GIET UNIVERSITY, GUNUPUR

DEPARTMENT

OF

COMPUTER SCIENCE & ENGINEERING

(SCHOOL OF ENGINEERING & TECHNOLOGY)

BACHELOR'S DEGREE PROGRAMME

(B. Tech)

Course Structure and Syllabus

For

Students Admitted in

[2025-29]

Academic Session

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY,ODISHA
(GIET UNIVERSITY, GUNUPUR)
SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision: To be globally acclaimed as a premier Computer Science and Engineering department through excellent teaching and a strong research environment, by creating professionally competent engineers with moral values committed to building a vibrant nation.

Mission:

M1: Provide a professional learning environment to master the fundamental principles of computing, problem-solving skills, leadership qualities, team spirit, and entrepreneurial capabilities for serving the needs of industry, government, and society at large.

M2: To enable the graduates to use modern tools and technologies by strongly encouraging a culture of research, innovation, and hands-on learning in the area of computer science and information technology.

M3: Establish Industry–Institute Interaction programs through the involvement of the IT industry in the design of the curriculum and applied learning through internship opportunities.

Program Educational Objectives:

PEO1: Develop proficiency in the fields of mathematics, basic science, and engineering fundamentals to analyze and solve a wide range of computational problems in industry or other work environments and succeed in an advanced degree program in a field such as engineering.

PEO2: Possess the ability to collaborate as team members and team leaders to facilitate technical solutions for computing systems with awareness and commitment to ethical responsibilities.

PEO3: Attain the ability to critically analyze the needs of society and provide innovative solutions in multidisciplinary areas by adopting technologies, leading to personal and professional growth as an entrepreneur.

PSO: Program Specific Outcomes

PSO1: Demonstrate practical competence and industry readiness by mastering a broad range of programming languages and open-source platforms through continuous exposure to value-added courses, training, and real-world applications.

PSO2: Apply analytical and programming skills to design, develop, and implement efficient computer-based systems of varying complexity in emerging domains such as artificial intelligence, big data analytics, and cybersecurity.

Program Outcomes

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

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)

Knowledge and Attitude Profile (WK)

WK1: A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.

WK2: Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.

WK3: A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.

WK4: Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at theforefront of the discipline.

WK5: Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.

WK6: Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.

WK7: Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.

WK8: Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.

WK9: Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Society & Basic Needs Goal

1: No Poverty – End poverty in all its forms everywhere.
2: Zero Hunger – End hunger, achieve food security, improve nutrition, and promote sustainable agriculture.
3: Good Health and Well-Being – Ensure healthy lives and promote well-being for all ages.
4: Quality Education – Ensure inclusive, equitable quality education and promote lifelong learning.
5: Gender Equality – Achieve gender equality and empower all women and girls.
6: Clean Water and Sanitation – Ensure availability and sustainable management of water and sanitation for all. Economy & Infrastructure.
7: Affordable and Clean Energy – Ensure access to affordable, reliable, sustainable, and modern energy.
8: Decent Work and Economic Growth – Promote sustained, inclusive economic growth and productive employment.
9: Industry, Innovation, and Infrastructure – Build resilient infrastructure, promote inclusive industrialization, and foster innovation.
10: Reduced Inequalities – Reduce inequality within and among countries.
11: Sustainable Cities and Communities – Make cities and human settlements inclusive, safe, and resilient.
12: Responsible Consumption and Production – Ensure sustainable consumption and production patterns. Environment & Climate.
13: Climate Action – Take urgent action to combat climate change and its impacts.
14: Life Below Water – Conserve and sustainably use the oceans, seas, and marine resources.
15: Life on Land – Protect, restore, and promote sustainable use of terrestrial ecosystems and halt biodiversity loss. Peace & Global Partnership.
16: Peace, Justice, and Strong Institutions – Promote peaceful and inclusive societies, provide access to justice, and build effective institutions.
17: Partnerships for the Goals – Strengthen the means of implementation and revitalize the global partnership for sustainable development.

COURSE CATEGORY	Course Category Code	CREDITS / SEMESTER								TOTAL CREDITS	% CREDITS
		I	II	III	IV	V	VI	VII	VIII		
Basic Science Courses	BSC	7	7	3						**17**	**10.63**
Value Education Courses: e.g. Understanding India, Environmental Science/ Education/ Digital and Tech solutions	VEC	2								**2**	**1.25**
Engineering Science Courses including workshops, drawing, basics of electronics/ electrical/ mechanical/ computer, etc.	ESC	9	8	4	4					**25**	**15.63**
Ability Enhancement Courses: English, Modern Indian Language	AEC	3	3	1	1					**8**	**5.00**
Humanities and Social Sciences including Management Courses	HSMC		3	2	2			2	2	**11**	**6.88**
Indian Knowledge Systems: Indian Architecture/ Maths/ Medicine	IKS			1						**1**	**0.63**
Mnadatory Course	MC	0								**0**	**0.00**
Co-Curricular Activities: Health and wellness/ Yoga/ Sports/ Cultural activities/ NSS/ NCC/ Applied visual performing arts	CCA		0							**0**	**0.00**
Audit Courses	AU				0					**0**	**0.00**
Professional/ Program Core Courses	PCC			11	14	15	11			**51**	**31.88**
Professional/ Program Elective courses relevant to the chosen specialization/ branch	PEC					3	6	6	3	**18**	**11.25**
Open Elective Courses - Electives from other technical and /or emerging subjects	OEC						2	2		**4**	**2.50**
Multi-disciplinary Minor: Different disciplines of engineering or different faculty altogether	MDM							3	3	**6**	**3.75**
Employment Enhancement Courses: Project/ Summer Internship/ Seminar	EEC					1	4	2	6	**13**	**8.13**
Research Methodology	RM					2				**2**	**1.25**
Vocational and Skill Enhancement Courses: Hands-on training corresponding to major/minor	VSEC				2					**2**	**1.25**
Total Credit		**21**	**21**	**22**	**23**	**21**	**23**	**15**	**14**	**160**	**100**
		42		**45**		**44**		**29**			

GENERAL COURSE STRUCTURE AND CREDIT DISTRIBUTION

Academic Regulation 2025 (AR 25)

SEMESTER I [FIRST YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	BSC	25BS1001	Applied Mathematics-I	3	1	0	4	10	30	60	100
2	BSC	25BS1002	Applied Physics	2	0	0	2	10	30	60	100
3	ESC	25ES1001	Basic Electrical and Electronics Engineering	2	1	0	3	10	30	60	100
4	ESC	25ES1003	Programming for Problem Solving	2	0	0	2	10	30	60	100
5	AEC	25AE1001	English for Technical Writing	2	0	0	2	10	30	60	100
6	VEC	25VE1001	Environmental Sciences	2	0	0	2	10	30	60	100
7	BSC	25BS1102	Applied Physics Laboratory	0	0	2	1	60	-	40	100
8	ESC	25ES1101	Basic Electrical and Electronics Engineering Laboratory	0	0	2	1	60	-	40	100
9	ESC	25ES1103	Programming for Problem Solving Laboratory	0	0	4	2	60	-	40	100
10	AEC	25AE1101	English for Technical Writing Laboratory	0	0	2	1	60	-	40	100
11	ESC	25ES1104	Engineering Graphics and Design	0	0	2	1	60	-	40	100
12	MC	25MC1301	*Induction Program (UHV-I)	0	0	0	0	-	-	-	-
			Total:	**13**	**2**	**12**	**21**	**360**	**180**	**560**	**1100**

*3 WEEKS COMPULSORY INDUCTION PROGRAM (UHV-I)

APPLIED MATHEMATICS-I

Pre-Requisite: Pre-sets and functions and properties, matrices

Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	Name of the Subject	L	T	P	C
25BS100B1	APPLIED MATHEMATICS-I	2	1	0	3
Classroom Instruction(CI) (L)		30 Hours		Total no. of Hours Per Semester	
Classroom Instruction(CI) (T)		15 Hours			
Lab Instruction (LI) (P)		-			
Term Work (TW) and Self Learning (SL) (TW+ SL)		45 Hours		90 Hours	

Course Educational Objectives

CEO1	Concept about the sets, tautology, contra positive and truth table.
CEO2	To use matrices, determinants and techniques for solving system of linear equations in Linear Algebra.
CEO3	To Provide the standard methods for differentiation.
CEO4	To Provide the standard methods for integration.

SYLLABUS	
UNIT-1 LOGIC AND SET THEORY	**(09 Hours)**
Introduction, statement, Negation, Conjunction, Disjunction, Conditional, converse, bi-conditional, Inverse, Contra positive, Truth table, Tautology, Logical Equivalence, Method of induction, sets, sub set and power set, Van Diagrams and Cartesian product of sets.	
UNIT-2 LINEAR ALGEBRA	**(09 Hours)**
Matrices, Elementary Matrices, Rank of matrix, Determinant of a matrices, Cofactors, Inverse of matrix, Solution of linear equations by Cramer's rule, symmetric and skew symmetric matrices.	
UNIT- 3 NUMERICAL METHODS	**(09 Hours)**
Algebraic and transcendental equations: Errors, Bisection method, Fixed point Iteration Method, and Newton-Raphson's method. Solution of system of Linear equations by Gauss elimination, Gauss Jordan. Inverse of a matrix by Gauss Jordan method.	
UNIT- 4 INTERPOLATION	**(09 Hours)**
Errors, Finite Differences, Newton's forward and backward interpolation formula (without proof), Newton's divided difference interpolation, Lagrange's interpolation.	

UNIT-5 PROBABILITY	**(09 Hours)**
Events, Sample space, definition of probability, Axioms of probability, Addition theorem of probability with examples, Multiplication theorem of probability with illustrate examples. Conditional probability, Baye's theorem, simple problems.	

Reference Books:

1. Advanced Engineering Mathematics by E. Kreyszig, Tenth Edition, Willey
2. Differential Calculus by Santi Narayan and Mittal, S.Chand Publications
3. Numerical Method by S. S. Sastry
4. Higher Engineering Mathematics by BS Grewal : Khanna Publishers, New Delhi.
5. Higher Engineering Mathematics by B.V.Ramana, McGraw Hills Education
6. Advanced Engineer methods by N. P. Bali & Manish Goyal.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Solve the problems involving sets, subsets, power sets, Venn diagrams, and Cartesian products.
CO2	Explain problems involving matrix operations, including calculating the rank, determinant, inverse, and transpose of matrices.
CO3	Apply numerical methods to solve algebraic and transcendental equations, and estimate values using interpolation techniques.
CO4	Execute the integration of basic functions using techniques such as substitution, integration by parts, and partial fractions.
CO5	Gain the skills to use numerical methods, including Newton's forward and backward interpolation formulas, divided difference interpolation, and Lagrange's interpolation.
CO6	Calculate measures of central tendency, perform correlation and regression analysis, and conduct hypothesis testing using small sample tests.

CO-PO Mapping

	PROGRAMME OUTCOMES										
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	2	3	-	-	-	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-
CO5	2	3	-	-	-	-	-	-	-	-	-
CO6	2	3	-	-	-	-	-	-	-	-	-
AVG.	2.33	3	-	-	-	-	-	-	-	-	-

APPLIED PHYSICS

Pre-Requisite: Mathematics and Physics

Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	Name of the Subject	L	T	P	C
25BS1002	APPLIED PHYSICS	2	0	0	2

Classroom Instruction(CI) (L)	34 Hours	Total no. of Hours Per Semester
Classroom Instruction(CI) (T)	-	
Lab Instruction (LI) (P)	-	
Term Work (TW) and Self Learning (SL) (TW+ SL)	26 Hours	60 Hours

SYLLABUS

UNIT-1 OSCILLATION, WAVES AND OPTICS	**(07 Hours)**
Introduction to Oscillatory motion, Damped harmonic oscillation and its characteristics, Waves and its characteristics, Superposition of Waves, coherent and incoherent superposition, Interference of light waves, Newton's Ring experiment, diameter of rings, determination of wavelength of unknown light.	
UNIT-2 ELECTROMAGNETIC THEORY	**(07 Hours)**
Vector Calculus, Review of grad, divergence and curl, Gauss divergence theorem and Stokes theorem (no derivations), fundamental laws of electrostatics, magneto-statics and electromagnetism, displacement current and conduction current, Maxwell's equations, Electromagnetic wave and its characteristics, electromagnetic wave equation for free space, Poynting vector, Poynting theorem(Statement Only).	
UNIT- 3 CRYSTAL STRUCTURE AND OPTOELECTRONICS	**(06 Hours)**
Crystal Structure: crystal plane, Miller indices, Inter planar spacing, Reciprocal Lattice and its characteristics, Reciprocal Lattice of SC, BCC and FCC, X-ray diffraction (Basic) and Bragg's law. LASER, Lasing action, characteristics and application of laser beam, Optical fiber, Acceptance angle, Numerical aperture, Step index and Graded index fibers, Attenuations in optical fibers, Block diagram of optical fiber communication Systems and applications.	
UNIT- 4 PHYSICS OF ENGINEERING MATERIALS	**(07 Hours)**
Super Conducting Materials: In,troduction, type-1 and type-2, Meissner effect, critical magnetic field and current density, Applications. Magnetic Materials: Properties of Magnetic Materials, types and applications. Di-electric Materials: Polar and non-polar dielectric, types of dielectric, piezoelectric, pyro-electric and ferroelectric, Applications.	
UNIT-5 QUANTUM MECHANICS AND APPLICATION	**(07 Hours)**

Introduction to dual nature: Black body radiation, photoelectric effect, Compton effect (qualitative ideas only), de-Broglie's hypothesis, Heisenberg's uncertainty principle and its application to non-existence of electron inside the nucleus, Basic postulates of Quantum Mechanics, Wave function and its characteristics, probability density, normalization, eigen values, eigen functions and expectation values, Schrödinger equation and its application a to free particle.
Reference Books: 1. Materials Science &Engg., V. Raghvan, Prentice Hall of India. 2. Concepts of Modern Physics, A. Beiser, S. Mahajan and S. R. Choudhary, Tata Mc Graw Hill.

Course Educational Objectives	
CEO1	Impart among the students an attitude of being inquisitive so that they are capable of independent and critical thinking.
CEO2	Enable the students to analyse problems starting from first principles, evaluate and validate experimental results, and draw logical conclusions.
CEO3	Prepare the students to pursue research careers, careers in academics, industries in Physical Science and allied fields.
CEO4	To attain fundamental information about Quantum mechanics with applications.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Understand the principles of oscillations and waves and analyze wave motion, superposition, and interference phenomena.
CO2	Comprehend the fundamental concepts of electromagnetism, Maxwell's equations, and properties of electromagnetic wave.
CO3	Investigate the crystal structures, crystal planes and X-ray diffraction techniques for understanding solid-state physics.
CO4	Explain the working principles of lasers and optical fibres, and their applications in communication systems.
CO5	Describe the properties and applications of superconducting, magnetic, dielectric materials in engineering field.
CO6	Apply the concepts of quantum mechanics to solve quantum problems using Schrodinger's equation.

CO-PO Mapping

PROGRAMME OUTCOMES											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	3	2	-	-	-	-	-	-	-	-	-
CO2	3	2	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	1	-	-	-	-	-	-	-	-	-
CO5	3	2	-	-	-	-	-	-	-	-	-
CO6	3	3	-	-	-	-	-	-	-	-	-
AVG.	3	2.2	-	-	-	-	-	-	-	-	-

BASIC ELECTRICAL AND ELECTRONICS ENGINEERING					
Pre-Requisite: Students should have a good understanding of Mathematics and Physics. **Teaching Methods: Chalk & Board/PPT/Video Lectures**					
Subject Code	**Name of the Subject**	**L**	**T**	**P**	**C**
25ES1001	**BASIC ELECTRICAL AND ELECTRONICS ENGINEERING**	**2**	**1**	**0**	**3**
Classroom Instruction(CI) (L)		**35 Hours**		**Total no. of Hours Per Semester**	
Classroom Instruction(CI) (T)		**10 Hours**			
Lab Instruction (LI) (P)		**-**			
Term Work (TW) and Self Learning (SL) (TW+ SL)		**45 Hours**		**90 Hours**	
SYLLABUS					
UNIT-1 FUNDAMENTALS OF DC CIRCUITS AND NETWORK THEOREMS				**(08 Hours)**	
Fundamentals of DC Circuits: Ohm's Law; Kirchhoff's Laws; Voltage and Current Sources; Series and Parallel Circuits. **Network Theorems:** Thevenin's Theorem; Norton's Theorem.					
UNIT-2 AC FUNDAMENTALS AND THREE-PHASE SYSTEMS				**(10 Hours)**	
AC Fundamentals: Sinusoidal Waveforms; RMS and Average Values; A.C. through resistance, inductance, and capacitance, simple RL, RC, and RLC circuits; Concept of Power and Power Factor. **Three-Phase Systems:** Introduction to 3-phase voltages and currents; Star (Y) and Delta (Δ) Connection.					
UNIT- 3 TRANSFORMERS AND DC MACHINES				**(09 Hours)**	
Single Phase Transformers: Introduction and principle of operation; EMF Equation; Transformer Construction; Practical and Ideal Transformer. **DC Machines:** DC Generator; DC Motor; Types of DC Motors; Applications of DC Machines.					
UNIT- 4 PN-JUNCTION DIODE AND BIPOLAR JUNCTION TRANSISTOR				**(10 Hours)**	
PN-Junction Diode: Intrinsic and Extrinsic Semiconductors; PN-Junction Diode Characteristics, Forward and Reverse Bias; Diode as Rectifier, Half-Wave and Full-Wave Rectifier. **Bipolar Junction Transistor (BJT):** Construction and Types (NPN/PNP); Input / Output Characteristics; Common Base, Common Emitter, and Common Collector Configurations; Transistor as a Switch.					
UNIT-5 OPERATIONAL AMPLIFIERS AND DIGITAL LOGIC				**(08 Hours)**	
Operational Amplifier: Symbol and Pin Diagram; Differential and Common-Mode Gain; CMRR; Inverting and Non-Inverting Amplifiers. **Introduction to Digital Logic:** Binary, Octal, Hexadecimal, and Decimal Number Systems and their Conversion; Logic Gates.					

Text Books:
01. D.P. Kothari & I.J. Nagrath – Basic Electrical and Electronics Engineering, McGraw Hill
02. M.S. Sukhija & T.K. Nagsarkar – Basic Electrical and Electronics Engineering, Oxford University Press
03. R.S. Sedha – Applied Electronics, S. Chand
04. B.L. Theraja, "A Textbook of Electrical Technology", Vol. I & II, S. Chand Publications.

Reference Books:
01. S.K. Bhattacharya "Basic Electrical and Electronics Engineering", Pearson Education, Fourth Edition, 2021.
02. Digital Design by M. Morris Mano, 5th Edition, Pearson Education.
03. S. Parker Smith: "Problems in Electrical Engineering" Asia Publications
04. Schaum's Outline of Basic Electrical Engineering.
05. Handbook of Modern Sensors: Physics, Designs, and Applications Paperback – 19 September 2014 by Jacob Fraden (Springer).

Course Educational Objectives

CEO1	To build a strong foundation in DC and AC circuit analysis, network theorems, and three-phase systems.
CEO2	To introduce the working principles and applications of transformers, DC machines, and semiconductor devices like diodes and transistors.
CEO3	To familiarize students with operational amplifiers and basic digital logic concepts essential for electronics.
CEO4	To develop analytical and problem-solving skills for practical engineering applications and future advanced studies.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Apply fundamental laws and network theorems to analyze DC electrical circuits.
CO2	Analyze AC fundamentals, three-phase systems, and explain the working principle and operation of single-phase transformers.
CO3	Describe the construction, operation, and applications of DC machines, generators, and motors.
CO4	Explain the characteristics and operation of PN-junction diodes and Bipolar Junction Transistors (BJTs) and their use in rectifier, amplification, and switching circuits.
CO5	Interpret operational amplifier parameters and design basic inverting and non-inverting amplifier circuits.
CO6	Understand number systems, implement basic logic gates, and design simple digital logic circuits.

CO-PO Mapping

PROGRAMME OUTCOMES

COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	3	3	2	-	-	-	-	-	-	-	-
CO3	3	2	-	-	-	-	-	-	-	-	-
CO4	3	2	-	-	-	-	-	-	-	-	-
CO5	3	2	2	-	-	-	-	-	-	-	-
CO6	3	2	2	-	-	-	-	-	-	-	-
AVG.	3	2.33	2	-	-	-	-	-	-	-	-

PROGRAMMING FOR PROBLEM SOLVING

Pre-Requisite: Computer Basics
Teaching Methods: Chalk & Board, PPT

Subject Code	Name of the Subject	L	T	P	C
25ES1003	PROGRAMMING FOR PROBLEM SOLVING	2	0	0	2

Classroom Instruction(CI) (L)	**40 Hours**	**Total no. of Hours Per Semester**
Classroom Instruction(CI) (T)	-	
Lab Instruction (LI) (P)	-	
Term Work (TW) and Self Learning (SL) (TW+ SL)	**20 Hours**	**60 Hours**

SYLLABUS

UNIT-1 **(08 Hours)**

Programming Basics: Basic structure of C program, C compilers, Compilation and Execution Process. Character set, C Tokens, data types, modifiers, format specifiers, escape sequence characters, constants, variables, standard I/O statements, Operators types, Operator precedence and associativity, Implicit and Explicit type casting, programs using sequential statements.

