Interface Design (HCI)	IoT Security Essentials[#]		
Embedded Vision	Wearable Technology		

[#]These courses are part of the Major Core courses for B.Tech. Computer Science and Engineering (Internet of Things)

Note: Students from B.Tech. CSE (IoT) must not opt for this Minor Stream.

MINOR STREAM 3 – Foundations of Cybersecurity

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Foundations of Cybersecurity[#*]	5	4	2	0	2	3
2	Digital Forensics[#*]	5	3	3	0	0	3
3	Vulnerability Assessment and Penetration Testing[#*]	6	3	3	0	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	3	3	0	0	3
6	MINOR STREAM BASKET 3	7	3	2	0	0	2
7	MINOR STREAM BASKET 4	8	2	2	0	0	2

*Mandatory courses for the Minor Stream

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Malware Analysis and Reverse Engineering[#]	Modern Cryptography[#]	Infrastructure Security	Financial Data Analytics
Embedded Systems and Security[#]	Network and Information Security [#]	Incidence Response	Open Source Intelligence
Programming Edge Accelerators	Cloud Computing and Security[#]	Wireless Ad Hoc and Sensor Network	Affective Computing
Cyber Laws and Ethics	Blockchain Technology[#]	DevSecOps	
Fault Tolerant Systems	Federated Learning		

[#]These courses are part of the Major Core courses for B.Tech. Computer Science and Engineering (Cybersecurity)

Note: Students from B.Tech. CSE (Cybersecurity) must not opt for this Minor Stream.

MINOR STREAM 4 - Machine Intelligence (Advanced AI and Data Science)

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Data Analysis through Statistical Inference*	5	4	2	0	2	3
2	Optimization Techniques for Machine Learning*	5	3	3	0	0	3
3	Deep generative models*	6	3	3	0	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	3	3	0	0	3
6	MINOR STREAM BASKET 3	7	2	2	0	0	2
7	MINOR STREAM BASKET 4	8	2	2	0	0	2

*Mandatory Course for the Minor Stream

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Tensor Computations of AI	Social Network Analysis	Machine Learning Operations (MLOps)	Speech Analytics
GPU Programming for AI	Human Computer Interactions and AI	Multi-Agent Systems	Video Analytics
Evolutionary Computing	Soft Computing	Edge Computing	Financial Data Analytics
Advanced Control Theory	Foundation of Robotics		
Affective Computing	Federated Learning		

MINOR STREAM 5 - Cyber Physical Systems (Advanced IoT)

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Modern Control Theory*	5	3	3	0	0	3
2	Embedded RTOS*	5	4	2	0	2	3
3	Programmable Logic Embedded System Design*	6	4	2	0	2	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	3	3	0	0	3
6	MINOR STREAM BASKET 3	7	2	2	0	0	2
7	MINOR STREAM BASKET 4	8	2	2	0	0	2

*Mandatory courses for the Minor Stream

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
IoT Embedded Firmware	Wearable Technology	Wheeled Mobile Robots	Industry IoT 4.0
Programming Edge Accelerators	Drone Technology	Smart Power Grids	Micro and Smart systems
Embedded Interface Design (HCI)	Automotive IoT	Wireless Ad Hoc and Sensor Network	Affective Computing
Embedded Vision	Federated Learning		

MINOR STREAM 6 - Advanced Cybersecurity

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Privacy Engineering*	5	4	2	0	2	3
2	Cyber Physical Systems*	5	3	3	0	0	3
3	Fault Tolerant Systems*	6	3	3	0	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	3	3	0	0	3
6	MINOR STREAM BASKET 3	7	3	2	0	0	2
7	MINOR STREAM BASKET 4	8	2	2	0	0	2

*Mandatory courses for the Minor Stream

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Social Media Analysis and Security	Quantum Cryptography	Infrastructure Security	Financial Data Analytics
High Performance Computing and Security	Federated Learning	Incidence Response	Open Source Intelligence
Cyber Laws and Ethics	Wireless security	SecOps	Affective Computing
Fault Tolerant Systems	Risk Management		

MICROSPECIALIZATION

The University proposes to offer Micro-Specializations to Undergraduate students. Its salient features are as follows:

1. Each Micro-Specialization has a defined structure in terms of three sequential components:
    - **Component I** - One Foundation course that constitutes a mandatory requirement and also a prerequisite for subsequent components.
    - **Component II** - Two courses from a specified vertical.
    - **Component III** - Project/Design/Term Paper
2. A Student would be required to complete all three components (10-12 credits) from the specified stream to earn a Micro-Specialization.

Eligibility for Micro-Specialization registration

To register for a Micro-Specialization, the student must have completed all curricular requirements up to the previous semester and have a CGPA $\geq$ 8.5. The student must maintain a CGPA $\geq$ 8.5 without any backlog in the subsequent semesters to keep the Micro-Specialization registration active. The Micro-Specialization must not overlap with the Major Core courses or Minor Stream courses credited as part of the degree program.

Provisional List of Micro-Specialization Verticals

The following is the list of Micro-Specialization verticals offered to our undergraduate students:

1. Artificial Intelligence and Data Science
2. Internet of Things
3. Cyber security
4. Medical Technology
5. Advanced Robotic Systems
6. Quantum Technology
7. Micro and Smart Systems Technology

SHIV NADAR UNIVERSITY CHENNAI

SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Bachelors in Technology Computer Science and Engineering (Cybersecurity)

REGULATION 2025

Syllabus

SEMESTER 1

Course code	Category	Communicative English	L	T	P	C	Contact Hours
EN1XXX	AEC		2	1	0	3	3
Course Level	100	*Prerequisite Courses: Nil*					

COURSE OBJECTIVES:

- To help learners develop the basic reading skills as required for academic purposes
- To help learners develop the writing abilities as required in academic contexts
- To help learners develop their listening skills, which will enable them to listen to lectures and comprehend them by asking questions and seeking clarifications
- To help learners develop their speaking skills and speak fluently in real contexts
- To help learners develop vocabulary, as required in academic contexts
- To help learners gain the expertise required in grammar for them to function well in academic contexts

UNIT 1 - SHARING INFORMATION RELATED TO ONESELF/FAMILY & FRIENDS (9 Hours)

Reading: Short comprehension passages, Practice in skimming-scanning and predicting, **Writing**: Completing sentences, developing hints, **Speaking**: Introducing oneself, Exchanging personal information, **Listening**: Listening comprehension of short texts, including formal and informal conversations, **Language Development**: Asking and answering - Wh- Questions and Yes / No questions, **Vocabulary Development**: Prefixes and Suffixes, Polite Expressions

UNIT 2 - GENERAL READING AND FREE WRITING (9 Hours)

Reading - Reading short narratives and descriptions from newspapers including dialogues and conversations, **Writing –** paragraph writing- topic sentence- (main ideas, organization, cohesive devices), **Listening-** telephonic conversations, short presentations and TV news , **Speaking –** sharing information of a personal kind—greeting – taking leave, **Language development –** prepositions, conjunctions, clauses, **Vocabulary development-**guessing meanings of words in context.

UNIT 3 GRAMMAR AND LANGUAGE DEVELOPMENT (9 Hours)

Reading- short texts and longer passages (close reading), **Writing-** Understanding text structure- use of reference words and discourse markers – coherence markers – Reordering jumbled sentences, **Listening –** Listening to TED talks and longer texts - product description- narratives from different sources**. Speaking-** Asking about routine actions and expressing opinions – Making short presentations, **Language development-** Degrees of comparison-

pronouns- Direct vs indirect speech, **Vocabulary development –** Idioms and phrases – single word substitutes-adverbs.