UNIT-2 **(08 Hours)**

Selection Control Statements: if, if..else, else if ladder, nested if, switch..case. Loop Control Statements: while, do-while and for loop, break, continue, nested loop.

UNIT- 3 **(08 Hours)**

1-D Array: declaration, initialization, array operations. 2-D Array: declaration, Initialization, 2-D array operations.
Character arrays and Strings: String operations, Library Functions: strcmp(), strcat(), strcpy(), strlen(), toupper(), tolower(), isupper(), islower().

UNIT- 4 **(08 Hours)**

User Defined Functions: Function categories, Parameter passing in functions, Passing arrays to functions, Recursive functions, storage classes.
Structure: Declaration and initialization of structures, typedef, accessing structure elements, nested structures, structures with array, structures with functions.

UNIT-5 **(08 Hours)**

Pointers: Declaration and initialization of pointers, Pointer arithmetic, Pointer to Array, Pointer to Function, call by value and call by address, pointer to structure, dynamic memory allocation(DMA), DMA with array, functions and structures.

Text Books:

01. E. Balaguruswamy, Programming in ANSI C, 7th edition, Tata McGraw-Hill
02. Let us 'C' by Yashwant Kanethekar, 16th edition, BPB Publications
03. Byron Gottfried, Schaum's Outline of Programming with C, 3rd edition, McGraw-Hill

Reference Books:

01. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language,2nd edition, Prentice Hall of India
02. Programming in C, by Reema Thareja, 2nd edition, OUP India
03. C Programming and Coding by swati saxena, BPB Publications

Course Educational Objectives

CEO1	To formulate algorithm, translate into program and then execute the programs for verifying its correctness.
CEO2	To analyse a problem for knowing its efficiency and decompose it into functions using divide and conquer approach.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Explain the fundamental features of structure-oriented programming and describe the basic elements of programming, such as data types, operators, control statements, and arrays.
CO2	Classify various types of statements and demonstrate programs using control structures, arrays, functions, and structures.
CO3	Solve problems using different programming logics by applying arrays, strings, and functions to discover efficient solutions.
CO4	Analyze programs through experimentation with functions, recursion, and pointers.
CO5	Implement the concepts of pointers and dynamic memory allocation to utilize memory effectively in programming.
CO6	Develop small application programs by integrating arrays, functions, pointers, and structures to meet specific problem requirements.

CO-PO Mapping

PROGRAMME OUTCOMES

COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	3	2	1	1	–	–	–	–	–	–	–
CO2	2	2	2	1	–	–	–	–	–	–	–
CO3	2	3	3	3	–	–	–	–	–	–	–
CO4	2	2	3	3	1	–	–	1	–	–	–
CO5	1	2	2	2	1	–	–	1	1	–	–
CO6	–	1	1	1	–	–	–	1	1	–	–
AVG.	2	2	2	1.83	1	–	–	1	1	–	–

ENGLISH FOR TECHNICAL WRITING

Pre-Requisite: Basic grammar, reading comprehension, writing skills, vocabulary knowledge, and willingness to participate actively are required.

Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	Name of the Subject	L	T	P	C
25AE1001	**ENGLISH FOR TECHNICAL WRITING**	**2**	**0**	**0**	**2**

	Classroom Instruction(CI) (L)	**30 Hours**	**Total no. of Hours Per Semester**
	Classroom Instruction(CI) (T)	-	
	Lab Instruction (LI) (P)	-	
	Term Work (TW) and Self Learning (SL) (TW+ SL)	**30 Hours**	**60 Hours**

SYLLABUS

	UNIT-1 VOCABULARY BUILDING	**(06 Hours)**
	1.1. Word Formation: Derivation, Compounding, Blending 1.2. Root Words from Greek and Latin: bio, geo, chrono, tele, auto, aqua 1.3. Prefixes and Suffixes: pre-, post-, inter-, -tion, -ment, -logy 1.4. Synonyms, Antonyms, and Standard Abbreviations: GDP, NASA, IIT, ASAP, AICTE, etc.	
	UNIT-2 BASIC WRITING SKILLS	**(06 Hours)**
	2.1. Types of Sentences: Simple, Compound, Complex 2.2. Use of Phrases and Clauses 2.3. Importance of Punctuation: Comma, Colon, Semicolon, Apostrophe, Quotation Marks 2.4. Creating Coherence in Sentences and Paragraphs 2.5. Paragraph Structure: Topic Sentence, Unity, Order, Coherence 2.6. Techniques for Writing Precisely	
	UNIT- 3 IDENTIFYING COMMON ERRORS IN WRITING	**(06 Hours)**
	3.1. Subject-Verb Agreement 3.2. Noun-Pronoun Agreement 3.3. Misplaced Modifiers 3.4. Prepositions: in/on, into/onto, among/between 3.5. Redundancies and Clichés: Identification and Elimination	
	UNIT- 4 NATURE AND STYLE OF SENSIBLE WRITING	**(06 Hours)**
	4.1. Describing: Person, Place, Event (Min. 100 words) 4.2. Defining: E.g., Leadership, Innovation, Integrity 4.3. Classifying: Types of Engineers, Types of Communication 4.4. Providing Examples or Evidence to support ideas 4.5. Writing Introductions and Conclusions for short essays or reports	
	UNIT-5 READING COMPREHENSION AND LITERARY APPRECIATION	**(06 Hours)**
	5.1. Reading Comprehension: Unseen passage (MCQs + Short Answer Questions) 5.2. Precis Writing: Condensing a passage (200 words to 60–70 words) 5.3. Essay Writing (200–250 words): • Role of Technology in Society • Importance of Communication in Professional Life 5.4. Critical Appreciation of Literary Texts: • “If” by Rudyard Kipling	

	• "Spoken English and Broken English" by G.B. Shaw • "On Saying Please" by A.G. Gardiner • "Where the Mind is Without Fear" by Rabindranath Tagore

	Reference Books: 01. AICTE Prescribed Textbook: English for Employability 02. G.B. Shaw – Spoken English and Broken English 03. Rudyard Kipling – If 04. A.G. Gardiner – On Saying Please 05. Rabindranath Tagore – Where the Mind is Without Fear 06. Wren and Martin – High School English Grammar and Composition 07. Raymond Murphy – English Grammar in Use

	Course Educational Objectives	
CEO1		To strengthen vocabulary, grammar, and sentence construction for academic and professional communication.
CEO2		To develop clear, coherent, and error-free writing skills.
CEO3		To enhance analytical thinking, theory-writing practices.
CEO4		To improve reading comprehension using relevant literary and non-literary texts.

	Course Outcomes:	
	Towards the end of the course, the students will be able to:	
CO1		Understand the evolution and importance of English language.
CO2		Develop vocabulary and grammar skills for professional contexts.
CO3		Apply rapid reading techniques to enhance comprehension and accuracy.
CO4		Compose error free professional writing.
CO5		Demonstrate essential soft-skills in personal and professional settings.
CO6		Exhibit proficiency in technical writing.

CO-PO Mapping

PROGRAMME OUTCOMES

COs	**1**	**2**	**3**	**4**	**5**		**6**	**7**	**8**	**9**	**10**	**11**
CO1	–	–	–	–	–		–	–	–	3	–	2
CO2	–	–	–	–	–		–	–	–	3		2
CO3	–	–	–	–	–		–	–	–	3	–	1
CO4	–	–	–	–	–		–	–	–	3	–	2
CO5	–	–	–	–	–		–	2	2	3	–	–
CO6	–	–	–	–	–		–	–	–	3	–	2
AVG.	–	–	–	–	–		–	2	2	3	–	1.8

ENVIRONMENTAL SCIENCE

Pre-Requisite: Basic understanding of Fundamental of Chemistry and Physics.
Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	Name of the Subject	L	T	P	C
25VE1001	ENVIRONMENTAL SCIENCE	2	0	0	2
	Classroom Instruction(CI) (L)	35 Hours		Total no. of Hours Per Semester	
	Classroom Instruction(CI) (T)	-			
	Lab Instruction (LI) (P)	-			
	Term Work (TW) and Self Learning (SL) (TW+ SL)	25 Hours		60 Hours	

Course Educational Objectives	
CEO1	To understand value education and self-exploration for happiness and prosperity.
CEO2	To foster harmony in self, family, society, and nature through ethical living.
CEO3	To build a holistic view of life and profession with human values and ethics.
CEO4	To evaluate global challenges using ethical and human-centric approaches.

SYLLABUS

UNIT-1 ENVIRONMENT AND ITS COMPONENTS	(07 Hours)
Ecosystem: Concept of an ecosystem. - Structure and function of an ecosystem; Producers, consumers and decomposers. Environmental gradients, Tolerance levels of environmental factors. Environmental Pollution: Definition, Cause, effects and control measures of Air pollution, Water pollution, Soil pollution, Noise pollution, Nuclear hazards. Solid Waste Management: Biomedical, Hazardous and e-waste management.	
UNIT-2 MULTIDISCIPLINARY NATURE OF ENVIRONMENTAL STUDIES	**(07 Hours)**
Importance of multilateral agreement: Stockholm and Rio Summit. Global Environmental Challenges: Global warming and climate change, acid rains, ozone layer depletion, population growth and explosion, eutrophication.	
UNIT- 3 NATURAL RESOURCES AND ITS CONSERVATION	**(06 Hours)**
Introduction and classification of resources: land resources (formation of soil, soil erosion); Water resources (Sources of fresh water, causes for the depletion of water resources); Forest resources (Deforestation, consequences of deforestation) Renewable and nonrenewable resources, Conventional and non-conventional energy resources.	

UNIT- 4 ENVIRONMENTAL LEGISLATION	(08 Hours)
Environmental Protection Act -Air (Prevention and Control of Pollution) Act.; Water (Prevention and Control of Pollution) Act; Wildlife Protection Act -Forest Conservation Act. Environmental impact Assessment: Origin and procedure of EIA, project screening for EIA, Scoping studies.	
UNIT-5 SAFETY MANAGEMENT	**(07 Hours)**
Occupational Safety and Health Acts, Safety procedures, Type of Accidents, Chemical and Heat Burns, Prevention of Accidents involving Hazardous Substances, and Human error. Hazard control measures, Fire prevention, detection and extinguishing fire, Electrical Safety, Product Safety, Safety management, Personal Protective Equipment.	

Text Books:

01. Environmental Engineering, G. Kiely, TMH, 2007
02. Environmental Engineering by Prof B.K. Mohapatra, Seven Seas Publications, 2015

Reference Books:

01. Environmental Studies, R. Rajagopalan, 3rd Edition, 2015, Oxford University Press.
02. Environmental Studies, P. N. Palanisamy, P. Manikandan, A. Geetha, and K. Manjula Rani; Pearson Education, Chennai. 2015
03. A Textbook of Environmental Studies, Shaashi Chawla, McGraw Hill Education, 2017
04. Industrial Safety Management, L. M. Deshmukh, Tata McGraw Hill Publication, 2005
05. Environmental Engineering and Safety , Raut & Sen, Scientific Publishers, 2017

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Explain value education and apply self-exploration for personal growth.
CO2	Distinguish self and body needs and show inner harmony.
CO3	Apply trust and respect to build healthy relationships.
CO4	Analyse natural interdependence and harmony in existence.
CO5	Evaluate ethical conduct in light of human values and sustainability.
CO6	Create ethical, human-focused responses to climate and social issues.

CO-PO Mapping

PROGRAMME OUTCOMES

COs	1	2	3	4	5	6	7	8	9	10	11
CO1	-	-	-	-	-	-	3	2	-	-	3
CO2	-	-	-	-	-	-	3	2	-	-	3
CO3	-	-	-	-	-	-	3	3	-	-	3
CO4	-	-	-	-	-	-	3	3	-	-	3
CO5	-	-	-	-	-	3	3	3	-	-	3

CO6	-	-	-	-	-	3	3	3	-	-	3
AVG.	-	-	-	-	-	3	3	2.7	-	-	3

APPLIED PHYSICS LABORATORY

Pre-Requisite: Good understanding of Mathematics and Physics
Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	Name of the Subject	L	T	P	C
25BS1102	**APPLIED PHYSICS LABORATORY**	**0**	**0**	**2**	**1**

Classroom Instruction(CI) (L)	-	**Total no. of Hours Per Semester**
Classroom Instruction(CI) (T)	-	
Lab Instruction (LI) (P)	**15 Hours**	
Term Work (TW) and Self Learning (SL) (TW+ SL)	**15 Hours**	**30 Hours**

SYLLABUS

(15 Hours)

01. Determination of the acceleration due to gravity by using Kater's pendulum.
02. Verification of the laws of transverse vibration by using sonometer.
03. Determination of Rigidity modulus of a wire by Static method.
04. Determination of wavelength of light by Newton's Rings apparatus.
05. Determination of no. of lines of a diffraction grating plate.
06. Determination of plank's constant using photo-voltaic cell.
07. Determination of band gap energy of PN junction (Ge/Si) diode.
08. Determination of the resistivity of a semiconductor with temperature by four- probe method.

ADD ON EXPERIMENTS

09. Determination of coefficient of Thermal conductivity (K) of a metal (Cu) by using Searle's apparatus.
10. Study of B-H curve of ferromagnetic substance.
11. Study the Hall Effect.
12 Determination of surface tension of water using Capillary- rise method.

Text Books:

1. Advanced Practical Physics for students, B.L. Flint and H.T. Workshop, 1971, Asia Publishing House.
2. Advanced level Physics Practical, Michael Nelson and Jon M. Ogborn, 4th Edition, reprinted 1985, Heinemann Educational Publishers.
3. A Text Book of Practical Physics, Indus Prakash and Ramakrishna, 11th Edition, 2011, Kitab Mahal, New Delhi.

Course Educational Objectives

CEO1	Providing fundamental information on basic instruments and their uses.
CEO2	To familiarize different apparatus and applications through different experiments.

Course Outcomes:

Towards the end of the course, the students will be able to:	
CO1	Understand the concepts of oscillation and waves through experimental observation.
CO2	Study and explain the experimental observation of interference and diffraction pattern.
CO3	Interpret the fundamental characteristics of various materials and semiconductor materials through experiments.
CO4	Analyse the quantum concept of light by experimental observation.
CO5	Study thermal conductivity and dielectric constant of a given solid.
CO6	Understand surface tension and viscosity properties of liquids.

CO-PO Mapping

PROGRAMME OUTCOMES											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	2	-		2	-	-	-	-	-	-	-
CO2	2	-		2	-	-	-	-	-	-	-
CO3	2	-		2	-	-	-	-	-	-	-
CO4	2	-		3	-	-	-	-	-	-	-
CO5	2	-		3	-	-	-	-	-	-	-
CO6	2	-		3	-	-	-	-	-	-	-
AVG.	2	-		2.5	-	-	-	-	-	-	-

BASIC ELECTRICAL AND ELECTRONICS ENGINEERING LABROTARY					
Pre-Requisite: Good understanding of Mathematics and Physics **Teaching Methods: Chalk & Board/PPT/Video Lectures**					
Subject Code	**Name of the Subject**	**L**	**T**	**P**	**C**
25ES1101	**BASIC ELECTRICAL AND ELECTRONICS ENGINEERING**	**0**	**0**	**2**	**1**
Classroom Instruction(CI) (L)		-		**Total no. of Hours Per Semester**	
Classroom Instruction(CI) (T)		-			
Lab Instruction (LI) (P)		**15 Hours**			
Term Work (TW) and Self Learning (SL) (TW+ SL)		**15 Hours**		**30 Hours**	
SYLLABUS					
					(15 Hours)

List of experiments/demonstrations:
Electrical Part:
Experiments: (Any 4 Experiments) :
01. Study of different electrical measuring equipment (Voltmeter, Ammeter & Wattmeter)
02. Verification of Thevenin's Theorem.
03. To study the connection and working of a fluorescent lamp and to learn about the

improvement of the power factor using capacitor.
04. Calculation of current, voltage and power in series R-L-C circuit excited by a single-phase AC supply and calculation of power factor.
05. No-load Test of single-phase transformer.
06. VI Characteristic of an incandescent lamp.

Electronics Part: (Any 4 Experiments):

01. Familiarization of electronic components and devices (testing of semiconductor diodes and transistors using digital multimeter)
02. Study and use of oscilloscope, signal generator to view waveforms and measure amplitude and frequency of a given waveform.
03. Study of input and output characteristics of a transistor.
04. Study and design of on half-wave and full-wave rectifier circuits without and with a capacitor filter.
05. Study and Truth table verification of different logic gates.
06. Study of Operational Amplifier circuits.

Reference Books:

1. Laboratory Courses in Electrical Engineering. S.G. Tarnekar , R.L. Kharbanda, S. Chand Publications, December 1 2006.
2. Handbook of Laboratory Experiments in Electronics Engineering Vol. 1 by A.M. Zungeru J.M. Chuma , H.U. Ezea, and M. Mangwala , 13 November 2016.

Textbooks:

1. A Handbook For The Electrical Laboratory And Testing Room V1 (1901) , by John Ambrose Fleming, Kessinger Publishing 19 March 2010.
2. Handbook of Laboratory Experiments in Electronics and Communication Engineering Vol. 2 by A.M. Zungeru J.M. Chuma M. Mangwala and L.K. Ketshabetswe , 8 March 2017.

Course Educational Objectives

CEO1	To make the students know about various electrical and electronics measuring instruments.
CEO2	To impart the knowledge of electrical DC network theorems.
CEO3	To gather the knowledge on logic gates, diode and its application.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Explain the concept of network theorems and apply them to laboratory measurements using different measuring devices.
CO2	Measure power and power factor in ac circuits using different lamps.
CO3	Understand and identify different parts of AC and DC machines.
CO4	Verify the characteristics of diodes, transistors, digital electronic components Experimentally and able to use oscilloscope, signal generator to measure amplitude and frequency of different waveforms.
CO5	Demonstrate the operating principle and VI characteristics of Semiconductor diode and transistor.
CO6	Understand the concept of various logic gates.

PROGRAMME OUTCOMES											
COs	1	2	3	4	5	6	7	8	9	10	11
CO1	3	2	–	2	2	–	–	–	1	1	–
CO2	3	2	–	2	2	–	–	–	1	–	–
CO3	3	2	1	–	2	–	–	–	–	–	–
CO4	3	2	–	2	3	–	–	–	1	1	–
CO5	3	2	–	2	2	–	–	–	–	–	–
CO6	3	2	–	–	2	–	–	–	–	–	–
AVG.	3	2	1	2	2.17	–	–	–	1	1	–

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

Pre-Requisite: Basics of Computer Science
Teaching Methods: Chalk & Board/PPT/Video Lectures

<table>
<tr><th>Subject Code</th><th>Name of the Subject</th><th>L</th><th>T</th><th>P</th><th>C</th></tr>
<tr><td></td><td>PROGRAMMING FOR PROBLEM SOLVING LABORATORY</td><td>0</td><td>0</td><td>4</td><td>2</td></tr>
<tr><td colspan="3">Classroom Instruction(CI) (L)</td><td colspan="2">-</td><td rowspan="3">Total no. of Hours Per Semester</td></tr>
<tr><td colspan="3">Classroom Instruction(CI) (T)</td><td colspan="2">-</td></tr>
<tr><td colspan="3">Lab Instruction (LI) (P)</td><td colspan="2">40 Hours</td></tr>
<tr><td colspan="3">Term Work (TW) and Self Learning (SL) (TW+ SL)</td><td colspan="2">20 Hours</td><td>60 Hours</td></tr>
<tr><td colspan="6">SYLLABUS</td></tr>
<tr><td colspan="5"></td><td>(15 Hours)</td></tr>
</table>

Lab Experiment 1: Familiarization with programming environment.
1) Introduction to OS: Before starting experiments explain the facilities and operations of OS.
2) Introduction to the C compiler, Compilation and Execution Process & writing simple programs.

Lab Experiment 2: Simple computational problems using arithmetic expressions.
1) Write a program to input radius of a circle and Find the area, perimeter of it.
2) Write a program to input 2 sides i.e: length and breadth of a rectangle. Find the area and perimeter of it.
3) Write a program to accept Fahrenheit and calculate its equivalent Celsius.

Lab Experiment 3: Simple computational problems using arithmetic expressions.
1) write a program to find simple interest when principle amount, no. of terms and rate of interest given.
[Given formula: si=(p*t*r)/100]
2) write a program to find the area of a triangle when the 3 sides of it given as input.
[Given formula: s=(a+b+c)/2 and area= sqrt(s*(s-a)*(s-b)*(s-c))]
3) John, Ram and shah were fishing in a river bank. Input the no. of fishes caught by each person and display who have caught more fishes (use conditional operator).

Lab Experiment 4: Problems involving using if statement
1) In a software company a project team of 3 members namely manvi, shyam and William. Input their job experience in no. of years. The team lead must have more experience. Display who can be team lead. (use if..else)
2) Write a program to input 3 co-efficient values and find the real roots of quadratic equation.
3) Write a program input a digit within 0 to o6. Display week day example: 0 for Sunday, 1 for Monday etc (use else if ladder).

Lab Experiment 5: Problems involving using switch..case

1) Write a program to input a lower case alphabet and test whether it is vowel or consonant.(using else..if and switch both)

2) Write a program to input an arithmetic operator and two operands. Calculate and display the result as per the given operator using switch..case.

3) Write a program to find the greatest among three numbers.(using else..if and switch both)

Lab Experiment 6: Programming with while loop and do..while loop:

1) Write a program to input a number and test whether it is prime number or not using while statement.

2) Write a program to input a positive number and test whether it is palindrome or not using do..while statement.

3) Write a program to accept a positive integer and test it for Armstrong or not using do..while statement.

Lab Experiment 7: Programming with For loops and nested Loop:

1) Write a program to generate a series of Fibonacci numbers using for statement

2) Write a program to calculate the following sum using nested for statement:

Sum = 1-(x2)/2! +(x4)/4! – (x6)/6! + (x8)/8! – (x10)/10!....(xn/n!)

3) Write a program to generate the following pyramid using nested for statement:

```
        1
      1 2 1
    1 2 3 2 1
      1 2 3 4 3 2 1
```

Lab Experiment 8: Programming with 1D Array and 2D Array

1) Write a program to accept 10 integers in to an array and find largest and smallest integers present in them.

2) Write a program to input values into a square matrix of size 3X3. Display the transpose of the matrix.

3) Write a program to input elements into two matrices A[3][4], B[4][3]. Multiply A and B store result into matrix C[3][3]. Display the resultant matrix C.

Lab Experiment 9: Programming with String Handling and User Defined Functions

1) Write a program to input a string and test it for palindrome or not using library functions.

2) Write a C program which contains three UDF's namely add(), subtract() and multiply(). Each function accepts two integers as their arguments and calculate and return the results.

3) Write a program to create an UDF which accepts an array of 10 integers and find the largest element and smallest element present in the array.

Lab Experiment 10: Programming with Recursive Functions

1) Write a program to find the factorial of a number using recursive function.

2) Write a program to accept 10 elements into an integer array. Find the largest element present using recursive function.

3) Write a program to generate Fibonacci series using a recursive function.

Lab Experiment 11: Programming with Structures

1) Write a program create a structure PRODUCT having members Product no, Name and Price. Using a pointer Input 5 product details into a structure array and then display those products whose price is >1000 rupees.

2) Write a program to create a structure EMPLOYEE to store 20 employee details having members: employee no, name, salary. Create a function which displays only those employee names whose salary>=50000.

3) Write a program to create a structure BOOKS to store 10 books details having members: book id, book name, quantity and price. Input 10 books details and create a function SHOW() for printing the books whose quantity>=100.

Lab Experiment 12: Programming with Pointers and DMA

1) Write a program to swap two numbers using User Defined Function by applying call by address concept.
2) Write a program to perform matrix addition. Create an UDF which accepts the two matrices using two pointers and performs matrix addition.
3) Write a program to store 11 cricket players' details into an array of structure. The structure having member's player name, team name and batting average. Displays the name of players whose batting average is >=30.

Topic Beyond Syllabus

1) Write a program to input a set of numbers into a file called NUM.TXT. Display only the even numbers present in the file and also display their sum.
2) Write a program to store a paragraph into a file A.TXT using command line arguments and then create a copy of it with name B.TXT.

Text Books:

01. E. Balaguruswamy, Programming in ANSI C, 7th edition, Tata McGraw-Hill
02. Let us 'C' by Yashwant Kanethekar, 16th edition, BPB Publications
03. Byron Gottfried, Schaum's Outline of Programming with C, 3rd edition, McGraw-Hill

References:

01. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language,2nd edition, Prentice
Hall of India
02. Programming in C, by Reema Thareja, 2nd edition, OUP India
03. C Programming and Coding by swati saxena, BPB Publications

Course Educational Objectives

CEO1	To develop programs for problems on different applications of array, functions, pointers and structure.
CEO2	To analyse different problems by comparing and implementing in programming.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Explain and demonstrate the fundamental features of structure-oriented programming through writing and debugging basic programs involving data types, operators, control statements, and arrays.
CO2	Classify and implement various types of statements by developing programs using control structures, arrays, functions, and structures.
CO3	Solve practical problems by applying arrays, strings, and functions, and optimize solutions through hands-on coding exercises.
CO4	Analyze program by experimenting with functions, recursion, and pointers in lab assignments to understand their impact on programs.
CO5	Implement pointer operations and dynamic memory allocation in programming tasks to achieve effective memory management.
CO6	Develop small application programs in the lab by integrating arrays, functions, pointers, and structures to address specific problem requirements.

CO-PO Mapping											
PROGRAMME OUTCOMES											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	1	1	1	1	–	–	–	–	–	–	–
CO2	2	3	2	1	–	–	–	–	–	–	–
CO3	2	2	3	3	–	–	–	–	–	–	–
CO4	2	1	3	3	–	–	–	1	–	–	–
CO5	1	1	1	2	–	–	–	1	1	–	–
CO6	–	–	1	2	–	–	–	1	1	–	–
AVG.	1.6	1.6	1.83	2	–	–	–	1	1	–	–

ENGLISH FOR TECHNICAL WRITING LABROTARY					
Pre-Requisite: Basic proficiency in English language skills (reading, writing, speaking, and listening), familiarity with computer usage, and a willingness to participate in individual and group activities **Teaching Methods: Chalk & Board/PPT/Video Lectures**					
Subject Code	**Name of the Subject**	**L**	**T**	**P**	**C**
25AE1101	**ENGLISH FOR TECHNICAL WRITING LABROTARY**	**0**	**0**	**2**	**1**
Classroom Instruction(CI) (L)		-		**Total no. of Hours Per Semester**	
Classroom Instruction(CI) (T)		-			
Lab Instruction (LI) (P)		**15 Hours**			
Term Work (TW) and Self Learning (SL) (TW+ SL)		**15 Hours**		**30 Hours**	
SYLLABUS					
UNIT-1					**(03 Hours)**
PROFESSIONAL WRITING Activity Techniques to write Circular/nonce writing **RESUME WRITING** Activity Types and format for writing an effective resume **WRITING COVER LETTER** Activity Techniques, types and format for writing over letter					
UNIT-2					**(03 Hours)**
GROUP DISCUSSION Activity-Case study basest, opinion based discussions **JOB INTERVIEW** Activity- Mock Interview **PPT PRESENTATION** Activity-Presenting any technical topics.					
UNIT- 3					**(03 Hours)**

DEBATE
Activity- Debate on current affairs.
ACTIVITIES INVOLVED IN A MEETING
Activity-Pre, during and port meeting activities. (agenda, MOM, etc.)
SWOT ANALYSIS
Activity -Creating SWOT matrix and self-analysis

UNIT- 4	**(03 Hours)**

VOCABULARY BASED ACTIVITY
Activity - Flash cards and worksheets will be given based on technical writing
Use of CALL and ESL (software) vocabulary games.
CONTENT WRITING
Activity-recording of reviews on various content and events
SOCIAL/BUSINESS ETIQUETTES
Activity: Visual presentation on social etiquette

UNIT-5	**(03 Hours)**

PROFILING A COMPANY
Activity Giving a detailed description of a company of the student's choice.
PICTURE DESCRIPTION
Activity-Gap filling
SUMMARIZING
Activity Summarize a book, research paper or news article

Reference Books:

01. Basic Communication Skills for Technology by Andrea J. Rutherford, Pearson Education Asia, Patparganj, New Delhi, 2001.
02. Business Communication by Varinder Kumar and Bodh Raj. Kalyaniya Publishers, 2104.
03. A Text book of English Phonetics for Indian Students by T.Balasubramanian, 2017.
04. Technical Communication, Principle and Practice by Meenakshi Raman & Sangeeta Sharma, Oxford Liniversity Press, 2015
05. How to Read better and Faster by NormanLewis 4th Edition Publisher: Crowell, 1978.
06. Soft Skills For Your Career by Kalyani Samantray Oxford University Press, 2016

Course Educational Objectives

CEO1	To develop communication skills of the students.
CEO2	To develop vocabulary and usage skills of students by practice.
CEO3	To enable students to participate in group discussions through listening and speaking.
CEO4	To enable students eliminate grammatical mistakes in speech and writing.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Understand the importance of effective communication for professional development.
CO2	Apply effective vocabulary and grammar in everyday communication.

CO3	Analyse various types of corporate communication.
CO4	Formulate crucial soft skills for career development.
CO5	Develop useful LSRW skills for professional growth.
CO6	Develop confidence for effective presentation.

CO-PO Mapping

PROGRAMME OUTCOMES											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	-	-	-	-	-	-	-	-	-	3	-
CO2	-	-	-	-	-	-	-	-	-	-	2
CO3	-	-	-	-	-	-	-	-	-	3	-
CO4	-	-	-	-	-	-	-	-	3	-	-
CO5	-	-	-	-	-	-	-	-	-	3	-
CO6	-	-	-	-	-	-	-	-	-	3	-
AVG.	-	-	-	-	-	-	-	-	3	3	2

ENGINEERING GRAPHICS AND DESIGN

Pre-Requisite: Elementary Maths and Physics
Teaching Methods: Chalk& Board, Computer software(AUTO CAD)

Subject Code	Name of the Subject	L	T	P	C
25ES1104	**ENGINEERING GRAPHICS AND DESIGN**	**1**	**0**	**2**	**2**

Classroom Instruction(CI) (L)	**15 Hours**	**Total no. of Hours Per Semester**
Classroom Instruction(CI) (T)	-	
Lab Instruction (LI) (P)	**15 Hours**	
Term Work (TW) and Self Learning (SL) (TW+ SL)	**30 Hours**	**60 Hours**

SYLLABUS

UNIT-1	**(05 Hours)**

Introduction: Drawing Instruments and their uses, BIS conventions, Types of line, Types of scale.
Projections : Introduction, Types, Definitions - Planes of projection, reference line and conventions employed, Orthographic Projections of points in all the four quadrants, Projections of straight lines (located in First quadrant/first angle only),True and apparent lengths, projection of line inclined to one plane and both the planes.

UNIT-2	**(08 Hours)**

Orthographic Projections of Plane Surfaces (First Angle Projection Only):
Introduction, Definitions–projections of plane surfaces–triangle, square, rectangle, pentagon, hexagon and circle, planes in different positions by change of position method, projection of inclined planes.
Orthographic Projections of Solids (First Angle Projection Only) :
Introduction, Definitions – Projections of right regular tetrahedron, hexahedron (cube), prisms, pyramids, cylinders and cones in different positions.

UNIT- 3	**(07 Hours)**

Isometric Projections: Introduction, projection of plane-square, triangular, circle, method of point and four center method
Isometric projection of solids: projection of prisms, pyramids method of drawing non-isometric lines, box method , offset method, projection some typical solids.

UNIT- 4	**(04 Hours)**

Sections and Development of Lateral Surfaces of Solids: Introduction, Section planes, Sections, Section views, Sectional views, Apparent shapes and True shapes of Sections of right regular prisms, pyramids, cylinders and cones resting with base on HP.

UNIT-5	**(06 Hours)**

Introduction to AUTO CAD: Layout of the software, standard tool bar/menus and description of most commonly used toolbars, navigational tools. Co-ordinate system and reference planes. Definitions of HP, VP, RPP &LPP. Creation of 2D/3D environment. Selection of drawing size and scale. Commands and creation of Lines, Co-ordinate points, axes, poly-lines, square, rectangle, polygons, splines, circles, ellipse, text, move, copy, off-set, mirror, rotate, trim, extend, break, chamfer, fillet, curves, constraints.

Reference Books:

01. Engineering Drawing - N.D. Bhatt & V.M. Panchal, Charotar Publishing House, Gujarat.
02. Computer Aided Engineering Drawing - S. Trymbaka Murthy, 4th Ed, University Press Engineering Drawing by N.S. Parthasarathy and Vela Murali Oxford University Press.
03. Engineering Graphics - K.R. Gopalakrishna, Subash Publishers Bangalore.
04. Fundamentals of Engineering Drawing with an Introduction to Interactive Computer Graphics for Design and Production-Luzadder Warren J., Duff John M., Eastern Economy Edition, Prentice-Hall of India Pvt. Ltd., New Delhi.
05. Computer Aided Engineering drawing, Prof. M. H. Annaiah, New Age International Publisher, New Delhi

Course Educational Objectives

CEO1	To enable students to acquire and use engineering drawing skills as a means of accurately and clearly communicating ideas, information and instructions.
CEO2	To enable students to acquire requisite knowledge, techniques and attitude required for advanced study of engineering drawing and basic of AUTOCAD.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Understand the basics of engineering drawing and method dimensioning in engineering systems.

CO2	Prepare the orthographic projections of points and straight lines placed in various quadrants.
CO3	Demonstrate the ability of drawing orthographic projections of various planes & solids parallel, perpendicular and inclined to different planes.
CO4	Sketch the isometric view of different planes and solids by showing all their features in 3D.
CO5	Draw and interpret the sectioned views of solids.
CO6	Apply the software skills to prepare engineering drawings using AutoCAD.

CO-PO Mapping

PROGRAMME OUTCOMES											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	3	-	-	1	-	-	-	-	-	-	-
CO2	2	-	2	2	-	-	-	-	-	-	-
CO3	2	-	2	2	-	-	-	-	-	-	-
CO4	2	-	2	2	-	-	-	-	-	-	-
CO5	2	-	2	2	-	-	-	-	-	-	-
CO6	3	-	2	-	3	-	-	-	-	-	-
AVG.	2.3	-	2	1.8	3	-	-	-	-	-	-

APPLIED MATHEMATICS-II (EXCEPT BIOTECH)

Pre-Requisite: Differential Calculus, Functions and Properties, Matrices.
Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	**Name of the Subject**	**L**	**T**	**P**	**C**
25BS1004	**APPLIED MATHEMATICS-II**	**2**	**1**	**0**	**3**
	Classroom Instruction(CI) (L)	**30 Hours**		**Total no. of Hours Per Semester**	
	Classroom Instruction(CI) (T)	**15 Hours**			
	Lab Instruction (LI) (P)	-			
	Term Work (TW) and Self Learning (SL) (TW+ SL)	**45 Hours**		**90 Hours**	

Course Educational Objectives

CEO1	To focus on partial derivative and its methods.
CEO2	To make them understand about Laplace and Fourier and transform.
CEO3	To calculate the gradients and directional derivatives of functions of several variables.

CEO4	To introduce the concept of Vector differentiation and integration that finds applications in various fields like solid mechanics, fluid flow, heat problems and potential theory.

SYLLABUS

UNIT-1 SEQUENCES AND SERIES	**(09 Hours)**

Sequence and series: Convergent, Divergent and Oscillatory Series, D'Alembert's Ratio Test, Cauchy's Root Test, Power Series, Radius of Convergence, Power Series Solution of ODE of First order and Second order.

UNIT-2 VECTOR CALCULUS	**(09 Hours)**

Vector and scalar functions, Directional Derivatives, gradient, divergence, curl, and their properties with simple applications. Evaluation of double and triple integrals for vector-valued functions. Evaluation of line integral, Surface integral, and volume integrals, Green's Theorem, Stoke's Theorem and Gauss Divergence Theorem and their simple applications.

UNIT- 3 COMPLEX VARIABLE	**(09 Hours)**

Continuity and differentiability of complex valued functions, Cauchy-Riemann equations, Analytic function, Harmonic Function, Conjugate Harmonic Function, Construction of Analytic functions, Properties of analytic functions, Conformal Mappings (Rotation, translation and inversion), Bilinear Transformations.

UNIT- 4 COMPLEX INTEGRATION	**(09 Hours)**

Line integral, Cauchy's integral theorem, Cauchy's integral formula. Taylor's series, Laurent's series, Classification of Singularities, zeros of analytic functions, Residues, Cauchy Residue theorem, Evaluation of indefinite integrals.

UNIT-5 FOURIER SERIES	**(09 Hours)**

Fourier Series, Dirichlet's conditions, Fourier series expansion of functions with period 2π, Fourier series expansion of functions with period 2L, Even and odd functions, Half Range Fourier Cosine and Sine Series.
.

Reference Books:

1. Advanced Engineering Mathematics by E. Kreyszig, Tenth Edition, Willey
2. Higher Engineering Mathematics by B.V.Ramana, McGraw Hills Education
3. Higher Engineering Mathematics by BS Grewal : Khanna Publishers, New Delhi
4. Advanced Engineering mathematics by H. K. Dass.
5. Differential Calculus by J. Sinha Roy & S. Padhi

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Identifying the nature of sequences and series and solving ordinary differential equations using power series methods.
CO2	Utilize Laplace and inverse Laplace transforms, and solve linear ordinary differential equations up to second order.
CO3	Calculate gradient, divergence, and curl of vector fields and apply Green's, Stokes', and Gauss' theorems to evaluate line, surface, and volume integrals in vector fields.

CO4	Determine aanalyticity through Cauchy-Riemann equations and identifying harmonic functions, and performing conformal mappings.
CO5	Apply Cauchy's integral theorem, Taylor's and Laurent's series, and classify singularities, compute residues, and evaluate real integrals using the residue theorem.
CO6	Develop mathematical concepts using series, Laplace transforms, vector calculus, and complex analysis to solve complex engineering and applied science problems.

CO-PO Mapping

PROGRAMME OUTCOMES

COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-
CO5	2	3	-	-	-	-	-	-	-	-	-
CO6	2	3	-	-	-	-	-	-	-	-	-
AVG.	2.5	3	-	-	-	-	-	-	-	-	-

APPLIED MATHEMATICS-II (ONLY FOR BIOTECH)

Pre-Requisite: Differential Calculus, Functions and Properties, Matrices.
Teaching Methods: Chalk & Board/PPT/Video Lectures

Subject Code	**Name of the Subject**	**L**	**T**	**P**	**C**
25BS100B4	**APPLIED MATHEMATICS-II**	**2**	**1**	**0**	**3**

Classroom Instruction(CI) (L)	**30 Hours**	**Total no. of Hours Per Semester**
Classroom Instruction(CI) (T)	**15 Hours**	
Lab Instruction (LI) (P)	-	
Term Work (TW) and Self Learning (SL) (TW+ SL)	**45 Hours**	**90 Hours**

Course Educational Objectives

CEO1	To Study the coordinate system.
CEO2	To study scalar and vector functions.
CEO3	To explore the standard methods for solving algebraic equations.

CEO4	To study the Statistical and its properties.

SYLLABUS

UNIT-1 CO-ORDINATE GEOMETRY	**(09 Hours)**

Straight Lines: Introduction, Slope of a line and angle between two lines. Various forms of equations of a line: parallel to axes, point-slope form, slope-intercept form, two-point form, intercepts form and normal form. General equation of a line. **Circles:** Circle, Equation of circle whose centre and radius is known, General equation of a circle, equation of tangent and normal Equation of circle passing through three given points, Equation of circle whose diameters is line joining two points (x_1, y_1) & (x_2, y_2).

UNIT-2 DIFFERENTIAL CALCULUS	**(09 Hours)**

Limit and Continuity, Differentiability, Derivatives using first principle, Derivatives of sum, difference, product and quotient of two functions, Differentiation of functions (Trigonometric, Algebraic, Exponential, Logarithmic functions), Chain Rule, Differentiation by substitution method.

UNIT- 3 INTEGRAL CALCULUS	**(09 Hours)**

Integration of elementary functions, Integration by substitution method, Definite Integrals, Integration by parts, Integration by partial fraction, Area under curves.

UNIT- 4 STATISTICS	**(09 Hours)**

Mean, Median, Mode, Correlation and Regression analysis, fitting of straight line by method of least squares method. Testing of hypothesis (small sample tests –t, F and chi-square test).

UNIT-5 VECTOR ANALYSIS	**(09 Hours)**

Scalars and Vectors, Algebra of Vectors, Position Vectors, Scalar Product , Vector Product, Unit Vectors, Scalar Triple Product and Vector Triple Product, Vector Functions, Derivatives, Gradient, Divergence, Curl and Their Simple Applications.

Reference Books:

1. Advanced Engineering Mathematics by E. Kreyszig, Tenth Edition, Willey.
2. Differential Calculus by Santi Narayan and Mittal, S. Chand Publications.
3. Higher Engineering Mathematics by BS Grewal: Khanna Publishers, New Delhi.
4. Higher Engineering Mathematics by B. V. Ramana, McGraw Hills Education.
5. Advanced Engineer methods by N. P. Bali & Manish Goyal.

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Analyze equations of straight lines, solving problems related to circles, tangents, normal, and circles passing through specific points.
CO2	Evaluate limits and derivatives using first principles and rules of differentiation.