UNIT 4 READING AND LANGUAGE DEVELOPMENT (9 Hours)

Reading- Reading longer texts- reading different types of texts (journalistic, literary), **Writing-** letter writing, informal or personal letters-e-mails-conventions of personal email, **Listening-** listening to dialogues or conversations and completing exercises based on them**, Speaking-** speaking about oneself- speaking about one's friend, role-plays, **Language development-** Tenses- simple and continuous, **Vocabulary development-** synonyms-antonyms- phrasal verbs.

UNIT 5 EXTENDED WRITING (9 Hours)

Reading- Longer academic texts including comparison and contrast ones, **Writing-** Brain storming - developing an outline- identifying main and subordinate ideas- dialogue writing - writing short essays, **Listening –** Listening to talks, lectures, **Speaking –** Participating in conversations- short group discussions, **Language development-**Modal verbs- perfect and perfect continuous tenses, **Vocabulary development -** Collocations- fixed and semi-fixed expressions

TEXTBOOK:

1. Board of Editors. *Using English: A Course Book for Undergraduate Engineers and Technologists*. Orient BlackSwan Limited, Hyderabad, 2015.

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Apply language skills to share personal and social information.	KL3
CO2	Analyze texts and construct organized written and spoken responses.	KL4
CO3	Apply grammar and vocabulary to interpret and present ideas.	KL3
CO4	Analyze diverse communications to compose messages and participate in role-plays.	KL4
CO5	Evaluate academic content to write essays and lead discussions.	KL5

Course code	Category	Mathematics I	L	T	P	C	Contact Hours
MA1XXX	MD		3	1	0	4	4
Course Level	100	*Prerequisite Courses: Nil*					

Course Objectives:

This course aims to:

1. Introduce the concepts of limits, continuity, and differentiation, including an introduction to partial derivatives and gradients.
2. Cover techniques of integration and their applications.
3. Familiarize with sequences, series, and power series.
4. Introduce the basic concepts, classifications, and methods for solving first-order ordinary differential equations.
5. Cover methods for solving higher-order ordinary differential equations and introduce optimization with constraints using Lagrange multipliers.

SYLLABUS

Unit 1: Differential Calculus (8-10 hours)

Regions in the plane, Level curves and surfaces, Continuity, Partial Derivatives, Increment Theorem, Chain Rules, Directional Derivative, Normal to Level Curve and Tangent Planes, Taylor's Theorem, Extreme Values, Antiderivatives - The Definite Integral - The Fundamental Theorem of Calculus - Techniques of Integration (Substitution, Integration by Parts, Trigonometric Substitution, Partial Fractions) - Improper Integrals.

Unit 2: Integral Calculus (10-12 hours)

- Applications of Integration (Area, Volume, Arc Length), Line Integral, Line Integral of Vector Fields, Conservative Fields, Green's Theorem, Curl and Divergence of a vector field,, Surface Area of solids of Revolution, Surface area, Integrating over a surface, Surface Integral of a Vector Field, Stokes' Theorem, Gauss' Divergence Theorem

Unit 3: Functions of Several Variables (10-12 hours)

Partial differentiation – Homogeneous functions and Euler's theorem – Total derivative – Change of variables – Jacobians – Partial differentiation of implicit functions – Taylor's series for functions of two variables – Applications: Maxima and minima of functions of two variables and Lagrange's method of undetermined multipliers.

Unit 4: Sequences and Series (8-10 hours)

Sequences - Convergence and Divergence - Series - Convergence Tests (Integral Test, Comparison Tests, Ratio Test, Root Test) - Alternating Series - Absolute and Conditional Convergence - Power Series - Taylor and Maclaurin Series - Optimization with Constraints: Introduction to Lagrange Multipliers (Basic concept and simple examples)

Unit 5: First-Order Differential Equations (8-10 hours)

Introduction to Differential Equations - Classification of Differential Equations (Ordinary vs. Partial, Order, Linearity) - Solutions of Differential Equations - Initial Value Problems - First-Order ODEs: Separable Equations, Linear Equations, Exact Equations, Integrating Factors - Applications of First-Order ODEs

TEXTBOOK:

1. Stewart, J. (2015). *Calculus*. Cengage Learning.
2. Thomas, G. B., Weir, M. D., & Hass, J. (2014). *Thomas' Calculus*. Pearson.

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Apply differential and integral calculus to solve mathematical problems.	KL3
CO2	Analyze vector calculus concepts and related theorems.	KL4
CO3	Apply multivariable calculus techniques to optimize functions.	KL3
CO4	Analyze convergence of sequences and series.	KL4
CO5	Apply methods to solve first-order differential equations.	KL3

Course code	Category	Engineering Physics	L	T	P	C	Contact Hours

PH1XXX	**MD**		**3**	**0**	**0**	**3**	**3**
Course Level	**100**	*Prerequisite Courses: Nil*					

Course Objectives:

This course aims to:

1. Introduce the physics of sound waves, acoustics of buildings, and the generation and applications of ultrasonic waves.
2. Familiarize with the need for quantum mechanics, wave-particle duality, and the Schrödinger equation.
3. Cover the basic concepts of semiconductor physics, including energy bands, conductivity, and p-n junctions.
4. Introduce the principles of stimulated emission, population inversion, and the working of different types of lasers.
5. Explore the applications of ultrasonics, quantum mechanics, semiconductors, and lasers in various engineering fields.

SYLLABUS

Unit 1: Acoustics and Ultrasonics (8-10 hours)

Acoustics: Sound Waves - Characteristics of Sound - Intensity and Loudness - Reverberation - Sabine's Formula for Reverberation Time - Absorption Coefficient - Acoustic Materials - Noise Pollution and Control - Acoustics of Buildings (Basic principles) Ultrasonics: Production of Ultrasonics (Piezoelectric and Magnetostriction methods) - Properties of Ultrasonics - Detection of Ultrasonics - Applications of Ultrasonics (NDT, Medical, Industrial)

Unit 2: Quantum Physics (10-12 hours)

Introduction to Quantum Mechanics - Black Body Radiation (Planck's Law basics) - Photoelectric Effect (Einstein's explanation) - Compton Effect - Wave-Particle Duality (De Broglie Hypothesis) - Heisenberg's Uncertainty Principle - The Wave Function (ψ) - Physical Significance of ψ - Schrödinger's Time-Independent Wave Equation - Particle in a 1D Box (Energy levels and wave functions)

Unit 3: Semiconductors (10-12 hours)

Introduction to Solid State Physics (Brief overview) - Energy Bands in Solids (Conductors, Insulators, Semiconductors) - Intrinsic Semiconductors - Extrinsic Semiconductors (n-type and

p-type) - Carrier Concentration - Conductivity of Semiconductors - p-n Junction Diode (Formation and V-I characteristics basics) - Introduction to Semiconductor Devices (LED, Photodiode basics)

Unit 4: Lasers (8-10 hours)

Introduction to Lasers - Spontaneous and Stimulated Emission - Population Inversion - Pumping Mechanisms - Optical Resonators - Properties of Lasers (Monochromaticity, Coherence, Directionality, Intensity) - Types of Lasers (He-Ne Laser, Ruby Laser, Semiconductor Laser - construction and working basics)

Unit 5: Applications of Modern Physics in Engineering (8-10 hours)

Applications of Ultrasonics (Revisited with more detail) - Applications of Quantum Mechanics (Quantum Computing basics, Electron Microscopy) - Applications of Semiconductors (Transistors, Integrated Circuits - conceptual overview) - Applications of Lasers (Medical, Industrial, Communication, Holography basics) - Introduction to Fiber Optics (Principle and applications basics)

TEXTBOOKS:

1. Singh, R. P. (2008). *Engineering Physics*. Allied Publishers.
2. Malik, H. K., & Singh, A. K. (2017). *Engineering Physics*. McGraw Hill Education.
3. Pillai, S. O. (2018). *Solid State Physics*. New Age International.
4. Tayal, D. C. (2010). *Engineering Physics*. Himalaya Publishing House.
5. Serway, R. A., & Jewett Jr., J. W. (2018). *Physics for Scientists and Engineers with Modern Physics*. Cengage Learning.

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Apply principles of acoustics and ultrasonics to engineering applications.	KL3
CO2	Apply quantum mechanics concepts to problems in modern physics and technology.	KL3
CO3	Analyze properties of semiconductor materials and operation of semiconductor devices.	KL4
CO4	Apply principles of laser operation to explain laser characteristics and uses.	KL3
CO5	Analyze engineering applications of ultrasonics, quantum mechanics, semiconductors, lasers, and fiber optics.	KL4

Course code	Category	Engineering Physics LAB	L	T	P	C	Contact Hours
PH1XXX	MD		0	0	4	2	4
Course Level	100	*Prerequisite Courses: Nil*					

Tentative List of Experiments:

1. Determination of velocity and compressibility of the given liquid - Ultrasonic interferometer
2. Determination of Planck's Constant

3. Determination of specific resistance of the given wire - Carey Foster Bridge
4. Determination of Energy bandgap of the given semiconductor - Band Gap of Semi-Conductor
5. Determination of grating element / average size of the particles of a given powder sample using laser
6. Determine the numerical aperture, acceptance angle & losses in fibres of the given optical fibre cable

Course code	Category	Problems Solving and Programming in C	L	T	P	C	Contact Hours
CS1XXX	MC		2	1	2	4	5
Course Level	100	*Prerequisite Courses: Nil*					

Course Objectives

This course aims to:

1. Understand fundamental concepts of computing, including algorithms, hardware, and software.
2. Develop systematic approaches to solve computational problems using algorithmic thinking.
3. Gain proficiency in the syntax, semantics, and structure of the C programming language.
4. Design, implement, and debug C programs using structured programming principles.
5. Utilize functions and arrays effectively to modularize code and manage data.
6. Understand and apply additional C concepts such as structures, file processing, and dynamic memory management.

SYLLABUS

Unit 1 Introduction to Computers (5 hours)

Computational Problems - Algorithms – Input – Output – Efficiency – Case Study: Hand of Cards – Bubble sort – Insertion sort – Selection Sort – Linear Search – Binary Search. Hardware and Software – Data Hierarchy – Machine, Assembly, and High-level Languages – Operating Systems – The C Programming Language.

Unit 2 Introduction to C Programming (9 hours)

Similarities Between Programming and Human Languages: Alphabets – Tokens – Syntax – Statements – Blocks – Program/Headers – Library – **Keywords** – Datatypes - ASCII Character Set - Print Statement and Operators: Memory Concepts - Arithmetic - Equality and Relational Operators – Assignment Operators - Increment and Decrement Operators - Logical Operators - Pointer Variable Definitions and Initialization - Pointer Operators - sizeof Operator - Bitwise Operators – Precedence and Associativity Chart – Secure C Programming.

Unit 3 Structured Program Development (10 hours)

Pseudocodes – Flowcharts – Algorithms vs Pseudocodes vs Flowcharts - Coding Best Practices - Control Structures - The if Selection Statement - The if…else Selection Statement - The while Iteration Statement - for Iteration Statement - switch Multiple-Selection Statement - do…while Iteration Statement - break and continue Statements – Secure C Programming.

Unit 4 Functions and Arrays (11 hours)

Modularizing Programs in C - Math Library Functions - Function Definitions - Function Prototypes - Function-Call Stack and Stack Frames – Headers - Passing Arguments to Functions by Reference - Function Pointers - Storage Classes - Scope Rules – Recursion - Recursion vs. Iteration. Arrays - Defining Arrays - Pointer Expressions and Pointer Arithmetic - Relationship between Pointers and Arrays - Arrays of Pointers - Multidimensional Arrays - Sorting Arrays – Searching Arrays – Variable-Length Arrays – Secure C Programming.

Unit 5 Additional Concepts (9 hours)

Problem Solving Patterns – Two Pointers – Sliding Window – Prefix Sum – Count Frequency Arrays - Structures – Structures Definitions - File Processing – Dynamic Memory Management – Variable-Length Argument Lists - Using Command Line Arguments - Unions – typedef – Enumeration Constants – Typical C Programming - Development Environment.

The following is an indicative list of experiments that complement the theory syllabus and help students achieve the course outcomes. The course instructor may design a curated list of exercises.

1. Linux File/Folder Creation: practice creating files and directories in Linux via both GUI and CLI.
2. Linux File Manipulation: practice copying, moving, renaming, and deleting files/folders in Linux via both GUI and CLI.
3. Write "Hello, World!" and arithmetic C programs; Explore C data types and operators; C Decision Making: Implement if-else, if-else if, and switch statements for even/odd, grading system, and a basic calculator.
4. C Iteration (Loops): Use for, while, and do-while loops for sum of N numbers, factorial, and a menu-driven program.
5. 1D Arrays & Linear Search: Declare/manipulate 1D arrays; implement linear search for a given element.
6. Sorting & Binary Search: Implement Bubble Sort and Binary Search algorithms on a 1D integer array.
7. C Functions (Call by Value): Create functions for mathematical operations (e.g., power, square root using math.h) and simple value-returning functions.
8. C Functions (Pointers & Recursion): Implement functions with call by reference (e.g., swap) and a recursive function (e.g., Fibonacci series).
9. Multidimensional Arrays & More Sorting: Perform matrix addition/multiplication; implement Insertion and Selection Sort for arrays.
10. C Pointers & Array Relationships: Demonstrate pointer arithmetic for array traversal and create an array of character pointers.
11. Structures, Unions & Typedef: Define and use structures for student/employee records; illustrate memory sharing with unions; apply typedef and enumeration constants.
12. File Handling, Dynamic Memory & CLI: Implement file read/write operations; use malloc/free for dynamic arrays; process command-line arguments.

TEXTBOOKS

1. Deitel, Paul J., Harvey M. Deitel, and Abbey Deitel. (2022). C: How to Program, 9th edition. Pearson.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms, 3rd edition. MIT Press and McGraw-Hill.
3. Griffiths, David, and Dawn Griffiths. “Head First C: A Brain-Friendly Guide." O'Reilly Media, Inc., 2012.