CO3	Solve the problems involving indefinite and definite integrals and compute areas under simple curves.
CO4	Apply fundamental concepts of probability, including addition and multiplication theorems, conditional probability, and Bayes' theorem.
CO5	Use vector algebra, and compute gradient, divergence, and curl with basic applications.
CO6	Apply combined knowledge from geometry, calculus, probability, and vector analysis to solve interdisciplinary problems encountered in science and engineering contexts.

CO-PO Mapping

PROGRAMME OUTCOMES

COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	2	3	-	-	-	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-
CO5	2	3	-	-	-	-	-	-	-	-	-
CO6	2	3	-	-	-	-	-	-	-	-	-
AVG.	2.33	3	-	-	-	-	-	-	-	-	-

SEMESTER III [SECOND YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	BSC	XXBS2001	Discrete Mathematics	2	1	0	3	10	30	60	100
2	PCC	XXPC2001	Digital Electronics	3	0	0	3	10	30	60	100
3	PCC	XXPC2002	Database Management System	3	0	0	3	10	30	60	100
4	PCC	XXPC2003	Computer Organization and Architecture	3	0	0	3	10	30	60	100
5	ESC	XXES2001	Object-Oriented Programming Through JAVA	3	0	0	3	10	30	60	100
6	HSMC	HSHS2001	Engineering Economics	2	0	0	2	10	30	60	100
7	PCC	XXPC2101	Digital Electronics Laboratory	0	0	2	1	40	-	60	100
8	PCC	XXPC2102	Database Management System Laboratory	0	0	2	1	40	-	60	100
9	ESC	XXES2101	Object-Oriented Programming Through JAVA Laboratory	0	0	2	1	40	-	60	100
10	AEC	HSAE2101	Language Proficiency and Communication Laboratory	0	0	2	1	40	-	60	100
11	IKS	25IK2301	Indian Knowledge Systems	-	-	-	1	-	-	-	-
			Total:	**16**	**1**	**8**	**22**	**220**	**180**	**600**	**1000**

DISCRETE MATHEMATICAL STRUCTURE

Pre-Requisite: Fundamentals of sets, functions and relation, algebraic operation, idea of graph
Teaching Methods: Chalk & Board, PPT

Subject Code	Name of the Subject	L	T	P	C
	DISCRETE MATHEMATICAL STRUCTURE	2	1	0	3

Classroom Instruction(CI) (L)	30 Hours	Total no. of Hours Per Semester
Classroom Instruction(CI) (T)	15 Hours	
Lab Instruction (LI) (P)	-	
Term Work (TW) and Self Learning (SL) (TW+ SL)	45 Hours	90 Hours

Course Educational Objectives

CEO1	To know logical equivalence and to apply in various proofs.
CEO2	To know about relation and recurrence relation.
CEO3	To study Boolean Algebra and its properties.
CEO4	To know about Graphs and trees.

SYLLABUS

UNIT-1 MATHEMATICAL LOGIC AND SET THEORY **(09 Hours)**

Propositional logic, Propositional Equivalence, Predicate Calculus, and Quantifiers Statement Calculus, Nested Quantifiers, Proof methods and Strategies, Mathematical Induction.

UNIT-2 RECURRENCE RELATION **(09 Hours)**

Recurrence relation, Solution to recurrence relation, generating functions, Inclusion and exclusion principle, Relation and their properties, Closure of relations, Equivalence relations, Partial order relations, Partially Ordered Set.

UNIT- 3 BOOLEAN ALGEBRA AND LATTICES **(09 Hours)**

Lattices, Distributive and Complemented Lattices, Sub-lattices, Boolean Lattices and Boolean algebra, Boolean Functions and Boolean Expressions.

UNIT- 4 GROUP THEORY **(09 Hours)**

Semigroups, Monoids, Groups, Abelian Group, Cyclic group, Subgroups, Ideals, Langrange's Theorem, Permutation groups, Normal subgroups, Homomorphism, isomorphism, Kernel.

UNIT-5 GRAPH THEORY **(09 Hours)**

Basic Definitions, Different types of Graphs, Matrix Representation of Graphs, Paths and circuits, Eulerian Theorem, Hamiltonian Theorem, connected graphs, Graph Coloring, Spanning Tree, Binary Tree, Minimum Spanning tree, Kruskal's algorithm, Prim's algorithm.

Reference Books:

01. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Sixth Edition, 2008, Tata McGraw Hill Education, New Delhi
02. C. L. Liu and D. Mohaptra, "Elements of Discrete Mathematics", Third Edition, 2008,
03. N. Deo, Graph Theory and Applications to Engineering and Computer Science, Prentice Hall of India
04. Discrete Mathematics by Schaum's Outlines (Second Edition)
05. Ralph P. Grimaldi," Discrete and Combinatorial Mathematics", Fifth Edition, 2005, Pearon Education, New Delhi

Course Outcomes:

Towards the end of the course, the students will be able to:

CO1	Understand of propositional logic, predicate calculus, and quantifiers is crucial for analyzing logical statements and constructing valid arguments.
CO2	Apply proof techniques such as direct proof, contradiction, induction, and recurrence relations to establish mathematical correctness.
CO3	Analyze combinatorial problems using set theory, inclusion-exclusion principles, and generating functions for problem-solving in discrete mathematics.
CO4	Evaluate different types of relations, equivalence relations, partial orders, and lattice structures to model and interpret real-world scenarios.
CO5	Create Boolean functions, Boolean expressions, and algebraic structures like semi-groups, monoids, and groups for computational and logic circuit applications.
CO6	Apply graph theory concepts, including Eulerian and Hamiltonian graphs, graph coloring, spanning trees, and minimum spanning tree algorithms (Kruskal's & Prim's), to solve real-world optimization problems.

CO-PO Mapping

	PROGRAMME OUTCOMES										
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-
CO5	2	3	-	-	-	-	-	-	-	-	-
CO6	2	3	-	-	-	-	-	-	-	-	-
AVG.	2.5	3	-	-	-	-	-	-	-	-	-

Course Code	Name of the Course	L	T	P	C
XXBS2001	**Discrete Mathematics**	**2**	**1**	**0**	**3**

Prerequisite: Applied Mathematics-I (25BS1001) & Applied Mathematics-II (25BS1004)

Course Educational Objectives

CEO1	To know logical equivalence and to apply in various proofs.
CEO2	To know about relation and recurrence relation.
CEO3	To study Boolean Algebra and its properties.
CEO4	To know about Graphs and trees.

Course Outcomes: Towards the end of the course students will be able to :

CO1	Understanding the concept of logic, groups, lattice. graphs and trees
CO2	Memorize the logical gates, recurrence formulas and different types of graphs
CO3	Evaluate the arguments, predicates and quantifiers, solution of recurrence relations, methods of solution
CO4	Analyze recurrence relation by generating functions and groups, binary trees
CO5	Categorize different types of graphs, and its property
CO6	Solve the logical equivalences, Boolean algebra and pre order and post order of arithmetic Operations.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	1	2											
CO2	2	3											
CO3	2	3											
CO4	2	3											
CO5	2	3											
CO6	2	3											
AVG	1.83	2.83											

SYLLABUS

UNIT I - MATHEMATICAL LOGIC AND SET THEORY (09 hours)

Propositional logic, Propositional Equivalence, Predicate Calculus, and Quantifiers Statement Calculus, NestedQuantifiers, Proof methods and Strategies, Mathematical Induction

UNIT II - RECURRENCE RELATIONS (09 Hours)

Recurrence relation, Solution to recurrence relation, Generating functions, Inclusion and exclusion Principle. Relations and their properties, Closure of relations, Equivalence relations, Partial ordering relations, Partially ordering sets.

UNIT III - BOOLEAN ALGEBRA AND LATTICES (09 Hours)

Lattices, Distributive and Complemented Lattices, Boolean Lattices and Boolean algebra, Boolean Functions, CNF, DNF and Boolean Expressions

UNIT IV- GROUP THEORY (09 Hours)

Semi groups, Monoids, Groups, Subgroups, and Permutation groups, Normal subgroups, Homomorphism,isomorphism, Kernel.

UNIT V - GRAPH THEORY (09 Hours)

Basic Definitions, Some Special Graphs, Matrix Representation of Graphs, Paths and circuits, Eulerian Theorem, Hamiltonian Theorem, connected graphs, Graph coloring, Trees - Spanning Trees, Binary Trees, Minimum Spanning tree, Kruskal's algorithm, Prim's algorithm, Tree Traversal.

Text Books:

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", 9th Edition, 2025, Tata McGraw Hill Education, New Delhi, ISBN: 9781266191541
2. C. L. Liu and D. Mohaptra, "Elements of Discrete Mathematics", Fourth Edition, reprint 2018, McGraw Hill Education (India), ISBN : 9781259006395

Reference Books:

1. N. Deo, Graph Theory and Applications to Engineering and Computer Science, reprint

2018,Prentice Hall of India, ISBN : 9788120301450

2. Discrete Mathematics by Schaum's Outlines ,Fourth Edition,2022, McGraw Hill, ISBN : 9781264258802
3. Ralph P. Grimaldi," Discrete and Combinatorial Mathematics", Fifth Edition, 2005, Pearson Education, New Delhi, ISBN: 978-0321385024

Subject Code	Name of the Subject	L	T	P	C
XXPC2101	Digital Electronics Laboratory	3	0	1	4

Course Educational Objectives

Pre -Requisite: DIGITAL ELECTRONICS(XXPC2001)

CEO1	To provide practical exposure to digital logic design concepts using gates, combinational and sequential circuits.
CEO2	To develop skills in designing and testing combinational logic and sequential circuits using hardware and HDL.
CEO3	To enable students to implement counters, registers, and memory units for real-time applications.

Course Outcomes: Upon successful completion of this course, students should be able to:

CO1	Demonstrate the functionality of basic logic gates and implement Boolean functions using universal gates.
CO2	Design and implement combinational circuits such as adders, subtractors, multiplexers, encoders, decoders, and code converters.
CO3	Analyze and implement sequential circuits including flip-flops, shift registers, and counters.
CO4	Investigate the operation of memory units and programmable logic devices through simulation and hardware implementation.
CO5	Develop Verilog/VHDL models for combinational and sequential circuits and verify through simulation.
CO6	Work effectively in a team to conduct experiments, analyze results, and prepare technical reports.

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	3	2	-	-	2	-	-	-	-	-	-	3	2
CO2	3	3	3	2	2	-	-	-	-	-	-	3	2
CO3	3	3	3	2	2	-	-	-	-	-	-	3	2
CO4	3	3	3	2	3	-	-	-	-	-	-	3	3
CO5	3	2	3	2	3	-	-	-	-	-	-	3	3
CO6	-	-	-	-	-	-	-	3	3	2	-	-	2
Average	3	2.60	3	2	2.40	-	-	3	3	2	-	3	2.33

List of Experiments

Experiment 1 : Digital Logic Gates: Investigate the logic behavior of AND, OR, NAND, NOR, EX-OR, EXNOR, Inverter, and Buffer Gates, use of Universal NAND/NOR Gate.
Experiment 2: Gate-Level Minimization: Two-level and multi-level implementation of Boolean functions.**s**
Experiment 3: Combinational Circuits: Design, assemble, and test: adders and subtractors, code converters, gray code to binary, and 7-segment displays.
Experiment 4: Design, implement, and test a given design example with (i) NAND Gates only, (ii) NOR Gates only, and (iii) using the minimum number of Gates'
Experiment 5: Design with Multiplexers and De-multiplexers
Experiment 6: Flip-Flop: Assemble, test, and investigate the operation of SR, D & JK flip-flops
Experiment 7: Shift Registers: Design and investigate the operation of all types of shift registers with parallel load
Experiment 8: Counters: Design, assemble, and test various ripple and synchronous counters - decimal counter, Binary counter with the parallel load

Experiment 9: Memory Unit: Investigate the behavior of the RAM unit and its storage capacity - 16×4 RAM: testing, simulating, and memory expansion
Experiment 10: Clock-Pulse Generator: Design, implement, and test
Experiment 11: Parallel Adder and Accumulator: Design, implement, and test
Experiment 12: Binary Multiplier: Design and implement a circuit that multiplies 4-bit unsigned numbers to produce an 8-bit product.
Experiment 13: Verilog/VHDL simulation and implementation of Experiments listed in Sl. No. 3 to 12.
Text Books: 1. M. Morris Mano, and Michael D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog. Pearson Education, Sixth Edition, 2018, ISBN: 978-9353062019 2. A. Anand Kumar, Fundamentals of Digital Circuits. PHI Learning Pvt. Ltd., New Delhi, Fourth Edition, 2016 ISBN: 9788120352681
Reference Books: 1. John P. Uyemura, A First Course in Digital Systems Design: An Integrated Approach. Thomson Press (India) Ltd., India Edition, 2012, ISBN: 9788131501085 2. William H. Gothmann, Digital Electronics: An Introduction to Theory and Practice, PHI Learning Pvt. Ltd., New Delhi, Se Edition, 2006,ISBN: 9788120303485 3. D.V. Hall, Digital Circuits and Systems. Tata McGraw-Hill Education Pvt. Ltd., 1989. ISBN: 9780070255371 Charles H. Roth, Digital System Design using VHDL. Tata McGraw-Hill Education Pvt. Ltd., Second Edition, 2012.ISBN: 9788131518304

Course Code	Name of the Course	L	T	P	C
XXPC2101	**Database Management System Lab**	**0**	**0**	**2**	**1**

Prerequisite: Database Management System (XXPC2001)	
Course Educational Objectives	
CEO1	Provide students with practical knowledge of SQL commands for creating, manipulating, querying, and managing relational databases.
CEO2	Develop the ability to design database schemas, enforce integrity constraints, and retrieve meaningful information using SQL queries and joins.
CEO3	Enable students to design and implement real-world database applications using relational database concepts while maintaining data integrity and consistency.
Course Outcomes: Towards the end of the course students will be able to:	
CO1	Apply SQL commands to create, modify, and manage databases and tables using DDL and DML statements.
CO2	Retrieve and manipulate data efficiently using SQL operators, functions, filtering techniques, grouping, and sorting.
CO3	Design relational database tables by implementing integrity constraints to ensure data consistency and accuracy.
CO4	Perform data retrieval from multiple related tables using different types of SQL joins.
CO5	Design and implement real-world database applications using SQL queries, views, stored procedures, and database objects.

CO6	Analyze database requirements and develop complete database solutions for practical applications such as student, library, and supermarket management systems.

CO-PO & PSO Mapping

	PROGRAMME OUTCOMES											PSOs	
COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	–	2	–	–	–	–	–	–	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	2
CO3	3	3	2	2	2	–	–	–	–	–	–	3	2
CO4	3	3	3	2	3	–	–	–	–	1	–	3	3
CO5	3	3	3	2	3	–	–	–	1	1	–	3	3
CO6	3	3	3	3	3	–	–	–	1	2	1	3	3
Average	3.00	2.83	2.33	1.67	2.50	0.00	0.00	0.00	0.33	0.67	0.17	2.83	2.33

List of Experiments

Experiment 1: Basics of MySQL
Introduction to the MySQL environment and execution of SQL statements. Declaration and use of variables, implementation of numeric, character, and date data types, and application of arithmetic, comparison, and logical operators for evaluating SQL expressions.

Experiment 2: SQL Commands
Implementation of SQL Data Definition Language (DDL) commands for creating, selecting, renaming, and deleting databases. Familiarization with database management operations using MySQL.

Experiment 3: DDL and DML Commands
Creation and modification of database tables using DDL commands and manipulation of records using DML commands. Operations include insertion, retrieval, updating, deletion, alteration of table structure, renaming of tables, truncation, and deletion of database objects.

Experiment 4: SQL Filtering Operators
Retrieval and manipulation of data using SQL filtering operators and clauses. Implementation of comparison, logical, and pattern matching operators, sorting of records, grouping and aggregation of data, and arithmetic operations on table attributes.

Experiment 5: SQL Functions in Employee Management System
Development of an Employee Management System to demonstrate the use of SQL built-in functions. Implementation of string, numeric, date, and aggregate functions for processing employee information and generation of analytical reports using SQL functions.

Experiment 6: SQL Constraints
Implementation of SQL integrity constraints to ensure data accuracy and consistency in relational databases. Creation and modification of database tables using PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, and DEFAULT constraints, along with validation of constraint enforcement through SQL operations.

Experiment 7: SQL Joins
Creation of related database tables and retrieval of information using different types of SQL joins. Implementation of Equi Join, Non-Equi Join, Self Join, Cartesian Join, and ANSI SQL joins including INNER, LEFT OUTER, RIGHT OUTER, and FULL OUTER JOIN for combining data from multiple tables.

Experiment 8: Case Study – Student Information System

Design and implementation of a relational database for a Student Information System. The experiment includes database design, table creation, data manipulation, integrity constraints, joins, aggregate functions, views, and stored procedures to manage students, courses, and enrollments.

Experiment 9: Case Study – Supermarket Product Sales System
Design and implementation of a relational database for a Supermarket Product Sales System. The experiment focuses on managing products, customers, and sales transactions through SQL queries, joins, aggregate functions, views, stored procedures, and database modification operations.

Experiment 10: Case Study – Library Management System
Design and implementation of a relational database for a Library Management System. The experiment covers the management of books, members, and issue-return transactions using SQL queries, integrity constraints, joins, aggregate functions, views, stored procedures, and database administration operations.

Text Books

1. **Paul DuBois**, *MySQL*, **5th Edition**, Addison-Wesley Professional, 2019. ISBN: **978-0135957059**.
2. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education, 2017, ISBN: 978-0133970777.

Reference Books

1. **Ben Forta**, *SQL in 10 Minutes, Sams Teach Yourself*, **5th Edition**, Sams Publishing, 2023. ISBN: **978-0135182796**.
2. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database System Concepts, 8th Edition, McGraw-Hill Education, 2022, ISBN: 978-1260515046.
3. Carlos Coronel and Steven Morris, Database Systems: Design, Implementation and Management, 14th Edition, Cengage Learning, 2023, ISBN: 978-0357673034.
4. Thomas Connolly and Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation and Management, 6th Edition, Pearson Education, ISBN: 978-1292061184.
5. Paul DuBois, MySQL, 5th Edition, Addison-Wesley Professional, ISBN: 978-0321833877.

Course Code	Name of the Course	L	T	P	C
XXPC2001	**Database Management System**	**3**	**0**	**1**	**4**

Prerequisite: Programming for Problem Solving (25ES1003)

Course Educational Objectives

CEO1	To provide students with a strong foundation in database management systems, data modeling, relational database concepts, and SQL.
CEO2	To develop the ability to design, organize, and manage relational databases using normalization, transaction management, concurrency control, and recovery techniques.
CEO3	To familiarize students with modern database technologies, including NoSQL, cloud, and vector databases, and their applications in contemporary computing environments.

Course Outcomes: Towards the end of the course students will be able to

CO1	Understand the fundamentals of database management systems, database architecture, and the ER model for conceptual database design.
CO2	Apply ER-to-relational mapping, relational algebra, relational calculus, and SQL for relational database operations.
CO3	Identify functional dependencies and apply normalization techniques to reduce data redundancy in relational databases.
CO4	Understand transaction concepts, concurrency control protocols, and locking mechanisms in database systems.
CO5	Identify database failures and explain recovery techniques for maintaining database consistency.
CO6	Understand the concepts of modern database systems, including NoSQL, cloud, and vector databases.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
CO1	3	2	1	–	–	–	–	–	–	–	–	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	2
CO3	3	3	3	2	1	–	–	–	–	–	–	3	2
CO4	3	3	2	2	1	–	–	–	–	–	–	3	2
CO5	2	2	1	–	2	–	–	–	–	–	–	2	3
CO6	2	2	1	–	3	–	–	–	–	–	–	2	3
Average	2.7	2.5	1.7	1.7	1.8	–	–	–	–	–	–	2.5	2.2

SYLLABUS

UNIT:1 Introduction to Database Systems **(9 Hours)**

Database Systems Concepts: Characteristics of Database Systems, Database Users, Database Languages, Database System Architecture, Three-Level Architecture, Schemas and Instances, Data Abstraction, Data Independence, Data Models, Database Constraints **Entity–Relationship (ER) Model:** Basic Concepts, Design Process, Attributes, Entity Sets, Relationship Sets, Keys, Mapping Constraints, ER Diagrams, Weak Entity Sets, Extended ER Features

UNIT:2 Relational Model and Relational Query Languages **(9 Hours)**

Relational Model and its terminologies, Relational Database Constraints, Mapping the ER model to Relational Model, Relational Algebra – Selection, Projection, Union, Intersection, Set Difference, Cartesian Product, Rename, Join, Division. Structured Query Language (SQL) – Introduction, Data Definition Language (DDL) —CREATE, ALTER, DROP; Data Manipulation Language (DML) —INSERT, UPDATE, DELETE; Data Query Language (DQL) —SELECT, WHERE, ORDER BY, GROUP BY, Aggregate Functions, Join Operations (Inner and Outer Joins). Relational Calculus – Tuple relational calculus, domain relational calculus.