	Course Outcomes	**Knowledge Level**
CO1	Apply fundamental algorithms, C syntax, operators, and control structures conceptually.	KL3
CO2	Analyze modular design principles using functions, pointers, recursion, and arrays.	KL4
CO3	Apply advanced concepts like problem-solving patterns, structures, file handling, and dynamic memory theoretically.	KL3
CO4	Apply Linux commands and C constructs to implement decision making, loops, arrays, pointers, and file operations in experiments.	KL3
CO5	Analyze and debug modular C programs incorporating recursion and problem-solving techniques through hands-on testing.	KL4

Course code	**Category**	**Foundations of Cybersecurity + LAB (Python)**	**L**	**T**	**P**	**C**	**Contact Hours**
CS1XXX	**MC**		**3**	**0**	**2**	**4**	**5**
Course Level	**100**	*Prerequisite Courses: Nil*					

Course Objectives:

This course aims to:

1. Understand the importance of cybersecurity and its impact on key industries.
2. Identify various cyber threats, attackers, and real-world breach cases.
3. Learn core security principles, risk management, and compliance standards.
4. Explore network fundamentals and tools for traffic analysis.
5. Gain skills in reconnaissance and ethical information gathering techniques.

SYLLABUS

Unit 1: The Cybersecurity World and Crime (9 hours)

The need for cybersecurity, Ensuring cybersecurity: Concepts, Best Practices, Impact of cyber threats on healthcare, energy and finance industries, Valuable data types and personal information, Security Threats: DoS/DDoS, Brute force attacks, MITM, Social Engineering, Phishing, Spear Phishing, Introduction to Malware types: virus, worm, trojan, ransomware, adware, Malware analysis: Static and Dynamic Analysis, Global cybersecurity regulations: GDPR, HIPAA, IT Act 2000, Case studies: Equifax, Target and Yahoo Breaches.

Unit 2: Attackers and Advanced Persistent Threats (APTs) (9 hours)

Types of attackers: Cyberterrorists, Industrial spies, Hacktivists, Insiders, cybercriminals, APT goals and stages, Famous APT groups (APT28, Lazarus Group) case studies and warning signs, Cyber kill chain stages, Types of Hackers: Black Hat, White Hat, Grey Hat, Stages of Ethical Hacking, Threat Actor Motivation Matrix, APT case study walkthrough.

Unit 3: Mitigating Risk and Taking Control (9 hours)

Concepts: Exploits Vulnerabilities, Zero-day attacks, payloads, RATs, Security practices: Strong passwords, CIA triads, PII, PCI, PHI., Risk Management processes and frameworks, Zero trust security model, Threat assessment and attack surface analysis, Common CVE's and MITRe CVE system, Security standards: NIST, ISO 27001, OWASP Top 100, Overview of SIEM systems.

Unit 4: Network Fundamentals (9 hours)

Network topologies: Star, Mesh, Ring, Bus, Networking Protocols: TCP, UDP, IP, Three-way handshake: Purpose and Mechanism, Ping and Basic networking tools, Common Ports and Protocols (HTTP, HTTPS, FTP, SSH, DNS, SMTP), Primer in subnetting (CIDR, IPV4), Networking Concepts: Firewalls, IDS, IPS, Wireshark: Introduction and packet Analysis

Unit 5: Reconnaissance and OSINT (9 hours)

Information gathering in cybersecurity, Target Identification in cyber attacks, Subdomain enumeration techniques, Identifying website technologies, Email enumeration and social engineering, OSINT: concepts and tools, Hunting for breached credentials, Active vs Passive reconnaissance.

TEXTBOOKS:

1. Charles J. Brooks, Christopher Grow, Philip Craig and Donald Shor, Cybersecurity Essential.
2. Jon Erickson, Hacking: The Art of Exploitation.
3. William Stallings, Network Security Essentials: Applications and Standards.

REFERENCES TEXTBOOKS:

1. Fernando Maymi and Shon Harris, Cissp All-In-One Exam Guide.
2. NIST cybersecurity framework documentation.
3. OWASP Top 10 official documentation.

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Apply cybersecurity principles and assess their impact on critical industries.	KL3
CO2	Analyze various cyber threats, attacker types, and real-world security breaches.	KL4
CO3	Analyze various risk management, security frameworks, and core network fundamentals for threat mitigation.	KL4
CO4	Apply Linux commands and network tools to monitor and secure network traffic.	KL3
CO5	Evaluate reconnaissance and ethical hacking techniques using practical cybersecurity tools and scripting.	KL5

Tentative List of Experiments:

The following list is an indicative list of experiments that complement the theory syllabus and help students achieve the course outcomes. The course instructor may design a curated list of exercises or a single project that covers the objectives of the following exercises.

Labs:

1. Kali Linux terminal: Basic Terminology
2. Linux Directory Structure
3. Common Networking Commands (ping, netstat, ifconfig)
4. Sudo command and package management (apt)
5. Text editing with nano
6. Bash scripting basics

Python for cybersecurity:

1. Why python is important for cybersecurity
2. Attack web forms using BeautifulSoup and requests
3. Cracking Hashed Passwords with hashlib
4. Sniff and Manipulate Traffic using Scapy
5. Regex for Log Parsing and Email detection
6. Basic Post scanner with socket and threading.

Course code	Category	DIGITAL DESIGN	SEMESTER I			
CS1XXX	MC		Credit			3
Course Level	100		L	T	P	Contact Hours
		Prerequisite Courses: Nil	2	0	2	4

Course Objectives:

1. Introducing the concept of digital and binary systems
2. Be able to design and analyze combinational logic circuits.
3. Be able to design and analyze sequential logic circuits and memory modules.
4. Understand HDL for the design and implementation of digital circuits and systems.

UNIT I-BOOLEAN ALGEBRA AND LOGIC FUNCTIONS

(6 Hours)

Number Systems – Binary Codes – Signed representation – Boolean Algebra and Theorems – Canonical forms Simplification using Karnaugh Maps and Quine–McCluskey – NAND and NOR Implementations.

Hardware Description language: *Design methodology - Introduction to HDL.*

UNIT II-COMBINATIONAL LOGIC

(6 Hours)

Analysis and Design of Combinational Circuits – Arithmetic Circuits – Code Conversion – Decoders and Encoders – Multiplexers – Comparators – Practical Combinational Circuits – Critical path analysis.

Hardware Description language: *Combinational circuits design through various modelling.*

UNIT III- SYNCHRONOUS SEQUENTIAL LOGIC

(6 Hours)

Sequential Circuits – Latches and Flip Flops – Counters – excitation tables; State Reduction and Assignment – FSM models –Shift Registers. Timing analysis – Setup /Hold Time.

Hardware Description language: *Sequential Logic Circuits Design.*

UNIT IV-ASYNCHRONOUS SEQUENTIAL LOGIC

(6 Hours)

Analysis and Design of Asynchronous Sequential Circuits – Reduction of State and Flow Tables – Race– free State Assignment.

Hardware Description language: *Datapath Design.*

UNIT V-MEMORY DEVICES AND PROGRAMMABLE LOGIC DEVICES

(6 Hours)

RAM and ROM – Memory Decoding – PROM – Flash – Programmable Logic Array – Programmable Array Logic – Sequential Programmable Devices – CPLDs, FPGA.

Hardware Description language: *Memory module implementation.*

PRACTICALS: *(Hands-on & HDL Programming)*

Syllabus (Practical component)

The following list is an indicative list of experiments that complement the theory syllabus and help students achieve the course outcomes. The course instructor may design a curated list of exercises or a single project that covers the objectives of the following exercises.

TENTAIVE LIST OF EXPERIMENTS:

1. Verification of Boolean Theorems using basic gates.
2. Design and implementation of combinational circuits using basic gates for arbitrary functions, code converters.
3. Design and implement half/full adder and subtractor.
4. Design and implement shift registers.
5. Design and implement synchronous and asynchronous counters.
6. Design and implementation of a practical real time digital system.
7. Design memory module implementation. (HDL only)

Textbooks:	
1.	M. Morris Mano and Michael D. Ciletti, "Digital Design", 5th Edition, Pearson, 2014.
2.	Charles H. Roth. "Fundamentals of Logic Design", 6th Edition, Thomson Learning, 2013
Reference Books:	
1.	Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", Pearson Education, Second Edition.
2.	Thomas L. Floyd, "Digital Fundamentals", 10th Edition, Pearson Education Inc, 2011
E-References:	
1.	Digital Circuits: https://archive.nptel.ac.in/courses/108/105/108105113/#

	Course Outcomes	**Knowledge Level**
CO1	Apply Boolean algebra concepts and simplification techniques to logic functions.	KL3
CO2	Analyze design principles of combinational and synchronous sequential circuits.	KL4
CO3	Apply theoretical concepts of asynchronous logic, memory devices, and programmable logic.	KL3
CO4	Apply combinational and sequential circuits using gates and HDL.	KL3

CO5	Evaluate practical design of real-time digital systems through hands-on experiments.	KL5

SEMESTER 2

Course code	Category	Technical Writing	L	T	P	C	Contact Hours
EN1XXX	MC		2	1	0	3	3
Course Level	100	*Prerequisite Courses: Communicative English*					

COURSE OBJECTIVES:

1. To Prepare effective Instructions, Checklists and Emails.
2. To write different kinds of paragraphs and argumentative essays.
3. To Develop and revise effective resumes, cover letters and SOPs
4. To Document and organise various kinds of reports, case studies and white papers.
6. To Create well-structured proposals and press releases.

UNIT 1 WRITING INSTRUCTIONS, CHECKLISTS AND EMAILS

Writing instructions and checklists; using bulleted lists and numbered lists. Writing professional emails: appropriate greetings, subject line, writing three-body paragraphs – introduction, body, and closing paragraph; discussion of samples. Reading sample instructional texts and professional email models.
Language Skills: Tenses – Present and past

UNIT 2 WRITING ARGUMENTATIVE PARAGRAPHS AND ESSAYS

Writing paragraphs. Structure of a paragraph: basic, compare and contrast. Product descriptions and process descriptions. Interpreting charts and graphs. Reading and writing model argumentative essays and sample analytical paragraphs; handling three types of arguments: Classical, Rogerian, Toulmin; structure of an argumentative essay: introduction, building the thesis, body paragraphs with three arguments or claims and supporting evidence, conclusion.
Sample essays.
Language skills: Connectors and definitions.

UNIT 3 WRITING JOB APPLICATIONS AND SUMMARIES

Writing job applications, resumes, and cover letters. Writing the statement of purpose (SOP): Outline – professional narrative, previous experience, research interests, and career goals. Writing summaries: reading the text; identifying main and supporting ideas. Writing paraphrases: step-by-step paraphrasing; text comparison; sentence analysis.
Language skills: subject-verb agreements.

UNIT 4 WRITING REPORTS, CASE STUDIES AND WHITE PAPERS

Writing datasheets: outline – descriptions, graphics, benefits, features, specifications, requirements, and contact information. Writing abstracts. Reading sample reports, case studies and whitepapers. Writing technical reports; practising IMRAD structure; Engineering reports &

Feasibility reports. Writing case studies and white papers: main components of an issue case study – problems, search for solutions, implementation of the solution and results.
Language skills: Active voice, passive voice

UNIT 5 WRITING PROPOSALS AND PRESS RELEASES

Writing business proposals: format -- title page, table of contents, executive summary, project details, deliverables and milestones, budget, conclusion. Writing for Websites: Inverted pyramid style; headings and subheading; bulleted lists; short paragraphs with one idea. Writing press releases: outline – headline, date of publication, contact information, summary, intro paragraph, detail paragraphs, and about section.
Language skills: Correction of errors, proofreading and editing.

TEXT BOOKS:

1. Laplante, P. A. (2018). Technical writing: A practical guide for engineers, scientists, and nontechnical professionals. CRC Press.
2. David Bonamy. (2013) Technical English Level 2 Course Book. Pearson

REFERENCES:

1. Kmiec, D., & Longo, B. (2017). The IEEE guide to writing in the engineering and technical fields. John Wiley & Sons.
2. Lewis Lansford; Peter Astley. (2013). Oxford English for Careers: Engineering 1: Student's Book. Oxford University Press.
3. Mark Ibbotson. (2008). Cambridge English for Engineering, 2008, Cambridge University Press.
4. Rubens, P. (2002). Science and technical writing: A manual of style. Routledge

COURSE OUTCOMES:

	Course Outcomes	Knowledge Level
CO1	Write instructions, checklists and emails with clarity and readability	KL3
CO2	Compose paragraphs, and structure an argumentative essay effectively	KL4
CO3	Create and edit resumes, cover letters and SOPs concisely	KL3
CO4	Document well-structured reports, case studies and white papers	KL5
CO5	Develop effective proposals and press releases	KL4

Course code	Category	Mathematics II	L	T	P	C	Contact Hours
MA1XXX	MC		3	0	0	3	3
Course Level	100	*Prerequisite Courses: Nil*					

Course Objectives:

This course aims to:

1. Introduce the core principles of probability theory, including sample spaces, events, and their relationships.
2. Familiarize with combinatorial methods for counting outcomes.
3. Cover the definitions and properties of discrete and continuous random variables and their key distributions.
4. Explain measures of central tendency, dispersion, and association for data analysis.
5. Provide a comprehensive understanding of statistical inference, including the Central Limit Theorem, confidence intervals, and various hypothesis tests.

SYLLABUS

Unit 1: Foundations of Probability and Counting (Approx. 8-10 hours)

Introduction to Probability - Sample Space - Events - Set Operations on Events - Probability Axioms - Counting: Permutations and Combinations - Conditional Probability - Independent Events - Mutually Exclusive Events - Marginal Probability - Joint Probability - Bayes' Theorem

Unit 2: Random Variables and Discrete Distributions (Approx. 10-12 hours)

Introduction to Random Variables - Discrete Random Variables - Probability Mass Functions (PMF) - Cumulative Distribution Functions (CDF) - Expected Value of Discrete Random Variables - Variance of Discrete Random Variables - Conditional Expectation and Variance - Uniform Distribution (Discrete) - Bernoulli Distribution - Binomial Distribution - Poisson Distribution

Unit 3: Continuous Distributions and Descriptive Statistics (Approx. 10-12 hours)

Continuous Random Variables - Probability Density Functions (PDF) - Cumulative Distribution Function (CDF) for Continuous Variables - Expected Value and Variance of Continuous Random Variables - Conditional PDF - Uniform Distribution (Continuous) - Exponential Distribution - Normal Distribution - Standard Normal Distribution - Introduction to t-distribution

- Introduction to Chi-squared distribution - Descriptive Statistics: Mean, Median, Mode, Standard Deviation, Correlation, Covariance

Unit 4: Sampling Distributions and Central Limit Theorem (Approx. 8-10 hours)

Populations and Samples - Sampling Methods - Introduction to Sampling Distributions - Sampling Distribution of the Mean - Central Limit Theorem - Applications of Central Limit Theorem - Sampling Distribution of the Proportion

Unit 5: Statistical Inference: Estimation and Hypothesis Testing (Approx. 8-10 hours)

Introduction to Statistical Inference - Confidence Intervals - Confidence Interval for Mean (Z-interval, t-interval) - Confidence Interval for Proportion - Introduction to Hypothesis Testing - Null and Alternative Hypotheses - Type I and Type II Errors - Significance Level - p-value - Hypothesis Tests for Mean (Z-test, t-test) - Hypothesis Tests for Proportion - Chi-squared test (for independence/goodness of fit basics)

TEXTBOOKS

1. Devore, J. L. (2015). *Probability and Statistics for Engineering and the Sciences*. Cengage Learning.
2. Ross, S. M. (2014). *A First Course in Probability*. Pearson.
3. Montgomery, D. C., & Runger, G. C. (2014). *Applied Statistics and Probability for Engineers*. Wiley.
4. Wackerly, D. D., Mendenhall III, W., & Scheaffer, R. L. (2008). *Mathematical Statistics with Applications*. Cengage Learning.