UNIT3: Normalization **(9 Hours)**

Introduction – Purpose of normalization, Functional Dependencies (Full, Partial, Transitive), Lossless Decomposition & Dependency Preservation, Normal Forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), Boyce–Codd Normal Form (BCNF), Multivalued Dependencies (MVDs), Fourth Normal Form (4NF), Join Dependencies, Fifth Normal Form (5NF).

UNIT4: Transaction and Recovery **(9 Hours)**

Transaction Concept, ACID Properties, Transaction States, Schedules, Serializability, Concurrency Control, Lock-Based Protocols, Timestamp-Based Protocol, Two-Phase Locking (2PL), Deadlocks. Recovery and Atomicity: Types of Database Failures, Recovery Concepts, Transaction Logs, Log Based Recovery, Rollback (Undo) Recovery, Commit (Redo) Recovery, Checkpoint Recovery

UNIT5: Modern Database Systems **(9 Hours)**

Evolution of Modern Databases, NoSQL Databases: Basic Concepts, Characteristics, CAP

Theorem, Types (Key–Value, Document, Column-Family and Graph Databases),Cloud Databases: Basic Concepts, Database as a Service (DBaaS), Advantages and Applications, Vector Databases: Basic Concepts, Vector Embeddings, Similarity Search,Comparison of Relational, NoSQL and Vector Databases, Applications of Modern Database Systems.

TEXT BOOK

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database System Concepts, 8th Edition McGraw-Hill Education, 2022-24 ISBN: 978-0078022159
2. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2020–2021 ISBN: 978-1260084504

REFERENCE BOOKS

1. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2003. ISBN: 978-0072465631.
2. C. J. Date, An Introduction to Database Systems, 8th Edition, Addison-Wesley, 2003. ISBN: 978-0321197849.
3. Thomas M. Connolly and Carolyn E. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition, Pearson, 2015. ISBN: 978-1292061184.
4. Pramod J. Sadalage and Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 1st Edition, Addison-Wesley Professional, 2012. ISBN: 978-0321826626.
5. Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, 1st Edition, O'Reilly Media, 2017. ISBN: 978-1449373320

Course Code	Name of the Course	L	T	P	C
XXPC2001	DIGITAL ELECTRONICS	3	0	0	3

Prerequisite:

Course Educational Objectives

CEO1	Introduce number systems, binary codes, and arithmetic operations essential for digital system design.
CEO2	Explain Boolean algebra, logic simplification techniques, and implementation using logic gates.
CEO3	Develop the ability to analyze and design combinational logic circuits for practical applications.

Course Outcomes: Towards the end of the course students will be able to :

CO1	Explain and interpret number systems, binary codes, and perform binary arithmetic operations.
CO2	Apply Boolean algebra and K-maps to simplify and implement logic expressions.
CO3	Analyze and design combinational logic circuits for practical applications.
CO4	Explain and design sequential logic circuits and finite state machines using flip-flops.
CO5	Design and implement registers and counters for digital applications.
CO6	Analyze and compare different memory and PLDs for digital system design applications.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	-	-	-	-	-	-	-	-	-	3	2
CO2	3	3	2	-	-	-	-	-	-	-	-	3	2
CO3	3	3	3	2	2	-	-	-	-	-	-	3	2
CO4	3	2	3	2	2	-	-	-	-	-	-	3	2
CO5	3	2	3	2	3	-	-	-	-	-	-	3	3
CO6	3	3	3	-	3	-	-	-	-	-	-	3	3
AVG	3	2.50	2.80	2	2.50	-	-	-	-	-	-	3	2.33

SYLLABUS

UNIT 1 - Number Systems and Codes (08 hours)

Binary, Octal, Hexadecimal, and Decimal Number System and their Conversion; Representation of Signed Binary and Floating-Point Number; Binary Arithmetic using 1's and 2's Complements, Binary Codes - BCD Code, Gray Code, ASCII Character Code

UNIT 2 -Boolean Algebra and Logic Gates (09 Hours)

Axioms and Laws of Boolean Algebra; Reducing Boolean Expressions; Logic Levels and Pulse Waveforms; Logic Gates; Exclusive-OR Function; Boolean Expressions and Logic Diagrams; Canonical and Standard Forms. Gate-level Minimization-NAND and NOR Implementation; K-maps - Two, Three, and Four Variable K-maps, Don't-Care Conditions.

UNIT 3 - Combinational Logic (10 Hours)

Combinational Circuits; Analysis Procedure; Design Procedure; Adders; Subtractors; Parallel Binary Adders; Binary Adder-Subtractor; Binary Multiplier; Magnitude Comparator; Code Conversion; Decoders; Encoders; Multiplexers; De-multiplexers.

UNIT 4- Synchronous Sequential Logic (10 Hours)

Sequential Circuits; Storage Elements: Latches, Flip-Flops, Master-Slave Flip-Flop; Conversion of Flip-Flops; Analysis of Clocked Sequential Circuits; Mealy and Moore Models of Finite State Machines; Design Procedure.

UNIT 5 - Registers and Counters (08 Hours)

Shift Registers; Data Transmission in Shift Registers; SISO, SIPO, PISO, and PIPO Shift Registers; Counters; Asynchronous Counters; Synchronous Counters; Design of Synchronous Counters. Memory and Programmable Logic- Introduction; Random-Access Memory; Read-Only Memory; Programmable Logic Array; Programmable Array Logic.

Text Books:

1. M. Morris Mano, and Michael D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog. Pearson Education, Sixth Edition, 2018, ISBN: 978-9353062019
2. A. Anand Kumar, Fundamentals of Digital Circuits. PHI Learning Pvt. Ltd., New Delhi, Fourth Edition, 2016 ISBN: 9788120352681

Reference Books:

4. John P. Uyemura, A First Course in Digital Systems Design: An Integrated Approach. Thomson Press (India) Ltd., India Edition, 2012, ISBN: 9788131501085
5. William H. Gothmann, Digital Electronics: An Introduction to Theory and Practice, PHI Learning Pvt. Ltd., New Delhi, Se Edition, 2006,ISBN: 9788120303485
6. D.V. Hall, Digital Circuits and Systems. Tata McGraw-Hill Education Pvt. Ltd., 1989. ISBN: 9780070255371
7. Charles H. Roth, Digital System Design using VHDL. Tata McGraw-Hill Education Pvt. Ltd., Second Edition, 2012.ISBN: 9788131518304

Course Code	Name of the Course	L	T	P	C
HSHS2001	Engineering Economics	2	0	0	2

Prerequisite: Applied Mathematics- I & II (25BS1001 & 25BS1004)

Course Educational Objectives

CEO1	To apply economic concepts to support engineering decision-making.
CEO2	To evaluate project feasibility using cost analysis and investment appraisal.
CEO3	To analyze financial and economic data for engineering projects.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Explain fundamental economic concepts, laws of demand and supply, and market equilibrium.
CO2	Calculate elasticity of demand, price/output under perfect competition, and perform break-even analysis.
CO3	Apply time value of money concepts and interest formulas to evaluate investment alternatives.
CO4	Analyse methods of depreciation and perform cost-benefit analysis for engineering projects.
CO5	Describe the role of banking, inflation, and national income in the economic environment affecting engineering decisions.
CO6	Interpret economic data and make informed recommendations for engineering economic decisions.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2				1							
CO2	2	3											
CO3	2	1			3	1					2		
CO4											3		
CO5	1					3							
CO6	1	1	3								1		
AVG	1.8	1.7	3		3	1.6					2		

SYLLABUS

UNIT I - Fundamentals of Microeconomics for Engineering Applications (10 Hours)

Meaning, Nature, Scope of Engineering Economics, Law of Demand and its exceptions, Determinants of demand, Elasticity of demand & its measurement (Simple numerical problems to be solved), Law of supply and its exception, Determination of market equilibrium (Simple numerical problems to be solved). Law of variable proportion, Law of returns to scale.

UNIT II -Market Structures and Cost Analysis (10 Hours)

Cost and revenue function, Price and output Determination under perfect competition (Simple numerical problems to be solved), Break Even Analysis - Linear approach (Simple numerical problems to be solved)

UNIT III - Time Value of Money and Investment Analysis (10 Hours)

Time value of money – Application of interest formulas, Cash flow diagrams, Present worth method, Future worth method, Annual worth method, Internal rate of return method.

UNIT IV- Project Evaluation and Depreciation (8 Hours)

Cost benefit analysis for public projects, Depreciation-Depreciation of capital assert, causes of depreciation, Methods of calculating depreciation (Straight line method, Declining balance method), Replacement Analysis

UNIT V - Overview of Banking, Inflation, and National Income (7 Hours)

Functions of commercial bank, Functions of Central Bank. Inflation-Meaning of inflation, types, causes, measures to control inflation. National Income - Concepts of national income & Method of measuring national income.

Text Books:

1. Deviga Vengedasalam, Karunagaran Madhavan, Principles of Economics, 3rd Edition, Oxford University Press, 2014, ISBN: 978-9834712754
2. R. Panneerselvam, Engineering Economics, 2nd Edition, PHI Learning, 2013, ISBN: 978-8120348370

Reference Books:

1. J. L. Riggs, D. D. Bedworth, S. U. Randhawa, Engineering Economics, 4th Edition, McGraw Hill Education, 2004, ISBN: 978-0070586703
2. Chan S. Park, Fundamentals of Engineering Economics, 4th Edition, Pearson, 2019, ISBN: 978-1292264851
3. William G. Sullivan, Elin M. Wicks, C. Patrick Koelling, Engineering Economy, 18th Edition, Pearson, 2025, ISBN: 978-1292477763
4. Gerald J. Thuesen, W. J. Fabrycky, Engineering Economy, 9th Edition, Pearson, 2000, ISBN: 978-0130281289
5. H. L. Ahuja, Principles of Microeconomics, 22nd Edition, S. Chand Publishing, 2024, ISBN: 978-9352837311
6. R. R. Paul, Money, Banking and International Trade, Kalyani Publishers, 2018, ISBN: 978-9327289572

Text Books:

1. E Balaguruswamy, 7th Edition 2023, Programming with Java, McGraw Hill Education, ISBN: 9789355325891
2. Herbalt Schelidt(13TH Edition 2024), Java: The Complete Reference, McGraw Hill Education, ISBN: 9789355326478

Reference Book:

1. SachinMalhotra/ SauravChoudhary, Second Edition, Programming in Java, Oxford Higher Education, 2nd Edition 2014, ISBN : 9780198094852
2. RashmiKanta Das, Core Java For Beginners, Vikas Publication,3rd Edition 2013, ISBN: 9788125950837
3. R Nageswara Rao(2016), Core Java: An Integrated Approach, Dreamtech Press. ISBN : 9789351199250

Course Code	Name of the Course	L	T	P	Credits
XXES2101	Object Oriented Programming Through Java Lab	0	0	2	1

Pre -Requisite: Object Oriented Programming Through Java(XXES2001)

Course Educational Objectives

CEO1	To introduce students to the fundamental concepts of Object-Oriented Programming (OOP) and familiarize them with Java syntax, control structures, arrays, and basic input/output mechanisms.
CEO2	To impart hands-on experience in implementing core OOP principles such as encapsulation, inheritance, polymorphism, and interfaces to build robust and reusable software components.
CEO3	To enable students to understand and apply advanced Java features, including exception handling, packages, string manipulation, multithreading, and file I/O operations for real-world problem-solving.

Course Outcome: Towards the end of the course students will be able to :

CO1	Learn to use object orientation to solve problems and use java language to implement them
CO2	To experiment with the syntax and semantics of java language and gain experience with java programming Course Outcomes:
CO3	Ability to create user friendly interfaces
CO4	Ability to write portable programs which work in all environments
CO5	Ability to solve the problem using object oriented approach and design solutions which are robust
CO6	Ability to design GUI using AWT/Swing

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2												
CO2	2	2	2											
CO3	2	3	3	1										
CO4	2	2	3	2	1				1					
CO5	2	3	2						1	1				
CO6	1	2												
	2.00	2.33	1.67	0.50	0.17				0.33	0.17				

Experiment-1(I/O in Java)

Q 1. Write a Program that accepts two Strings as command line arguments and generate the output in the required format

Q 2. Write a Java program to find if the given number is prime or not.

Q 3. Write a program to print the sum of all the digits of a given number.

Q 4. Write a Java program to find if the given number is palindrome or not

Q 5. Write a program to print first 5 values which are divisible by 2, 3, and 5.

Experiment 2: (Array)

Q 1. Write a program to initialize an integer array and find the maximum and minimum value of the array.

Q 2. Write a program to initialize an integer array with values and check if a given number is present in the array or not.

Q 3. Write a program to find the largest 2 numbers and the smallest 2 numbers in the given array.

Q 4. Write a program to remove the duplicate elements in an array and print the same.

Q 5. Write a program to reverse the elements of a given 2*2 array. Four integer numbers needs to be passed as Command Line arguments.

Experiment 3:(Class, Object, Static method, Constructor)

Q 1 Create a class Box that uses a parameterized constructor to initialize the dimensions of a box. The dimensions of the Box are width, height, depth. The class should have a method that can return the volume of the box. Create an object of the Box class and test the functionalities.

Q 2. Create a new class called Calculator with the following methods:

1. A static method called powerInt(int num1,int num2)

This method should return num1 to the power num2.

2. A static method called powerDouble(double num1,int num2).

This method should return num1 to the power num2.

3. Invoke both the methods and test the functionalities.
Hint: Use Math.pow(double,double) to calculate the power.
Q 3 Design a class that can be used by a health care professional to keep track of a patient's vital statistics. The following are the details.
Name of the class - Patient
Member Variables -
patientName(String),height(double),width(double)
Member Function - double computeBMI()
The above method should compute the BMI and return the result. The formula for computation of BMI is weight (in kg) ÷ height*height(in metres).
Create an object of the Patient class and check the results.
Q 4 Create a class Author with the following information.
Member variables : name (String), email (String), and gender (char)
Parameterized Constructor: To initialize the variables
Create a class Book with the following information.
Member variables : name (String), author (of the class Author you have just created), price (double), and qtyInStock (int)
[Assumption: Each book will be written by exactly one Author]
Parameterized Constructor: To initialize the variables
Getters and Setters for all the member variables
In the main method, create a book object and print all details of the book (including the author details)

Experiment 4: (Inheritance and polymorphism)

Q 1. Create a class named 'Animal' which includes methods like eat() and sleep(). Create a child class of Animal named 'Bird' and override the parent class methods. Add a new method named fly(). Create an instance of Animal class and invoke the eat and sleep methods using this object. Create an instance of Bird class and invoke the eat, sleep and fly methods using this object.
Q 2. Create a class called Person with a member variable name. Save it in a file called Person.java
Create a class called Employee that will inherit the Person class.T he other data members of the Employee class are annual salary (double), the year the employee started to work, and the national insurance number which is a String.Save this in a file called Employee.java
Your class should have the necessary constructors and getter/setter methods. Write another class called TestEmployee, containing a main method to fully test your class definition.
Q 3. Create a school application with a class called Person. Create name and dateOfBirth as member variables.
Create a class called Teacher that inherits from the Person class. The teacher will have additional properties like salary, and the subject that the teacher teaches.
Create a class called Student that inherits from Person class. This class will have a member variable called studentId.
Create a class called College Student that inherits from Student class. This class will have collegeName, the year in which the student is studying (first/second/third/fourth) etc.
Create objects of each of this classes, invoke and test the methods that are available in these classes.
Q 4. Create a base class Fruit with name ,taste and size as its attributes.
Create a method called eat() which describes the name of the fruit and its taste.
Inherit the same in 2 other classes Apple and Orange and override the eat() method to represent each fruit taste.
Q 5. Write a program to create a class named shape. It should contain 2 methods, draw() and erase() that prints "Drawing Shape" and "Erasing Shape" respectively.
For this class, create three sub classes, Circle, Triangle and Square and each class should override the parent class functions - draw () and erase ().
The draw() method should print "Drawing Circle", "Drawing Triangle" and "Drawing Square" respectively.
The erase() method should print "Erasing Circle", "Erasing Triangle" and "Erasing Square" respectively.
Create objects of Circle, Triangle and Square in the following way and observe the polymorphic nature of the class by calling draw() and erase() method using each object.
Shape c=new Circle();

Shape t=new Triangle();
Shape s=new Square();

Experiment 5: (Interface and Wrapper Class)

Q 1. Write a program to create an interface Animal with a method makeSound(). Implement it in a class Dog and display appropriate output.

Q 2. Write a program to create an interface Shape with a method draw (). Implement it in two classes: Circle and Rectangle. Call both using interface reference.

Q 3 Write a program to create an interface Greeting with: One abstract method sayHello(), One default method sayBye(). Implement and demonstrate both methods.

Q 4. Write a java program to create an interface NumberPrinter with method printNumber(Integer num). Implement and display the number.

Q 5. Write a java program to create two interfaces:

Printer → print()

Scanner → scan()

Implement both in a single class and demonstrate functionality.

Q 6. Write a program using an interface Vehicle with method start (). Implement it in classes Car and Bike.

Experiment 6: (String and StringBuffer)

Q 1. Write a Program to check whether a given String is Palindrome or not.

Q 2. Write a java program that will concatenate 2 strings and return the result. The result should be in lowercase.

Q 3 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.

Q 4. Write a java program that will return the first half of the string, if the length of the string is even. It should return null for odd length string.

Q 5. Write a java program that accepts a string and returns a new string without the first and last character of the input string.

Q 6. Given a string and an integer n, print a new string made of n repetitions of the last n characters of the string. You may assume that n is between 0 and the length of the string, inclusive.

Experiment 7: (Package and Exception Handling)

Q 1. Write a program to create a package and include a class inside it. Demonstrate how to compile and run the package program.

Q 2. Write a program to demonstrate the use of different access specifiers (public, protected, default, private) within and outside a package.

Q 3 Write a program to create a user-defined package for mathematical operations and use it in another class.

Q 4. Write a program that handles multiple exceptions such as ArrayIndexOutOfBoundsException and ArithmeticException.

Q 5. Write a program to demonstrate the use of a finally block that executes regardless of exception occurrence.

Q 6. Write a program to create and use a user-defined exception (e.g. insufficient balance).

Experiment 8: (Multithreading and Inner class)

Q 1. Write a program to create a thread using the Runnable interface that prints a message continuously for a few iterations.

Q 2. Write a program to create two threads: One prints even numbers, One prints odd numbers. Execute both simultaneously.

Q 3 Write a program to create a user-defined package for mathematical operations and use it in another class.

Q 4. Write a program that handles multiple exceptions such as ArrayIndexOutOfBoundsException and ArithmeticException.

Q 5. Write a program to demonstrate the use of a finally block that executes regardless of exception occurrence.

Q 6. Write a program to create and use a user-defined exception (e.g. insufficient balance).

Experiment 9: (File Handling)

Q 1. Write a Java program to count the number of characters in a file.

Q 2. Write a program to copy contents from one file to another file.

Q 3 Write a program to replace all occurrences of a particular word in a file with another word.

Q 4. Write a Java program to display only those lines from a file that contain a given keyword.

Q 5. Write a program to read a file and display its contents in reverse order (line by line).

Q 6. Write a program to count the frequency of each word in a file.

Experiment 10: (AWT and Event Handling)

Q 1. Create a program with two buttons: "Show" and "Exit". Handle their actions using event handling.

Q 2. Write a Java AWT program to add two numbers entered in text fields when a button is clicked.

Q 3 Create a program to count how many times a button is clicked and display the count.

Q 4. Create a program that changes the text in a label when a key is pressed using KeyListener interface in Java.

Q 5. Write a Java AWT program where selecting different options from checkboxes displays selected items when a button is clicked.

Q 6. Develop a Java AWT login form and validate username and password on button click.

Experiment 11: (SWING and Event Handling)

Q 1. Write a Java Swing program to create a window using JFrame class in Swing and display a button. Show a message when the button is clicked using ActionListener interface in Java.

Q 2. Write a program to properly close a window using WindowAdapter class in Java.

Q 3 Write a Swing program to display mouse click coordinates using MouseListener interface in Java.

Q 4. Write a Swing program where selecting multiple options using checkboxes displays selected items when a button is clicked.

Q 5. Create a Swing application to change font style and size of text using buttons or menu items.

Q 6. Create a simple calculator using Swing components and handle operations through button events.

Course Code	**Name of the Course**	**L**	**T**	**P**	**C**
XXES2001	**Object Oriented Programming Through Java**	3	0	0	3

Course Educational Objectives	
CEO1	The model of object-oriented programming: abstract data types, encapsulation, inheritance, and polymorphism
CEO2	Fundamental features of an object-oriented language like Java: object classes and interfaces, exceptions, and libraries of object collections
CEO3	How to take the statement of a business problem and from this determine suitable logic for solving the problem; then be able to proceed to code that logic as a program written in Java.

Course Outcomes: Towards the end of the course students will be able to :	
CO1	Knowledge of the structure and model of the Java programming language.
CO2	Use the Java programming language for various programming technologies.
CO3	Develop software in the Java programming language.
CO4	Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.
CO5	Propose the use of certain technologies by implementing them in the Java programming language to solve the given problem.
CO6	Choose an engineering approach to solving problems, starting from the acquired knowledge of programming.