COURSE OUTCOMES:

	Course Outcomes	Knowledge Level
CO1	Apply fundamental probability concepts, set operations, and counting techniques to solve problems.	KL3
CO2	Analyze random variables and discrete probability distributions including binomial and Poisson distributions.	KL4
CO3	Apply continuous probability distributions and descriptive statistics to interpret data.	KL3
CO4	Analyze sampling distributions and the Central Limit Theorem in statistical inference.	KL4
CO5	Evaluate statistical inference techniques including estimation, confidence intervals, and hypothesis testing.	KL5

Course code	Category	DATA STRUCTURES + LAB	L	T	P	C	Contact Hours
CS1XXX	MC		2	1	2	4	5
Course Level	100	*Prerequisite Courses: Problem solving and Programming in C*					

Course Objectives:

This course aims to:

1. Introduce the basic concepts of data structures and abstract data types.
2. Familiarize with linear data structures like arrays, linked lists, stacks, and queues.
3. Cover non-linear data structures such as trees and graphs.
4. Introduce fundamental sorting and searching algorithms and their analysis.
5. Provide hands-on experience in implementing and using data structures in C++.

SYLLABUS

Unit 1: Introduction and Arrays(8-10 hours)

Introduction to Data Structures - Abstract Data Types (ADT) - Data Types vs. Data Structures - Introduction to C++ for Data Structures (Classes, Objects, Pointers, References) - Arrays - Static and Dynamic Arrays - Array Operations (Insertion, Deletion, Traversal, Searching) - Analysis of Array Operations

Unit 2: Linked Lists, Stacks, and Queues (10-12 hours)

Linked Lists - Singly Linked Lists - Doubly Linked Lists - Circular Linked Lists - Operations on Linked Lists (Insertion, Deletion, Traversal, Searching) - Stacks - Stack ADT - Array-based and Linked List-based Implementation - Applications of Stacks (Expression Evaluation, Function Call Stack) - Queues - Queue ADT - Array-based and Linked List-based Implementation - Applications of Queues (BFS, Spooling)

Unit 3: Trees (10-12 hours)

Introduction to Trees - Tree Terminology - Binary Trees - Binary Tree Traversal (In-order, Pre-order, Post-order) - Binary Search Trees (BST) - BST Operations (Insertion, Deletion, Searching) - AVL Trees (Introduction and Rotations basics) - Introduction to Heaps (Min Heap, Max Heap)

Unit 4: Graphs (10-12 hours)

Introduction to Graphs - Graph Terminology - Graph Representation (Adjacency Matrix, Adjacency List) - Graph Traversal (Breadth-First Search - BFS, Depth-First Search - DFS) - Applications of BFS and DFS - Introduction to Minimum Spanning Trees (MST) (Prim's and Kruskal's basics) - Introduction to Shortest Path Algorithms (Dijkstra's basics)

Unit 5: Searching, Sorting, and Hashing (8-10 hours)
Searching Algorithms (Linear Search, Binary Search - Analysis) - Sorting Algorithms (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort - Analysis) - Introduction to Hashing - Hash Functions - Collision Resolution Techniques (Chaining, Open Addressing)

TEXTBOOKS:

1. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). *Data Structures and Algorithms in C++*. Wiley.
2. Sahni, S. (2004). *Data Structures, Algorithms, and Applications in C++*. McGraw-Hill Education.
3. Weiss, M. A. (2014). *Data Structures and Algorithm Analysis in C++*. Pearson.

Tentative List of Experiments:

The following is an indicative list of experiments that complement the theory syllabus and help students achieve the course outcomes. The course instructor may design a curated list of exercises.

1. Implementing dynamic arrays in C++ and performing basic operations.
2. Implementing a singly linked list in C++ and performing insertion, deletion, and traversal.
3. Implementing a stack using an array or a linked list in C++ and demonstrating its applications.
4. Implementing a queue using an array or a linked list in C++ and demonstrating its applications.
5. Implementing a Binary Search Tree (BST) in C++ and performing insertion, deletion, and searching.
6. Implementing graph representation (Adjacency Matrix or List) and performing BFS or DFS traversal in C++.
7. Implementing and comparing the performance of different sorting algorithms (e.g., Bubble Sort, Merge Sort) in C++.
8. Implementing a hash table with a collision resolution technique in C++.

	Course Outcomes	**Knowledge Level**
CO1	Analyze fundamental data structures concepts including arrays, linked lists, stacks, and queues.	KL4
CO2	Analyze tree and graph structures with their traversal and basic algorithmic operations.	KL4
CO3	Analyze searching, sorting, and hashing techniques to solve computational problems efficiently.	KL3

CO4	Apply linear abstract datatypes - lists, stacks, queues and their operations in C++.	KL3
CO5	Apply non-linear abstract datatypes - tree, graph, sorting, hashing and their operations in C++.	KL3

Course code	**Category**	**COMPUTER ORGANIZATION AND ARCHTECTURE**	**L**	**T**	**P**	**C**	**Contact Hours**
CS1XXX	**MC**		**3**	**0**	**0**	**3**	**3**
Course Level	**100**	*Prerequisite Courses: Problem solving and Programming in C*					

Course Objectives:

This course aims to:

1. Introduce the basic structure, components, and performance metrics of computer systems.
2. Familiarize with instruction set architecture design principles and different addressing modes.
3. Explain the design and operation of the CPU, including datapath, control, and pipelining concepts.
4. Introduce different types of memory, their organization, and I/O mechanisms.
5. Provide a focused understanding of the characteristics and structure of the RISC-V ISA.

SYLLABUS

Unit 1: Basic Structure of Computers (8-10 hours)

Computer Types - Functional Units - Basic Operational Concepts - Bus Structures - Performance (CPU performance equation, clock rate, CPI) - Multiprocessors and Multicomputers (conceptual overview) - Machine Instructions and Programs - Memory Locations and Addresses - Memory Operations - Instructions and Instruction Sequencing - Addressing Modes - Assembly Language (basic concepts and examples) - Basic Input/Output Operations

Unit 2: Instruction Set Architecture and Data Representation (8-10 hours)

Instruction Set Architecture (ISA) - CISC vs. RISC Architectures (fundamental differences) - Instruction Formats (fixed vs. variable length) - Data Representation (Signed and Unsigned Integers, Fixed-Point and Floating-Point Representation - IEEE 754 basics) - Character Representation (ASCII, Unicode basics) - Error Detection Codes (Parity, CRC basics)

Unit 3: Basic Processing Unit (8-10 hours)

Fundamental Concepts - Execution of a Complete Instruction (Fetch, Decode, Execute) - Single-Cycle vs. Multi-Cycle Implementation (conceptual) - Multiple-Bus Organization - Control Unit Design: Hardwired Control vs. Microprogrammed Control (principles, advantages/disadvantages) - Introduction to Pipelining (Basic concepts, Pipeline Hazards - structural, data, control - conceptual solutions)

Unit 4: Memory System and I/O Organization (8-10 hours)

Memory System: Basic Concepts - Semiconductor RAM Memories (SRAM, DRAM) - Read-Only Memories (ROM, PROM, EPROM, EEPROM) - Speed, Size, and Cost - Cache Memories (Mapping Functions - direct, associative, set-associative; Replacement Algorithms - LRU, FIFO; Write Policies) - Virtual Memory (Paging and Segmentation - basic concepts, TLB) - Secondary Storage I/O Organization: Accessing I/O Devices - Programmed I/O, Interrupt-Driven I/O - Direct Memory Access (DMA) - Bus Interconnection (PCI, USB basics)

Unit 5: RISC-V Architecture 8-10 hours)

Introduction to RISC-V Architecture: Overview of RISC-V ISA - Key Features (Open, Modular, Extensible) - Base Integer Instruction Set (RV32I/RV64I) - Register Set - Basic Instruction Formats (R, I, S, B, U, J types - conceptual understanding) - Load/Store Architecture - Common Instruction Categories (Arithmetic, Logical, Control Flow) - Comparison with other ISAs (e.g., ARM, x86 - high-level differences and advantages of RISC-V) - Introduction to RISC-V Extensions (e.g., M for Multiplication/Division, A for Atomics - conceptual)

TEXTBOOKS:

1. Hamacher, C. V., Vranesic, Z. G., & Zaky, S. G. (2011). *Computer Organization and Architecture*. McGraw-Hill.
2. Stallings, W. (2019). *Computer Organization and Architecture: Designing for Performance*. Pearson.
3. Patterson, D. A., & Hennessy, J. L. (2017). *Computer Organization and Design RISC-V Edition: The Hardware/Software Interface*. Morgan Kaufmann. (Primary reference for RISC-V)
4. Hayes, J. P. (1998). *Computer Architecture and Organization*. McGraw-Hill.

	Course Outcomes	**Knowledge Level**
CO1	Apply fundamental computer organization concepts including functional units, instruction sequencing, and memory operations.	KL3
CO2	Analyze instruction set architectures, comparing CISC and RISC designs, and data representation methods.	KL4
CO3	Apply processing unit design principles including instruction execution, control unit design, and pipelining concepts.	KL3

CO4	Analyze memory systems and I/O organization including cache, virtual memory, and device interfacing techniques.	KL4
CO5	Apply basic RISC-V architecture concepts including instruction sets, register organization, and instruction formats.	KL3

Course code	**Category**	**OBJECT ORIENTED PROGRAMMING + LAB**	**L**	**T**	**P**	**C**	**Contact Hours**
CS1XXX	**MC**		**2**	**0**	**2**	**3**	**4**
Course Level	**100**	*Prerequisite Courses: Problem solving and Programming in C*					

Course Objectives:

This course aims to:

1. Introduce the core concepts of Object-Oriented Programming (OOP) paradigms.
2. Familiarize with the syntax and features of C++ for implementing classes, objects, and related concepts.
3. Cover key OOP principles like inheritance and polymorphism in C++.
4. Introduce techniques for handling runtime errors using C++ exception handling mechanisms.
5. Provide hands-on experience in developing object-oriented C++ applications through laboratory exercises.

SYLLABUS

Unit 1: Introduction to OOP and C++ Basics (5 hours)

Introduction to Programming Paradigms - Procedural vs. Object-Oriented Programming - Key Concepts of OOP (Abstraction, Encapsulation, Inheritance, Polymorphism) - Introduction to C++ - Basic Syntax and Structure - Data Types, Variables, Operators - Control Flow Statements - Functions

Unit 2: Classes, Objects, Constructors, and Destructors (7 hours)

Classes and Objects - Defining Classes - Creating Objects - Member Variables and Member Functions - Access Specifiers (public, private, protected) - Constructors (Default, Parameterized, Copy Constructors) - Destructors - `this` Pointer - Friend Functions and Friend Classes

Unit 3: Inheritance and Polymorphism (7 hours)

Inheritance - Types of Inheritance (Single, Multiple, Multilevel, Hierarchical, Hybrid) - Access Control in Inheritance - Constructors and Destructors in Inheritance - Polymorphism - Compile-time Polymorphism (Function Overloading, Operator Overloading basics) - Runtime Polymorphism (Virtual Functions, Pure Virtual Functions, Abstract Classes)

Unit 4: Operator Overloading and Exception Handling (6 hours)

Operator Overloading - Overloading Unary and Binary Operators - Overloading using Member Functions and Friend Functions - Introduction to Exception Handling - Why Exception Handling? - `try`, `catch`, and `throw` keywords - Handling Multiple Exceptions - Custom Exception Classes

Unit 5: Templates, File I/O, and STL Basics (5 hours)

Templates - Function Templates - Class Templates - File Handling in C++ - Input/Output Streams - Reading from and Writing to Files - Standard Template Library (STL) basics - Introduction to Containers (Vector, List basics) - Introduction to Iterators

TEXTBOOKS:

1. Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley Professional.
2. Lippman, S. B., Lajoie, J., & Moo, B. E. (2012). *C++ Primer*. Addison-Wesley Professional.
3. Lafore, R. (2002). *Object-Oriented Programming in C++*. Sams.
4. Deitel, P., & Deitel, H. (2017). *C++ How to Program*. Pearson.
5. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. (For design principles context)

Tentative List of Experiments:

The following is an indicative list of experiments that complement the theory syllabus and help students achieve the course outcomes. The course instructor may design a curated list of exercises.

1. Write C++ programs to demonstrate the basic syntax, data types, control structures, and functions.
2. Implement classes and objects in C++ with different access specifiers, constructors, and destructors.
3. Write C++ programs to demonstrate different types of inheritance and the use of base and derived class constructors/destructors.
4. Implement C++ programs to demonstrate compile-time polymorphism (function overloading) and runtime polymorphism (virtual functions).

5. Implement operator overloading for both unary and binary operators in C++.
6. Write C++ programs to demonstrate exception handling using `try`, `catch`, and `throw`, including handling multiple exceptions.
7. Implement a custom exception class and use it in a C++ program.
8. Write C++ programs using function templates and class templates, and perform basic file input/output operations.

	Course Outcomes	Knowledge Level
CO1	Apply OOP principles and C++ basics including classes, objects, constructors, and destructors.	KL3
CO2	Analyze inheritance types and polymorphism mechanisms in C++ for program design.	KL4
CO3	Apply operator overloading, exception handling, templates, file I/O, and STL in C++ programs.	KL3
CO4	Apply C++ programs demonstrating basic syntax, classes, inheritance, and polymorphism.	KL3
CO5	Analyze and implement advanced features like operator overloading, exceptions, and templates in experiments.	KL4

Course code	Category	DISCRETE MATHEMATICS	L	T	P	C	Contact Hours
CS1XXX	**MC**		**3**	**0**	**0**	**3**	**3**
Course Level	**100**	*Prerequisite Courses: Nil*					

Course Objectives:

This course aims to:

1. Introduce the principles of mathematical logic as a foundation for formal methods in computing.
2. Familiarize with fundamental discrete structures and their properties.
3. Cover basic algebraic concepts relevant to computer science.
4. Explain advanced counting techniques and their application in algorithm analysis.
5. Provide a comprehensive understanding of graph theory concepts and algorithms crucial for various CSE domains.