CO-PO & PSO Mapping (Articulation Matrix):

COs	**PROGRAMME OUTCOMES**												**PSOs**	
	01	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**01**	**02**
01	3	1											1	
02	3	2											1	

03	2	2	2										2	
04	2	1	2										1	
05	2	1	3										1	
06	1	3	3										1	
Avg.	2.17	1.67	2.5										1.33	

Syllabus		
UNIT I	**Introduction to Object Oriented Programming**	**10 Hours**
Features of Object Oriented Programming Introduction to Java. Features of Java, First Java Program, Architecture of JVM. Taking Input from keyboard-Command Line Arguments, Scanner Class, BufferedReader class. Java Tokens-Datatypes, Operators, Control Structures, Conditional Statements, Iteration Statement/Loops, Jumping Statements,Java Arrays - 1-dimensional array, array copying, 2-dimensional array, Jagged Array		
UNIT II	**Classes and Objects**	**15 Hours**
Instance variables, methods, creating objects, method overloading, Constructors, static keyword, this keyword, Array of Objects, Access Modifiers (public, private, protected, default). Inheritance- Types of Inheritance, Method overriding, Method overloading Vs Method Overriding, Usage of super keyword, Usage of final keyword. Polymorphism- Upcasting, Dynamic Method Dispatch. Abstract class and Interface-abstract method, abstract class, Usage of interface in inheritance, abstract class vs interface, marker interface. String Handling in Java-String and its methods, StringBuffer and its methods, StringBuilder and its methods. Wrapper classes-Types of wrapper classes, Auto boxing and unboxing, methods of wrapper class		
UNIT III	**Java Packages**	**12 Hours**
User-Defined Packages, Accessing Packages. Java Exception Handling-Types of exceptions, Hierarchy of Exception classes, Java exception keywords: try, catch, finally, throw, throws, User defined exceptions. Multithreading- Thread in Java, Life cycle of a thread, Thread creation, Thread scheduler, Thread execution prevention methods: sleep (), join () and yield (), Thread priority, Concept of Synchronization, Inter Thread Communication, Basics of Deadlock, Demon Thread. Inner Classes-Introduction, Member inner class, static inner class, local inner class, anonymous inner class.		
UNIT IV	**Java IO**	**10 Hours**
Java IO-IO Streams (java.io package), Byte Stream and Character Stream, Serialization and deserialization, Role of transient keyword in serialization. AWT-The AWT class hierarchy, AWT components and containers, Usage of Frame class. Event Handling- java.awt.event package – event classes, Event Delegation Model (EDM), listener interfaces, ActionListener in event handling, Event handling using TextArea, Checkbox, CheckboxGroup, Choice box.		

UNIT V	**Listener and Adapter Classes**	**10 Hours**
Listener and Adapter Classes-Handling MouseListener, MouseMotionListener Interface, WindowListener Interface, Adapter classes. Swings-Introduction to Swings, Difference between AWT and Swing, Hierarchy of swing components. Containers, Top level containers -JFrame, JWindow, JDialog, JPanel, JButton, JToggleButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JComboBox, JTabbedPane, JTree.		

Text Books:

3. E Balaguruswamy, 7th Edition 2023, Programming with Java, McGraw Hill Education,ISBN: 9789355325891
4. R Nageswara Rao(2016), Core Java: An Integrated Approach, Dreamtech Press. ISBN : 9789351199250
5. Herbalt Schelidt(13TH Edition 2024), Java: The Complete Reference, McGraw Hill Education, ISBN: 9789355326478

Reference Book:

4. SachinMalhotra/ SauravChoudhary, Second Edition, Programming in Java, Oxford Higher Education, 2nd Edition 2014, ISBN : 9780198094852
5. RashmiKanta Das, Core Java For Beginners, Vikas Publication,3rd Edition 2013,ISBN: 9788125950837

Course Code	Name of the Course	L	T	P	C	
XXPC2002	Computer Organization and Architecture	3	0	0	3	

Course Educational Objectives

CEO1	Identify the functional units in a digital computer system.
CEO2	Distinguish between the various ISA styles and trace the execution sequence of an instruction through the processor.
CEO3	Compare different approaches used for implementing a functional unit and evaluate different computer systems based on performance metrics.

Course Outcomes: Towards the end of the course, Students will be able to:

CO1	Understand the basics of instructions sets and their impact on processor design
CO2	Demonstrate an understanding of the design of the functional units of a digital computer system.
CO3	Develop and implement assembly language program for real time application.
CO4	Design a pipeline for consistent execution of instructions with minimum hazards
CO5	Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
CO6	Manipulate representations of numbers stored in digital computers.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2											1		
CO2	3	3	3	1										1	
CO3	2	2	2	3									1		
CO4	3	3	2	3										2	
CO5	3	3	3	2									1		
CO6	2	2	2											1	
Avg.	2.6	2.5	2	1.5									1		

SYLLABUS

UNIT:1 Fundamentals of Computer Architecture & Organization **(10 Hours)** Functional Units of a computer, Basic Operational Concepts- instruction execution cycle, data flow, Bus Structures- single bus, multiple bus, system interconnects, Memory Organization- memory location, byte addressability, hierarchy, Instruction Set Architecture- instruction formats, addressing modes, RISC vs CISC, modern ISA examples (ARM, x86, GPU), Performance Metrics- latency, throughput, CPI, MIPS, Amdahl's Law- limits of parallel speedup, Pipeline Processing- instruction pipelining, hazards, superscalar execution.

UNIT:2 Assembly Language Program **(9 Hours)** Instruction Basics , Instruction Format, Types of Instruction Set – data transfer, arithmetic/logic, control, I/O, stack operations, Addressing Modes – immediate, direct, indirect, register, indexed, relative, Instruction Sequencing & Branching – program flow, conditional/unconditional jumps, loops, Basic Programs – arithmetic operations, array processing, string manipulation, Stack & Subroutines – push/pop, procedure calls, parameter passing, recursion basics, Basic Input/output Operations – keyboard input, display output, system calls.

UNIT:3 Arithmetic Logic Unit **(8 Hours)**

Carry Look-Ahead Adder – fast addition using parallel carry computation, Fast Adders techniques like carry-save and carry-select adders, Binary Multiplication – basic shift-and-add method, Booth's Algorithm – efficient signed multiplication, Fast Multiplication Methods – Wallace tree, array multipliers, Binary Division Methods – restoring, non-restoring, SRT division, Floating Point Representation – IEEE 754 standard, normalized numbers, Floating Point Arithmetic – addition, subtraction, multiplication, division.

UNIT:4 Basic Processing Unit **(9 Hours)**

Processor Components – ALU, registers, control unit, memory interface, Data Path & Control – instruction cycle, control signals, execution flow, Instruction Execution – fetch, decode, execute, memory access, write-back, Control Unit Design – hardwired vs microprogrammed control, Pipelining Concepts – instruction pipeline stages, throughput improvement, Pipeline Hazards – structural, data, control hazards; hazard mitigation techniques, Flynn's Classification – SISD, SIMD, MISD, MIMD architectures, Memory Architectures – UMA Vs NUMA.

UNIT:5 Memory & Peripheral Devices (10hours)

Memory Types – primary, secondary, tertiary, Memory Hierarchy, Cache Memory – levels (L1, L2, L3), mapping functions, performance considerations, Virtual Memory – paging, segmentation, demand paging, Secondary Memory – magnetic disks, solid-state drives, optical storage, Direct Memory Access (DMA) – high-speed data transfer bypassing CPU, Interrupts – hardware/software interrupts, priority handling, I/O Interfaces – need for standards; PCI, SCSI, USB, SATA, NVME

Text Books:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky and Naraig Manjikian Computer Organization and Embedded Systems, 6th Edition, McGraw-Hill Education, 2022, ISBN: 9781260076974

2. David A. Patterson & John L. Hennessy Computer Organization & Design, 6th Edition, Morgan Kaufmann / Elsevier, 2020. ISBN: 9780128201097

Reference Books:

1. M. Morris Mano – Digital Design *Computer System Architecture*, 3rd Edition, Prentice Hall -Pearson, 2017**.** ISBN: 9789332585607

2. William Stallings – Computer Organization and Architecture-Designing, 11th Edition, Pearson Education, 2018. ISBN: 9780134092669

3. John P. Hayes – Computer Architecture & Organization, 4th Edition, McGraw-Hill Education, 2003. ISBN: 9780072463149

SEMESTER IV [SECOND YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	PCC	XXPC2004	Operating System	3	0	0	3	10	30	60	100
2	PCC	XXPC2005	Computer Networks	3	0	0	3	10	30	60	100
3	PCC	XXPC2006	Ethical Software Engineering	3	0	0	3	10	30	60	100
4	PCC	XXPC2007	Design and Analysis of Algorithm	3	0	0	3	10	30	60	100
5	ESC	XXES2001	Python Programming	3	0	0	3	10	30	60	100
6	HSMC	HSHS2002	Organizational Behaviour	2	0	0	2	10	30	60	100
7	PCC	XXPC2104	Design and Analysis of Algorithm Laboratory	0	0	2	1	40	-	60	100
8	PCC	XXPC2105	Computer Networks Laboratory	0	0	2	1	40	-	60	100
9	ESC	XXES2101	Python Programming Laboratory	0	0	2	1	40	-	60	100
10	AEC	HSAE2101	Workplace Communication and Presentation Laboratory	0	0	2	1	40	-	60	100
11	VSEC	XXSE2201	Micro Project (Skill Development Project and Hands on Training)	0	0	4	2	-	-	100	100
12	AU	25AU2301/ 25AU2302	Constitution of India/ Essence of Indian Knowledge Tradition	2	0	0	0	-	-	-	-
			Total:	**19**	**0**	**12**	**23**	**220**	**180**	**700**	**1100**

Course Code	**Name of the Course**	**L**	**T**	**P**	**C**	
XXPC2005	**Computer Networks**	**3**	**0**	**0**	**3**	
Course Educational Objectives						
CEO1	To discuss the digital data communication techniques					

CEO2	Gain knowledge on basic concepts of data communication layers, protocols and performance
CEO3	Understand a few representative protocols and network components

Course Outcomes: Towards the end of the course students will be able to:

CO1	Demonstrate the fundamental concepts and components of data communication and transmission media.
CO2	Analyze and apply data link layer protocols and error handling techniques for reliable Communication.
CO3	Evaluate network layer protocols, routing algorithms, and IP addressing schemes.
CO4	Implement transport layer protocols and manage end-to-end communication effectively.
CO5	Identify and use application layer protocols for various data communication services.
CO6	Explore advanced topics in data communication and assess their impact on modern network architectures.

CO-PO MAPPING

Cos	**PROGRAMMEOUTCOMES**												**PSOs**		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	1	1										1	1	
CO2	3	1	1										2	1	1
CO3	2	2	1										3	2	1
CO4	2	3	1	1									2	2	1
CO5	2	1	1										2	1	
CO6	2	2	2	3									2	1	
Avg.	2.3	1.6	1.1	0.6									2	1.3	0.5

SYLLABUS

UNIT 1: Introduction to Computer Networks (9 Hours)

Introduction to Computer Networks, Components, Devices, Topology, OSI model, layers in OSI model, TCP/IP Model, Address types (Physical, Logical, Port, Application-Specific Addresses), Analog and Digital Signal. Transmission Impairement(Attenuation, Distortion, Noise), Data Rate Limits(Nyquist Bit rate, Shannon Capacity). Performance (Bandwidth , Throughput, latency, Jitter).Transmission Media (Guided Media, Unguided Media).

UNIT 2: Physical Layer (9 Hours)

line encoding, Block coding, Binary data to Digital Signal Conversion, Digital to Analog Signal Conversion, Analog to Analog Signal Conversion, Analogue to Digital Signal conversion. Digital Transmission types (Serial and Parallel), Transmission Modes (Simplex, Half-Duplex and Full-Duplex), Multiplexing (FDM, TDM, WDM, CDM).

Switching and Types: Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

UNIT 3: Data link layer (9 Hours)

Introduction, Error Detection and Correction techniques (Checksum, Cyclic Redundancy Check, Hamming Code (1-bit error rectification)), Framing, HDLC Frame Structure and Types, HDLC (NRM, ABM, ARM), Point to Point Protocol, Flow and Error Control, Noiseless Channels protocols (Simplest, stop–and-Wait), Noisy Channels protocols (Stop-and-Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ).

UNIT 4: Medium Access sub layer & Network Layer (9 Hours) Multiple Access - Random Access Protocols (ALOHA, CSMA (non-persistent, 1-persistent, p-persistent), CSMA/CD, CSMA/CA), Controlled Access Protocols (Reservation, Polling, Token Passing), Channelization techniques (FDMA, TDMA, CDMA). Network Layer- Inter-networking, IP Address, IPv4 (Header structure, Class types, sub-netting), IPv6 (Header structure, Address structure), IPV4 to IPV6 Address Mapping, Address Resolution Protocol, ICMP.

UNIT 5: Transport Layer & Application Layer (9 Hours)

Transport Layer-User Datagram Protocol, Transmission Control Protocol, Process-To-Process Delivery, Congestion Control and its Type, Quality of Service. Application Layer- Domain Name System - Address Resolution, TELNET, E-MAIL (SMTP, POP3), FTP, HTTP.

Text Book:

1. Data Communications and Networking – Behrouz A. Forouzan, Fourth Edition TMH,2006,ISBN: 9780070634145
2. Computer Networks -- Andrew S Tanenbaum, 6th Edition,2021, Pearson Education, ISBN: 9789356063600

Reference Book:

1. An Engineering Approach to Computer Networks-S.Keshav,2nd Edition 1997, Pearson Education ,ISBN: 9780201634426
2. Understanding communications and Networks,3rd Edition 2004, W. A. Shay, Cengage Learning, ISBN: 9788178001791
3. Computer and Communication Networks , Nader F. Mir, Pearson Education,2nd Edition 2014, ISBN: 9780133814835
4. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, K.W. Ross, 9th Edition 2025, Pearson Education, ISBN: 9789356061316

Course Code	Name of the Course	L	T	P	C
XXPC2105	**Computer Network Lab**	**0**	**0**	**2**	**1**

CEO1	To introduce the foundational hardware devices, transmission media topologies, and structural commands essential to modern computer networks.
CEO2	To provide practical instruction on network layer and application layer flow profiling using industry-standard protocol analyzer utilities
CEO3	To impart hands-on skills in configuring switching/routing infrastructures and executing network performance simulations
Course Outcomes: Towards the end of the course students will be able to :	
CO1	Understand the fundamental concepts and devices of a computer network.
CO2	Analyze the network packets (like TCP/UDP, DNS) using protocol analyzers to understand real-world data flow.
CO3	Simulate network traffic, congestion control, and routing algorithms (e.g., Link State, Distance Vector) to test performance.
CO4	Configure and operate network devices like switches and routers using basic administrative commands.
CO5	Implement and analyze error detection and correction codes (e.g., CRC, Hamming) to ensure reliable data transmission
CO6	Develop a network related project for real time data transmission.

COs	**PROGRAMME OUTCOMES**												**PSOs**	
	PO1	**PO**	**PO**	**PO4**	**PO5**	**PO**	**PO**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**1**	**2**
CO1	3	1	1										1	1
CO2	3	1	1										2	1
CO3	2	2	1										3	2
CO4	2	3	1	1									2	2
CO5	2	1	1										2	1
CO6	2	2	2	3									2	1
Avg.	2	2	1	2									2	1

LIST OF EXPERIMENT
EXPERIMENT 1: Study of different types of Network Devices.
EXPERIMENT 2: Study of different types of Network Cable for transmission.
EXPERIMENT 3: To design different network topology and measure its performance.
EXPERIMENT 4: Different types network commands in Linux and Windows.
EXPERIMENT 5: Creating a network by implementing different topologies through LAN trainer software Supported Netsim tool kit.
EXPERIMENT 6: Measure the performance of token bus and token ring protocols through simulation.
EXPERIMENT 7: Method for transmission flow control using Netsim2.
EXPERIMENT 8: Write a java program to demonstrate checksum.
EXPERIMENT 9: Write a java program to demonstrate hamming code.
PROJECT DEVELOPMENT
EXPERIMENT 10: Develop a micro chat application between client and server where the specification is as follows - client will write first to the server then server will receive and print the text. Then server will write to the client and client will receive and print the text. The step goes on.
Text Books: **1)** Data Communications and Networking with TCP/IP Protocol Suite – Behrouz A. Forouzan,6th Edition 2022, ISBN: 9781260597820 **2)** Computer Networks – Andrew S. Tanenbaum, Nick Feamster, David J. Wetherall,6th Edition 2022,ISBN: 9789356063600
Reference Books: **1)** Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide – Laura Chappell,2nd Edition 2012,ISBN: 9781893939943 **2)** Computer Networking: A Top-Down Approach – James F. Kurose, Keith W. Ross,9th Edition 2025,ISBN: 9780135415603

Course Code	Name of the Course	L	T	P	C
XXPC2104	**Design and Analysis of Algorithms Lab**	**0**	**0**	**2**	**1**

Pre -Requisite:	
CEO1	To provide a solid foundation in Python programming language, enabling students to write, debug, and execute programs using basic programming constructs such as variables, data types, control structures, functions, and modules.
CEO2	To develop problem-solving skills through algorithmic thinking, allowing students to design and implement Python solutions for real-world problems using structured programming techniques.
CEO3	To introduce essential programming paradigms and tools in Python, including file handling, exception handling, and the basics of object-oriented programming (OOP), thereby equipping students for further study and application development.

Course Outcome: Towards the end of the course students will be able to	
CO1	Implement algorithms based on brute force, divide and conquer, and greedy techniques to solve basic computational problems.
CO2	Apply dynamic programming and backtracking methods to develop optimized solutions for complex problems.
CO3	Design and implement graph algorithms such as BFS, DFS, shortest path, and minimum spanning tree.
CO4	Analyze the time and space complexity of algorithms through experimental and theoretical evaluation.
CO5	Demonstrate proficiency in selecting appropriate algorithmic strategies for given problems and justify their efficiency.
CO6	Develop the ability to work individually or in teams to design, implement, and test algorithmic solutions in a programming environment.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2											
CO2	2	2	2										
CO3	2	3	3	1									
CO4	2	2	3	2	1				1				
CO5	2	3	2						1	1			
CO6	1	2											
	2.00	2.33	1.67	0.50	0.17				0.33	0.17			1.02

EXPERIMENT 1: Implementation of different Recursive algorithms

(a) Factorial () (b) GCD () (c) Power () (d) Tower_of_Hanoi ()
(e) Fibonacci ()

EXPERIMENT 2: Implementation of linear search and binary search

EXPERIMENT 3: Sort Algorithms

1. Write a program on the Quick Sort algorithm.
2. Write a program on Heap sort using a max heap.
3. Write a program on the Merge Sort algorithm.

EXPERIMENT 4: Using dynamic programming

4. Implementation of Matrix-chain Multiplication algorithm.
5. Implementation of LCS algorithm.

EXPERIMENT 5: Using a Greedy Approach

6. Write a program for solving the Activity Selection Problem
7. Write a program for solving the Fractional Knapsack Problem

EXPERIMENT 6: Graph Algorithms

8. Implement DFS/ BFS for a connected graph.
9. Implement MST using Kruskal / Prim algorithm.
10. Implementation of Dijkstra's Single Source Shortest Path finding algorithm.

EXPERIMENT 7: String Matching

11. Implementation of Knuth-Morris-Pratt String-Matching Algorithm.
12. Implementation of Rabin-Karp String-Matching Algorithm

EXPERIMENT 8: Backtracking

13. Find out the solution to the N-Queen problem.
14. Find out the solution to Sub Set Sum Problem.

Text Book:

1. Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, 4th Edition 2022, PHI Learning Pvt. Ltd, ISBN: 9780262046305

2. Fundamentals of Algorithm, Horowitz & Sahani:, 2nd Edition 2008,Universities Press, ISBN: 9788173716126

Reference Book:

1. Algorithms, Design and Analysis, H. Bhasin, First Edition 2015, Oxford University, ISBN: 9780199456666
2. Design and Analysis of Algorithm, S. Sridhar, Oxford University Press,1st Edition 2014, ISBN:9780198093695
3. Algorithms, Sanjay Dasgupta, UmeshVazirani , McGraw-Hill Education,1st Edition 2023, ISBN: 9789355325525

Course Code	Name of the Course	L	T	P	C	
XXPC2007	Design and Analysis of Algorithms	3	0	0	3	

Course Educational Objectives

CEO1	Analyze the asymptotic performance of algorithms
CEO2	Demonstrate familiarity with major algorithms
CEO3	Apply important algorithmic design paradigms and methods of analysis

Course Outcomes: Towards the end of the course students will be able to

CO1	Apply efficient algorithms to reduce space and time complexity of both recurrent and non-recurrent relations
CO2	Solve problems using divide and conquer approaches
CO3	Apply greedy and dynamic programming types techniques to solve polynomial time problems
CO4	Create exponential problems using backtracking and branch and bound approaches
CO5	Interpret various approximation algorithms and interpret solutions to evaluate P type, NP Type, NPC, NP Hard problems
CO6	Create algorithms that are efficient in space and time complexities by using divide conquer, greedy, backtracking technique

CO-PO & PSO Mapping

COs	PROGRAMMEOUTCOMES												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	1	1	1												
CO2	2	2	2												
CO3	3	1	1												
CO4	3	1	1												
CO5	2	2	3												
CO6	1	3	2												
Avg.	**2.00**	**1.67**	**1.67**												

SYLLABUS

Unit I : Introduction **(9 Hours)**

Introduction to Algorithm, Definition, Characteristics of the algorithm, Performance Analysis of Algorithms: Space and Time Complexity, Correctness of Algorithms, Growth of Functions, Asymptotic Notations and Types, limit theorem. Stirling's Formula, **Recurrences:** Solution of recurrences by Substitution, Iteration, Masters method, Recursion Tree, and Change of variable method, Mathematical analysis for Recursive and Non-recursive algorithms, Amortized Analysis.

Unit II :Dynamic Programming **(9 Hours)**

Divide-and-conquer Paradigm- Binary search, Quick sort, Merge sort, Heap Sort, Priority Queue Dynamic programming- General concept, Elements of dynamic programming, Applications: Matrix-chain Multiplication, Longest Common Subsequence, 0/1 Knapsack problem, Assembly-line Scheduling.

Unit III : Greedy Algorithm and Graph Algorithms **(9 Hours)**

Greedy Algorithms-**G**eneral Concept, Elements of Greedy strategy, Applications: Activity Selection Problem, Fractional knapsack problem, Huffman codes. Graph Algorithms-A data structure for disjoint sets, Disjoint set operations, Disjoint set forests, Graph Algorithms and their characteristics, Breadth-first search and depth-first search, Minimum Spanning Trees (Kruskal's and Prim's algorithms), single-source shortest paths (Bellman-ford Algorithm and Dijkstra's), All pair Shortest Path (Floyd-Warshall's Algorithm).

Unit IV: String Matching and Backtracking **(9 Hours)**

String Matching Algorithms-Naive String-Matching Algorithm, The Rabin-Karp Algorithm, Knuth-Morris-Pratt Algorithm. Backtracking-N-Queen problem, Sub Set Sum Problem, Hamiltonian Circuit Problem, Graph Coloring Problem

Unit V : Branch and Bound Algorithms **(9 Hours)**

Branch and Bound- Assignment Problem, Travelling Salesman Problem. Complexity Classes-Problem Classification P, NP, NPC, NP-Hard; Reducibility, NP Completeness, NP-Complete & NP-Hard, 3SAT, Vertex Cover, Clique, Approximation algorithms (Travelling Salesman Problem, Vertex Cover).

Text Book:

3. Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, 4th Edition 2022, PHI Learning Pvt. Ltd, ISBN: 9780262046305
4. Fundamentals of Algorithm, Horowitz & Sahani:, 2nd Edition 2008,Universities Press, ISBN: 9788173716126

Reference Book:

4. Algorithms, Design and Analysis, H. Bhasin, First Edition 2015, Oxford University,ISBN: 9780199456666
5. Design and Analysis of Algorithm, S. Sridhar, Oxford University Press,1st Edition 2014, ISBN:9780198093695
6. Algorithms, Sanjay Dasgupta, UmeshVazirani , McGraw-Hill Education,1st Edition 2023, ISBN: 9789355325525

Course Code	Name of the Course	L	T	P	C
XXPC2006	Ethical Software Engineering	3	0	0	3

Course Educational Objectives

CEO1	**To impart a foundational understanding of the Software Development Life Cycle (SDLC), process models, and the fundamental professional codes of ethics (such as ACM/IEEE) that govern responsible software engineering practice.**
CEO2	**To equip students with the skills to effectively execute requirement engineering and structured system design (UML/DFD) while strictly adhering to intellectual property rights, data privacy laws, and compliance standards.**
CEO3	**To instill ethical technical practices during the implementation, testing, and maintenance phases of software development, prioritizing security-by-design, accessibility, and high-quality coding practices.**

Course Outcomes: Towards the end of the course students will be able to

CO1	**Explain** the Software Development Life Cycle (SDLC), various process models, and professional codes of ethics.
CO2	**Apply** requirement engineering techniques and **interpret** intellectual property, privacy, and ethical laws in software development.

CO3	**Design** software solutions using UML/DFD models and **demonstrate** ethical coding, testing, and maintenance practices.
CO4	**Analyze** ethical and societal challenges in emerging technologies (AI, IoT, Blockchain, Cloud) with focus on accountability, fairness, and sustainability.
CO5	**Evaluate** global/national IT laws, governance frameworks, and **assess** real-world ethical case studies for compliance and professional responsibility.
CO6	**Explain** the Software Development Life Cycle (SDLC), various process models, and professional codes of ethics.

CO-PO & PSO Mapping

Cos	PROGRAMMEOUTCOMES												PSOs	
	1	2	3	4	5	6	7	8	9	10	11		1	2
CO1	3	2	2		1	2		3		1	1			
CO2	2	3	2	1		3				1	1		1	
CO3	2	2	1	2		3	1			1				
CO4	2	3	1	2	1	3	2							
CO5	1	3		3		3					1		1	1
CO6	2			1					1				2	2
Avg.														

SYLLABUS

UNIT 1: Introduction to SDLC and Foundations of Software Engineering Ethics (12 Hours)

Introduction to SDLC and its Phases, & Process Models: Waterfall, Incremental, RAD, V-Model, Spiral, Agile & Scrum, COCOMO Cost Estimation Model, Introduction to Ethics: Morality, Laws, and Professional Codes, ACM/IEEE Software Engineering Code of Ethics, Responsibility of Software Engineers: Stakeholders, Public Interest, and Harm Mitigation, Case Studies: Therac-25, Boeing 737 Max, Ethical Decision-Making Models.

UNIT 2: Software Requirements Engineering and Ethics in Soft. Development Practices (12 Hours)

Requirements Types: Functional & Non-functional, User & System Requirements, Requirement Engineering Process: Elicitation, Feasibility, Validation, Management, Intellectual Property: Copyrights, Patents, Trade Secrets, Privacy and Data Protection Laws (GDPR, HIPAA, etc.), Ethical Testing and Debugging, Software Licensing, Open Source Ethics, Plagiarism, Misrepresentation, Whistleblowing.

UNIT:3 Ethical Design, Implementation, and Testing (10 Hours)

Design Concepts: UML, DFD, Transform & Transaction Analysis, Interface Design, Coding, Practices: Refactoring, Reviews, Stubs, Clean Code Ethics, Testing Techniques: White-box, Black-box, Regression, Validation, Debugging, SEI-CMM Characteristics of Software Maintenance, Reverse Engineering, Re-engineering & Reliability Metrics, Ethical Dimensions: Security-by-design, accessibility, test-skipping, bug suppression.

UNIT 4: AI Systems and Emerging Technology Ethics (8 Hours)

Ethical Issues in Blockchain, IoT, and Cloud Computing, Human-Centered Design and Accessibility, Automation and Job Displacement, Digital Divide and Global Software Ethics, Green Computing and Sustainability, Bias & Discrimination in AI, Transparency (XAI), Accountability, Ethical AI Use Cases: Surveillance, Hiring, Criminal Justice.

UNIT 5: Case Studies, Legal Frameworks, and Governance (8 Hours)

Global IT Laws and Cyber Laws, National Ethical Guidelines (India: MeitY, NITI Aayog AI Principles), Organizational Ethics Frameworks, Case Study Analysis: Real-world Ethical Failures and Successes, Ethics Audits and Compliance.

Text Book:

1. Rajiv Mall, Fundamentals of Software Engineering, 5th Edition, PHI Learning Pvt. Ltd, 2018, ISBN: 9789388028028
2. Roger S. Pressman, Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw-Hill, ISBN: 9789355325044

Reference Book :

1. Ethics in Information Technology – George Reynolds, Cengage,6th Edition 2019, ISBN: 9781337405874
2. IEEE/ACM Software Engineering Code of Ethics – https://ethics.acm.org/code-of-ethics/
3. EU AI Ethics Guidelines – https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Course Code	Name of the Course	L	T	P	C	
XXES2001	**Python Programming**	3	0	0	3	

Course Educational Objectives

CEO1	To provide a solid foundation in Python programming language, enabling students to write, debug, and execute programs using basic programming constructs such as variables, data types, control structures, functions, and modules.
CEO2	To develop problem-solving skills through algorithmic thinking, allowing students to design and implement Python solutions for real-world problems using structured programming techniques.
CEO3	To introduce essential programming paradigms and tools in Python, including file handling, exception handling, and the basics of object-oriented programming (OOP), thereby equipping students for further study and application development.

Course Outcomes: Towards the end of the course students will be able to

CO1	Understanding the basic features of Python programming, Syntax Rules, Use of Operators, and built-in methods.
CO2	Solving different real-world problems by implementing concepts of control structures, functions, and packages.
CO3	Implementing different programming examples using files and object-oriented programming concepts.
CO4	Analyze and learn the concepts of debugging and multi-threading using complex programs
CO5	Develop case studies by using database connectivity and GUI for interactive applications
CO6	Design projects for solving real-world problems using key features of Python.

CO-PO & PSO Mapping:

COs	PROGRAMME OUTCOMES												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3														
CO2	3	3	2												
CO3	2	2										1			
CO4	2				2										
CO5		3	3												
CO6		2	2												
Avg.	**2.50**	**2.50**	**2.33**		**2.00**							**1**			

SYLLABUS

UNIT 1: Introduction **(09 Hours)**

Python and its features, First Python Program, Identifiers, Reserved Words, Lines and Indentation, Multi-Line Statements, Quotation & Comments; Assigning Values to Variables, Multiple Assignment. Data Types & Operators-Data Type Conversion; Basic Operators: Arithmetic, Comparison, Assignment, Bitwise, Logical, Membership, Identity, Operators Precedence, Mathematical functions, Input and Output Statements. Strings and built-in methods, Lists and built-in methods, Tuples and built-in methods, Set, and built-in methods Dictionary, and built-in methods;

UNIT 2: Flow Control **(09 Hours)**

Conditional blocks using if, else, and elif, Simple for loops in Python, for loop using ranges, string, list, and dictionaries, Use of while loops in Python, Loop manipulation using pass, continue, break, and else. Python Functions, Modules, and Packages- Functions and their types, Types of arguments, Usage of the global variable, Recursive functions, Anonymous, Lambda, map, filter and reduce functions, Nested Functions, Generator and Decorator functions. Importing own and external modules, Understanding Packages: Creating and accessing packages.

UNIT 3: File Operation,OOPS **(09 Hours)**

Python File Operation-Programming using file operations, Manipulating file pointer using seek, Understanding read functions: read(), readline(), and readlines(), Understanding write functions: write() and writelines(), Handling CSV files, Zipping and Unzipping of files. Python Regular Expression-Powerful pattern matching, Power of pattern searching using regex in Python, Pattern finding programs using regular expression. Python Object Oriented Programming – OOPS- Concept of class, object, and instances, Methods and its types, Inner class, Constructor, and destructors, Inheritance, overloading, and overriding.

UNIT 4: Exception Handling and Multithreading **(09 Hours)**

Python Exception Handling- Error, Exception and its Hierarchy, Usage of try, except, finally and else, Types of exception, Programming using Exception handling. Python Multithreading- Understanding threads, sleep(), join() methods, Daemon Thread, Synchronizing the threads using lock, Programming using multithreading.

UNIT 5 : Database and GUI **(09 Hours)**

Python Database Interaction- SQL Database connection using Python, Creating and searching tables, Reading and storing on the database, Programming using database connections. Case study: Oracle and MySQL database. Python GUI Introduction- GUI application Using Tkinter, creating checkboxes, radio buttons, List Box, Buttons, menus and accessing files.

Text Books:

1. "Python: The Complete Reference", Martin C. Brown, McGraw-Hill/Osborne Media, Revised Edition 2022, ISBN: 9789387572942
2. "Learning Python", Mark Lutz, O'Reilly Media, Inc., Sixth Edition, 2025,ISBN: 9781098171261

Reference Books:

1. "Expert Python Programming", Tarek Ziad, Packt Publishing,4th Edition 2021,ISBN: 9781801076197
2. 2. "Python Essential Reference, David Beazley", Dveloper's Library, Sams Publishing,4th Edition 2009,ISBN: 9780672329784

Course Code	Name of the Course	L	T	P	C
XXPC2004	Operating System	3	0	0	3

Course Educational Objectives

CEO 1	To understand the fundamental concepts, architecture, and techniques of Operating Systems.
CEO 2	To learn about processes, CPU scheduling algorithms, and address issues related to Process Synchronization.
CEO 3	To comprehend Memory Management mechanisms, with a focus on Deadlock principles and related challenges.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Demonstrate the basic principles of operating systems, their functions, and various system architectures.
CO2	Demonstrate and solve different process scheduling algorithms used in operating system and measure their performance.
CO3	Analyze and solve various process synchronization techniques and different methods of deadlock.
CO4	Design memory management systems and implement techniques like paging, segmentation, and virtual memory.
CO5	Evaluate file systems and mass storage structures for efficient data management and retrieval.
CO6	Demonstrate different protection mechanisms to ensure the security and integrity of the operating system.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	1	0	0	0	0	0	0	0	0	0		
CO2	2	3	4	2	0	0	0	0	1	1	0	0		
CO3	3	3	4	3	0	0	0	0	0	0	0	0		
CO4	2	3	3	2	0	0	0	0	1	1	0	1		
CO5	2	4	3	1	0	0	0	0	1	0	0	0		
CO6	1	2	1	1	0	0	0	0	1	0	0	0		
Avg.	2.16	2.83	2.66	1.5	0	0	0	0	0.66	0.33	0	0.16		

UNIT I : Introduction to Operating System (9 Hours)

Operating Systems Objectives and functions, OS Structure, Evolution of Operating Systems: Batch Processing system, Multi programming system, Time sharing system, Multi Processing system, Distributed Systems, Clustered system, Real-Time Systems, Operating System Components, OS services, OS interfaces, System Calls, Types of System Calls, Operating System Design and Implementation.

UNIT II :Process concepts (9 Hours)

Process concepts - The Process, Process State, Process Control Block, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Pre-emptive Scheduling, Dispatcher, Scheduling Criteria. CPU Scheduling algorithms: First-Come, First-Served, Shortest Job First, Non-Preemptive Priority Scheduling , Shortest Remaining Time First, Round Robin, Priority Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Multiple-Processor Scheduling, Thread concept, Single and Multithreading Models.

UNIT: III : Process Synchronization (9 Hours)

Process Synchronization, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization: Producer-Consumer, Readers-Writers, The Dining Philosophers Problem, Monitors. Deadlocks - Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

UNIT IV : Memory Management (9 Hours)

Memory Management and Virtual Memory: Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Tables, Segmentation, Segmentation with Paging, Virtual Memory,

Demand Paging, Performance of Demanding Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

UNIT V : File System **(9 Hours)**

File System Interface – Files Concepts, Access methods, Directory Structure, File System Mounting, Sharing, and Protection. File Allocation methods, Free-space Management. Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, Look, Disk Management: Formatting, partitioning, raid structure. System Protection and principles, Goals of Protection, Case studies in Linux, Windows.

Text Books:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 10th Edition, Wiley, 2021, ISBN: 9781119800361
2. William Stallings, Operating Systems: Internals and Design Principles, 9th Edition, Pearson, 2026, ISBN: 9780134670959

Reference Books:

1. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 5th Edition, Pearson, 2024, ISBN:9781292459660
2. D. M. Dhamdhere, Operating Systems: A Concept-Based Approach, 3rd Edition, McGraw Hill Education, 2017, ISBN: 9781259005589
3. Brian L. Stuart, Principles of Operating Systems, 1st Edition, Lulu.com, 2021, ISBN: 9781300668558
4. 4. Ramez Elmasri, A. G. Carrick, David Levine, Operating Systems: A Spiral Approach, 1st Edition, McGraw Hill, 2009, ISBN: 9780072449815

Subject Code	Name of the Subject	L	T	P	C
HSH2002	Organizational Behaviour	2	0	0	2

Course Educational Objectives

CEO1	To develop a fundamental understanding of the concepts, evolution, scope, significance, and applications of Organizational Behaviour in contemporary organizations.
CEO2	To familiarize students with the influence of individual behaviour, including attitude, personality, perception, emotional intelligence, and motivation, on employee performance and organizational effectiveness.
CEO3	To enable students to understand group behaviour, team dynamics, leadership approaches, organizational culture, and change management practices for improving organizational performance.

Course Outcomes: Towards the end of the course, Students will be able to:

CO1	Explain the fundamental concepts, evolution, scope, significance, and applications of Organizational Behaviour in modern organizations.
CO2	Examine the influence of attitude, personality, and emotional intelligence on individual behaviour and workplace performance.
CO3	Analyse the role of perception and motivation theories in enhancing employee performance, satisfaction, and organizational effectiveness.
CO4	Demonstrate an understanding of group behaviour, team dynamics, and various leadership approaches for improving organizational performance.
CO5	Evaluate the impact of organizational culture and change management practices on organizational adaptability and effectiveness.
CO6	Apply Organizational Behaviour concepts and theories to analyse and propose solutions for real-world workplace and managerial issues in engineering and technology organizations.

CO-PO-Mapping

Course Outcomes (COs)	PROGRAMME OUTCOMES										PSO			
	1	2	3	4	5	6	7	8	9	10	11	1	2	3

CO1	2										2			
CO2		2				2	3	3	2					
CO3		3					2	2						
CO4		2				2	2	3	3	2				
CO5		2	2			3	2	2	2	3				
CO6		3	2	2		2	2	2	3	2	3			
Average	**0.33**	**2.00**	**0.67**	**0.33**		**1.50**	**1.83**	**2.0**	**1.67**	**1.17**	**0.83**	**1.23**		

SYLLABUS

Unit-I **(6 hours)**

Fundamentals of OB: Definition, scope and importance of OB, Evolution of OB Limitations of OB, Emotional Intelligence: Meaning, Components, Importance in Organizations, Daniel Goleman's Five Components of Emotional Intelligence.

Unit-II **(6 hours)**

Attitude: Components of attitude; Job attitude, Barriers to changing attitudes.
Personality: Definition and importance of Personality for performance, Trait theory, The Myers-Briggs Type Indicator and The Big Five personality model, personality and job – fit theory.

Unit III. **(8 hours)**

Perception: Meaning and concept of perception, Perceptual process social perception (stereotyping and halo effect).
Motivation: Definition & Concept of Motivation, The Content Theories of motivation (Maslow's Need Hierarchy & Herzberg's Two Factor Model Theory), The Process Theories (Vroom's expectancy, Equity Theory of Work Motivation)

Unit IV **(7 hours)**

Foundations of Group Behavior: The Meaning of Group & Group behavior & Group Dynamics, Types of Groups, The Five — Stage Model of Group Development.
Leadership: Concept of Leadership, Styles of Leadership, Trait Approach, Contingency Leadership Approach, Contemporary Leadership Approaches: Transformational, Transactional Leadership.

Unit V **(8 hours)**

Organizational Culture: Meaning & Definition of Organizational Culture, Culture, creating & Sustaining Organizational Culture, Types of Culture (Strong vs. Weak Culture & Formal vs. Informal Culture)
Organizational Change: Meaning, Definition & Nature of Organizational Change Implementing Organizational Change: Resistance to Change, how to overcome the Resistance to Change, Kurt Lewin's-Three step model, Kotter's Eight-Step plan for Implementing Change

Text Books:

1. Stephen P. Robbins & Timothy A. Judge, Organizational Behaviour, Global Edition, 20th Edition, 2026, ISBN: 9780135431771
   Stephen P. Robbins & Timothy A. Judge, Essentials of Organizational Behaviour, Global Edition, 16th Edition, 2026, ISBN: 9781292487625
2. Fred Luthans, Brett C. Luthans & Kyle W. Luthans, Organizational Behaviour: An Evidence-Based Approach, 13th Edition, Information Age Publishing, 2015, ISBN: 9781681231013

Reference Books:

1. **K. Aswathappa, Organisational Behaviour, 17th Edition, Himalaya Pblishing House,2023, ISBN: 9789355965981**
2. Uma Sekaran, Organizational Behaviour,17th Edition, Himalaya Publishing House, 2023, ISBN: 9789355965981
3. S.S. Khanka, Organizational Behaviour, S. Chand Publishing, 2020, ISBN: 9789352837878

Course Code	Name of the Course	L	T	P	Credits
XXES2101	Python Programming Lab	0	0	2	1

Course Educational Objectives	
CEO1	To provide a solid foundation in Python programming language, enabling students to write, debug, and execute programs using basic programming constructs such as variables, data types, control structures, functions, and modules.
CEO2	To develop problem-solving skills through algorithmic thinking, allowing students to design and implement Python solutions for real-world problems using structured programming techniques.
CEO3	To introduce essential programming paradigms and tools in Python, including file handling, exception handling, and the basics of object-oriented programming (OOP), thereby equipping students for further study and application development.
CEO4	To enable students to apply Python in practical domains, such as data processing, simple graphical interfaces, or introductory data science, thereby promoting the use of Python as a tool for interdisciplinary problem solving.