Syllabus:

Unit 1: Logic and Proofs (8-10 hours)

Propositional Logic: Propositions, Logical Connectives, Truth Tables, Tautologies, Contradictions, Logical Equivalences, Rules of Inference. First-Order Logic (Predicate Logic): Predicates, Quantifiers (Universal, Existential), Nested Quantifiers, Rules of Inference for Quantified Statements. Introduction to Proof Techniques: Direct Proofs, Proof by Contrapositive, Proof by Contradiction, Mathematical Induction.

Unit 2: Sets, Relations, Functions, Partial Orders, and Lattices (10-12 hours)

Sets: Basic Concepts, Set Operations (Union, Intersection, Complement, Difference, Cartesian Product), Power Sets. Relations: Properties of Relations (Reflexive, Symmetric, Antisymmetric, Transitive), Equivalence Relations and Partitions, Representation of Relations. Functions: Types of Functions (Injective, Surjective, Bijective), Composition of Functions, Inverse Functions. Partial Orders: Partially Ordered Sets (Posets), Hasse Diagrams, Maximal and Minimal Elements, Lattices (Definition and Properties).

Unit 3: Algebraic Structures and Basic Combinatorics (8-10 hours)

Monoids: Definition, Examples, Homomorphisms. Groups: Definition, Examples (Permutation Groups, Cyclic Groups), Subgroups, Cosets, Lagrange's Theorem (Statement and significance). Combinatorics: The Basics of Counting - The Pigeonhole Principle, Permutations and Combinations (with and without repetition).

Unit 4: Advanced Combinatorics: Recurrence Relations and Generating Functions (10-12 hours)

Recurrence Relations: Modeling with Recurrence Relations (e.g., Fibonacci numbers, Tower of Hanoi), Solving Linear Homogeneous Recurrence Relations with Constant Coefficients (Characteristic Equation method). Generating Functions: Introduction to Generating Functions, Solving Recurrence Relations using Generating Functions, Applications in Counting.

Unit 5: Graphs (10-12 hours)

Introduction to Graphs: Graph Terminology (Vertices, Edges, Degree), Types of Graphs (Directed, Undirected, Simple, Multigraphs), Graph Representation (Adjacency Matrix, Adjacency List). Connectivity: Paths, Cycles, Connected Components, Strong Connectivity (for directed graphs). Graph Traversal: Breadth-First Search (BFS) and Depth-First Search (DFS) (Conceptual overview and applications). Matching: Introduction to Matching, Bipartite Graphs, Hall's Marriage Theorem (Statement). Coloring: Graph Coloring, Chromatic Number, Applications of Graph Coloring.

References:

1. Rosen, K. H. (2018). *Discrete Mathematics and Its Applications*. McGraw-Hill Education.

2. Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage Learning. (Relevant for logic and foundational concepts)
3. Kolman, B., Busby, R., & Ross, S. (2000). *Discrete Mathematical Structures*. Pearson.
4. Levin, O. (2021). *Discrete Mathematics: An Open Introduction*. (Available online)
5. Goodaire, E. G., & Parmenter, M. M. (2006). *Discrete Mathematics with Graph Theory*.

	Course Outcomes	**Knowledge Level**
CO1	The student will be able to understand and apply fundamental concepts of propositional and first-order logic for formal reasoning.	**KL3**
CO2	The student will be able to work with discrete structures such as sets, relations, functions, partial orders, and lattices.	**KL3**
CO3	The student will be able to comprehend basic algebraic structures like monoids and groups.	**KL2**
CO4	The student will be able to apply combinatorial techniques, including recurrence relations and generating functions, for counting and problem-solving.	**KL3**
CO5	The student will be able to analyze and solve problems related to graphs, including connectivity, matching, and coloring.	**KL4**

Course code	**Category**	**SHELL SCRIPTING**	**L**	**T**	**P**	**C**	**Contact Hours**
CS1XXX	**MC**		**0**	**1**	**2**	**2**	**3**
Course Level	**100**	*Prerequisite Courses: Nil*					

Course Objectives:

This course aims to:

1. Introduce students to shell scripting and its significance in system administration and security.
2. Provide hands-on experience with automating security-related tasks using shell scripts.
3. Familiarize students with tools and utilities relevant to cybersecurity operations.
4. Enable students to write scripts for system monitoring, log analysis, and incident detection.
5. Demonstrate secure data handling techniques using shell-based encryption and hashing tools.

Syllabus:

Unit 1: Introduction to Shell Scripting and Linux Commands (1 hours)

Linux operating system overview-Shell types: sh, bash, zsh - File system navigation and permissions-Common shell commands: ls, cd, chmod, grep, awk, sed - Writing and executing basic shell scripts

Unit 2: Scripting Fundamentals and Control Structures (3 hours)

Variables, user input, and command-line arguments - Conditional statements: if, case - Loops: for, while, until - Functions, exit status, and return values - Debugging techniques (set -x, trap)

Unit 3: Automation for Cybersecurity Tasks (4 hours)

Automating user and permission management - Cron jobs and scheduled tasks - Detecting unauthorized access or privilege escalation - Port scanning using Nmap in shell scripts - Firewall configuration with iptables or ufw

Unit 4: Log Analysis and Intrusion Detection (4 hours)

System logs and directories (/var/log, auth.log, syslog) - Text processing with grep, cut, awk, sed - Scripting for detecting brute-force and login anomalies - Creating custom alert/reporting systems using logs

Unit 5: Secure File Handling (3 hours)

File Encryption-Decryption technique, Hashing techniques – md5 and sha256, Secure file transfer: scp, rsync - Scripting for integrity checks and backup automation.

References:

1. Shotts, W. E. (2019). The Linux Command Line. No Starch Press.
2. Blum, R., & Bresnahan, C. (2021). Linux Command Line and Shell Scripting Bible. Wiley.
3. Beighley, L. (2009). Head First Programming. O'Reilly Media.
4. Cooper, M. (2002). Advanced Bash-Scripting Guide. TLDP.
5. Security Tools Documentation: Nmap.org, OpenSSL.org

	Course Outcomes	Knowledge Level
CO1	Apply fundamental Linux commands and shell scripting syntax.	**KL3**
CO2	Apply scripts using conditionals, loops, and functions for cybersecurity automation.	**KL3**
CO3	Analyze shell scripts with tools like Nmap to solve security-related tasks effectively.	**KL4**
CO4	Analyze scripting techniques for log analysis and threat detection.	**KL4**
CO5	Analyze secure shell scripts for encryption, file integrity checks, and secure transfers.	**KL4**

Tentative List of Experiments:
The following list is an indicative list of experiments that complement the theory syllabus and help students achieve the course outcomes. The course instructor may design a curated list of exercises or a single project that covers the objectives of the following exercises.

Labs:

1. Introduction to Linux Shell and Command-Line Utilities
2. Writing Basic Shell Scripts
3. Automating User and File Security Management
4. Log Monitoring and Anomaly Detection
5. Port Scanning and Vulnerability Detection
6. Secure File Handling and Integrity Verification
7. Scheduling Security Scripts