Course Outcomes: Towards the end of the course students will be able to	
CO1	Develop and execute basic Python programs using variables, data types, operators, and control flow statements.
CO2	Apply functions and modular programming techniques to design reusable and well-structured code in Python.
CO3	Manipulate strings, lists, tuples, dictionaries, and sets to solve computational problems effectively.
CO4	Implement file handling operations for reading from and writing to different file formats using Python.
CO5	Use exception handling and debugging techniques to build robust and error-free Python programs.
CO6	Apply object-oriented programming concepts such as classes, objects, inheritance, and polymorphism to build modular and maintainable applications.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	
CO1	3	2											
CO2	2	2	2										
CO3	2	3	3	1									
CO4	2	2	3	2	1				1				
CO5	2	3	2						1	1			
CO6	1	2											
	2.00	2.33	1.67	0.50	0.17				0.33	0.17			1.02

Experiment – 01

1. WAP to enter two integers from the keyboard and perform all the arithmetic operations on them.
2. WAP to enter three sides of a triangle. Find out the area of the triangle.
3. WAP to enter an integer number and determine its binary, octal, and hexadecimal equivalents.
4. WAP to enter a quadratic equation's coefficients and find the equivalent roots.
5. WAP to enter a temperature in Fahrenheit and convert it into Celsius.
6. WAP to convert distance in meters to Kilometers.
7. WAP to enter time in minutes and print it in hour and minute format.

8. WAP to enter two numbers and swap them using bitwise operators.

Experiment – 02

1. WAP to enter a percentage and display the gradation system.
2. WAP to enter a 5-digit number and print the digits present at odd locations.
3. WAP to enter a number and check whether it is a palindrome number or not.
4. WAP to enter a number and check whether the number is a perfect number or not.
5. WAP to enter a 3-digit number and print all the prime factors of the number.
6. WAP to enter a string and print it in reverse also print the number of vowels and consonants in it.

Experiment – 03

1.WAP to print twin prime numbers present between 1 and N.
2.WAP to find the factorial of a number (without using math module).
3.WAP to check whether a year is a leap year or not.
4.WAP to check whether the string is symmetrical or palindrome.
5.WAP to read a sentence and print all the even-length words present in it.
6.WAP to read a string and remove all duplicates from it.

Experiment – 04

1.WAP to print the second largest and second smallest element in a list of 10 integers without using the sort function.
2.WAP to create two lists first list containing 5 integers and second list containing 5 strings.print both lists one element from each list combined at a time.
3.WAP to create an integer list of 20 elements increase the odd valued elements by 5.

Experiment – 05

1.WAP to enter a set and copy the content of the set into a new set one element at a time.
2.WAP to create a function that prints the first 15 terms of the Fibonacci series without using recursion.
3.WAP to create a function that takes list as argument and returns the even values of the list. Print the new list with even values.
4.WAP to check if a value is present in the list or not, using lambda function.
5. WAP to print the intersection of 2 arrays using lambda function.

Experiment – 06

1.WAP to print a matrix along with summation of row elements and column elements after entering a 3*3 matrix
2.WAP to print the transpose of a matrix of n*n order
3.WAP to triple all numbers in a given list of integers. Use map()
4.WAP to add three given list using python map and lambda
5.WAP to convert all the characters into uppercase and lowercase and eliminate duplicate letters from a given sequence. Use map() function.
6.WAP to convert a given list of integer and a tuple of integer in a list of string using map() function.

Text Books:

1. "Python: The Complete Reference", Martin C. Brown, McGraw-Hill/Osborne Media, Revised Edition 2022, ISBN: 9789387572942
2. "Learning Python", Mark Lutz, O'Reilly Media, Inc., Sixth Edition, 2025,ISBN: 9781098171261

Reference Books:

1. *"Expert Python Programming", Tarek Ziad, Packt Publishing,4th Edition 2021,ISBN:* 9781801076197
2. *"Python Essential Reference, David Beazley", Dveloper's Library, Sams Publishing,4th Edition 2009,ISBN:* 9780672329784

SEMESTER V [THIRD YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	PCC	XXPC3001	Theory of Computation	3	0	0	3	10	30	60	100
2	PCC	XXPC3002	Introduction to Machine Learning	3	0	0	3	10	30	60	100
3	PCC	XXPC3003	Data Mining and Data Warehousing	3	0	0	3	10	30	60	100
4	PCC	XXPC3004	Introductory Cyber Security	3	0	0	3	10	30	60	100
5	PEC	XXPE301Y	Professional Elective-I Option 1:Cloud Computing Option 2: IOT	3	0	0	3	10	30	60	100
6	RM	RMOE3001	Research Methodology	2	0	0	2	10	30	60	100
7	PCC	XXPC3101	Introduction to Machine Learning Laboratory	0	0	2	1	60	-	40	100
8	PCC	XXPC3102	Data Mining and Data Warehousing Laboratory	0	0	2	1	60	-	40	100
9	PCC	XXPC3103	Professional Elective I LAB Option 1:Cloud Computing Option 2: IOT	0	0	2	1	60	-	40	100
10	EEC	XXEE3201	Summer Internship-I	-	-	-	1	60	-	40	100
			Total:	**17**	**0**	**6**	**21**	**300**	**180**	**520**	**1000**

Course Code	Name of the Course	L	T	P	C
XXPC3003	Data Mining and Data Warehousing	3	0	0	3

Prerequisite: Database Management Systems (XXPC2002)

Course Educational Objectives

CEO1	Provide a strong conceptual foundation in data mining principles, the knowledge discovery process, and data preprocessing techniques.
CEO2	Develop competence in data warehouse architecture, multidimensional modeling schemas, and OLAP technologies for decision support systems.
CEO3	Enable students to apply frequent pattern mining, classification, and clustering algorithms to solve real-world analytical and business intelligence problems.

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Explain the fundamental concepts of data mining, the KDD process, and data preprocessing techniques for preparing raw data.
CO2	Describe data warehouse architecture, multidimensional schemas, and OLAP operations used for building decision support systems.
CO3	Apply frequent pattern and association rule mining algorithms to discover correlations in large transactional datasets.
CO4	Implement classification and prediction techniques, including decision trees, Bayesian methods, and neural networks, for predictive analytics.
CO5	Apply partitioning, hierarchical, and density-based clustering algorithms to group data objects and evaluate cluster quality.
CO6	Analyze emerging trends in web mining, text mining, and advanced data mining applications to address real-world problems.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSO	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	1			1					1		3	1
CO2	3	2	1		2					1		3	1
CO3	3	2	2		2				1	1		3	1
CO4	3	2	2	2	3					1		2	3
CO5	2	3	2	2	3					1		2	3
CO6	2	3	3	3	3	1			1	2	1	3	3
Avg.													

SYLLABUS

UNIT 1: Introduction to Data Mining **(9 Hours)**

Data Mining Basics- Motivation, importance, and definition of data mining, Kinds of Data Structured, semi-structured, and unstructured data types, Data Mining Functionalities- Characterization, discrimination, association, classification, clustering, and outlier analysis, Classification of Data Mining Systems- Based on data types, knowledge representation, and mining techniques, Data Mining Task Primitives- Basic operations for specifying mining tasks, Integration with Databases/Warehouses- Architecture and system integration, Major Issues in Data Mining Efficiency, scalability, privacy, and data quality challenges.

Data Preprocessing: Data Cleaning- Handling missing values, noise, and inconsistencies, Data Integration and Transformation- Combining data sources and transforming formats, Dimensionality reduction- numerosity reduction, and data compression, Data Discretization & Concept Hierarchy- Converting

continuous data into categorical; building hierarchies, Knowledge Discovery in Databases (KDD)- End-to-end process: selection, pre-processing, transformation, data mining, and interpretation.

UNIT 2: Data Warehousing and OLAP Technology **(9 Hours)**

Data Warehouse Definition-- Concept and role in decision support systems, Operational DB vs Data Warehouse Key differences in purpose, design, and usage, Multidimensional Data Model Representation of data for analytical processing, Data Cubes OLAP cube structures for multidimensional analysis, Schema Designs Star, snowflake, and fact constellation schemas, Data Warehouse Architecture- Layers: ETL, storage, metadata, and access tools, Data Warehouse Implementation- Practical steps and challenges in deployment.

OLAP Technology: Introduction to OLAP- Online Analytical Processing for multidimensional queries, OLAP Operations Roll-up, drill-down, slice, dice, and pivot, Types of OLAP Servers- ROLAP, MOLAP, and HOLAP — differences and use cases, Data Warehouse Usage- Querying, reporting, and decision support, From Warehousing to Mining- Transition from storage and analysis to knowledge discovery.

UNIT 3: Mining Frequent Patterns and Association Rules **(9 Hours)**

Frequent Patterns & Association Rules- Basic concepts and motivation for discovering relationships in data, Market Basket Analysis Identifying co-occurrence of items in transactional datasets, Apriori Algorithm- Method for mining frequent item sets; generating association rules, Efficiency Improvements in Apriori- Techniques to optimize performance and reduce computation, FP-Growth Algorithm- Alternative to Apriori; mining frequent patterns without candidate generation., Multilevel & Multidimensional Rules Discovering rules across hierarchies and multiple dimensions, Quantitative Association Rules- Handling numeric attributes in rule mining, From Association to Correlation- Moving beyond co-occurrence to measure strength and significance of relationships

UNIT 4: Classification and Prediction **(9 Hours)**

Classification Basics- Fundamental concepts and issues in classification and prediction, Decision Tree Induction- Tree-based models for classification; splitting criteria and pruning, Bayesian Classification- Probabilistic approach; Naïve Bayes classifier, Rule-Based Classification- IF-THEN rules for interpretable models, Classification by Backpropagation-- Neural networks and learning via backpropagation, Support Vector Machines (SVM) -Margin-based classification; kernel methods, k-Nearest Neighbour (k-NN) Instance-based learning using similarity measures.

Prediction: Regression Models- Linear and nonlinear regression for prediction tasks, Accuracy and Error Measures- Metrics such as precision, recall, F1-score, RMSE, Evaluating Classifiers and Predictors- Cross-validation, confusion matrix, and performance evaluation.

UNIT 5: Cluster Analysis and Advanced Topics **(9 Hours)**

Cluster Analysis Basics- Types of data used in clustering; motivation and applications, Clustering Methods- Categorization of major approaches, Partitioning Methods- k-Means and k-Medoids algorithms, Hierarchical Methods Agglomerative and divisive clustering; BIRCH algorithm, Density-Based Methods -DBSCAN and related approaches, Grid-Based & Model-Based Methods- Clustering using grids and statistical models, Outlier Analysis- Identifying anomalies and rare patterns.

Advanced Mining Applications: Web Mining-- Web content mining, structure mining, and usage mining, Text Mining- Extracting knowledge from textual data, Applications and Trends- Emerging applications in business, science, and AI; current research directions.

Text Books:

1. Han, Jiawei; Kamber, Micheline; Pei, Jian. Data Mining: Concepts and Techniques. 4th Edition, Morgan Kaufmann (Elsevier), 2022, ISBN: 9780128117606
2. Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd Edition, Wiley, 2013, ISBN: 9781118530801

Reference Books:

1. Tan, Pang-Ning; Steinbach, Michael; Karpatne, Anuj; Kumar, Vipin. Introduction to Data Mining. 2nd Edition, Pearson, 2019, ISBN: 9780133128901
2. Inmon, W. H. Building the Data Warehouse. 4th Edition, Wiley, 2005, ISBN: 9780764599446
3. Ponniah, Paulraj. Data Warehousing Fundamentals for IT Professionals. 2nd Edition, Wiley, 2010, ISBN: 9780470462072
4. Aggarwal, Charu C. Data Mining: The Textbook. 1st Edition, Springer, 2015, ISBN: 9783319141411

Course Code	Name of the Course	L	T	P	C
XXPC2101	**Data Mining and Data Warehouse Laboratory**	**0**	**0**	**2**	**1**

Prerequisite: Database Management Systems Lab (XXPC2101), Data Mining and Data Warehousing Theory (XXPC3003)

Course Educational Objectives

CEO1	Provide students with practical knowledge of data preprocessing, integration, and multidimensional data modeling techniques.
CEO2	Develop the ability to design data warehouses, construct multidimensional data cubes, and implement OLAP operations for analytical applications.
CEO3	Enable students to implement and evaluate standard frequent pattern mining, classification, and clustering algorithms using state-of-the-art data analytics tools.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Apply various data cleaning, integration, transformation, and reduction techniques to raw datasets using open-source data mining tools.
CO2	Design a data warehouse schema (star, snowflake, fact constellation) and construct multidimensional data cubes
CO3	Perform multi-axis analytical queries using OLAP operations such as roll-up, drill-down, slice, dice, and pivot.
CO4	Implement and analyze association rule mining algorithms (Apriori and FP-Growth) on transactional datasets.
CO5	Deploy and evaluate the performance of classification algorithms including Decision Trees, Naïve Bayes, and Support Vector Machines.
CO6	Implement partitioning, hierarchical, and density-based clustering models to group unstructured datasets and assess cluster quality metrics.

CO-PO & PSO Mapping

	PROGRAMME OUTCOMES											PSOs	
COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	–	2	–	–	–	–	–	–	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	2
CO3	3	3	2	2	2	–	–	–	–	–	–	3	2
CO4	3	3	3	2	3	–	–	–	1	1	–	3	3
CO5	3	3	3	2	3	–	–	–	1	1	–	3	3
CO6	3	3	3	3	3	–	–	–	1	2	1	3	3
Average	**3.00**	**2.83**	**2.33**	**1.67**	**2.50**	**0.00**	**0.00**	**0.00**	**0.33**	**0.67**	**0.17**	**2.83**	**2.33**

List of Experiments

Experiment 1: Introduction to Data Mining Toolsets

Familiarization with open-source data mining environments and analytical platforms (e.g., Weka, Orange, R, or Python Pandas/Scikit-Learn). Exploration of GUI interfaces, loading various file formats (.arff, .csv), and understanding dataset attributes and metadata statistics.

Experiment 2: Data Preprocessing – Formatting and Cleaning

Implementation of data cleaning techniques on realistic noisy datasets. Exercises involve handling missing or null values via mean/median imputation, identifying and removing duplicate records, and handling inconsistent data formats to ensure downstream analytical robustness.

Experiment 3: Data Transformation and Reduction

Execution of data preprocessing operations including normalization (Min-Max scaling, Z-score standardization) and data discretization (binning methods, concept hierarchy generation). Performance of data reduction techniques such as attribute selection and dimensionality reduction.

Experiment 4: Multidimensional Data Warehouse Schema Modeling

Design and structural setup of a data warehouse using database platforms. Construction of a multidimensional dimensional model defining fact tables, dimension tables, and setting up Star, Snowflake, or Fact Constellation mapping architectures.

Experiment 5: Implementation of Data Cubes and OLAP Operations

Populating the schema and building multidimensional Data Cubes. Practical execution of core Online Analytical Processing (OLAP) workflows including Roll-up (aggregation), Drill-down (navigation to detail), Slice and Dice (sub-setting data blocks), and Pivot (re-orienting multi-axis views).

Experiment 6: Frequent Pattern Mining using Apriori Algorithm

Implementation of the classic Apriori frequent itemset generation algorithm on transactional market-basket datasets. Configuring support and confidence thresholds to discover, parse, and filter significant association rules.

Experiment 7: Advanced Association Mining using FP-Growth Algorithm

Execution of the Frequent Pattern Growth (FP-Growth) algorithm on highly dense transactional transactional datastores. Comparative efficiency analysis against the Apriori baseline regarding computational execution time and memory footprint bounds.

Experiment 8: Classification Modeling using Decision Trees

Building predictive classification models utilizing Decision Tree Induction algorithms (e.g., ID3, C4.5, or CART). Evaluation of structural models using cross-validation variants, split metrics, and confusion matrices.

Experiment 9: Advanced Classification and Performance Evaluation

Implementation of alternative statistical classification models including Bayesian Classifiers (Naïve Bayes) and Support Vector Machines (SVM). Evaluation and benchmarking using metrics like accuracy, error rates, precision, recall, and F1-score optimization scales.

Experiment 10: Cluster Analysis Modeling and Optimization

Implementation of unsupervised clustering algorithms to isolate data object patterns. Practical applications explore partitioning techniques (k-Means, k-Medoids), Hierarchical Clustering (Agglomerative/Divisive algorithms), and Density-Based structures (DBSCAN) accompanied by cluster quality assessments.

Course Code	Name of the Course	L	T	P	C
XXPE301Y	Internet of Things	3	0	0	3

Prerequisite: Computer Networks(XXPC2005)

Course Educational Objectives

CEO1	Provide a solid conceptual foundation in the architecture, characteristics, and fundamental building blocks of the Internet of Things (IoT).
CEO2	Develop engineering competence in selecting, configuring, and interfacing sensors, actuators, microcontrollers, and communication protocols for IoT solutions.
CEO3	Enable students to design, develop, and evaluate secure, end-to-end IoT applications integrating cloud computing, data analytics, and edge intelligence

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Explain the core concepts, multi-layered architectures, and enabling technologies of the Internet of Things.
CO2	Analyze and select appropriate smart sensors, actuators, and hardware platforms (Raspberry Pi, Arduino) for specific physical deployment contexts.
CO3	Evaluate and configure IoT link-layer, network-layer, and application-layer communication protocols (MQTT, CoAP, LoRaWAN, HTTP).
CO4	Design end-to-end IoT system pipelines spanning data acquisition, local ingestion, and remote ingestion interfaces.
CO5	Implement cloud-integrated IoT applications with real-time data streaming, visual analytics dashboards, and edge processing topologies
CO6	Analyze critical security profiles, device vulnerability landscapes, privacy concerns, and emerging deployment models in modern IoT networks.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSO	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	1			1					1		3	1
CO2	3	2	1		2					1		3	1
CO3	3	2	2		2				1	1		3	1
CO4	3	2	2	2	3					1		2	3
CO5	2	3	2	2	3					1		2	3
CO6	2	3	3	3	3	1			1	2	1	3	3
Avg.	3	1			1					1		3	1

SYLLABUS

UNIT 1: Introduction and Vision of IoT **(9 Hours)**

Foundations of IoT- Definition, vision, evolution, and paradigm shifts shaping IoT, IoT Architecture- Physical and logical designs: functional blocks, communication models, and APIs, Enabling Technologies Wireless sensor networks, cloud computing, big data analytics, communication protocols, and embedded systems, IoT

Deployment Models Levels of IoT and deployment templates for real-world applications, M2M vs Commonalities, differences, and technological transitions from machine-to-machine systems to IoT ecosyst

UNIT 2: Smart Sensing, Actuation, and Hardware Platforms (9 Ho

Smart Sensing Principles- Fundamentals of smart sensors: classification, characteristics, and interfa frameworks, Actuation and Control Types of actuators, mechanics, and device control methods, Core Hard Platforms -Arduino ecosystem, Raspberry Pi architecture, and ESP8266/ESP32 Wi Fi SoC modules, Hard Interfacing-- GPIO pin configurations and serial communication protocols (I2C, SPI, UART), Embe Programming- Programming fundamentals for IoT endpoint devices.

UNIT 3: IoT Networking and Communication Protocols (9 Ho

Network Infrastructure Protocols- IPv4/IPv6 addressing, 6LoWPAN for low power networks, and RPI routing in lossy environments, Data Link Standards- IEEE 802.15.4, ZigBee, Bluetooth Low Energy (BLE) LoRaWAN for short and long range communication, Application Layer Protocols Lightweight messaging service protocols: MQTT, CoAP, XMPP, AMQP, and RESTful web services..

UNIT 4: Cloud Integration, Ingestion, and Data Analytics (9 Ho

IoT Data Pipeline- End-to-end pipeline: ingestion mechanisms, gateways, data management, and sto architectures, **Cloud Platforms-** Integration with AWS IoT Core, Microsoft Azure IoT Hub, Google Cloud and open-source platforms like ThingsBoard, **Visual Analytics- Tools** Time-series databases (Influx dashboard orchestration (Grafana), and real-time stream processing, **Remote Device Management-** Monito control, and management of IoT devices through cloud-based services..

UNIT 5: IoT Security, Privacy, and Emerging Landscapes (9 Ho

IoT Vulnerability Landscape- Device vulnerabilities, network attacks, and firmware exploits, **Secu Frameworks** Identity management, authentication, end-to-end encryption, and TLS profiles, **Privacy Regulations**- Privacy challenges and compliance with global data protection regulations, **Emerging Tre** Industrial IoT (IIoT), edge and fog computing topologies, TinyML (machine learning at the edge), decentralized privacy-preserving IoT systems..

Text Books:

1. Bahga, Arshdeep; Madisetti, Vijay. Internet of Things: A Hands-On Approach. 1st Edition, VPT (Univers Press), 2014. ISBN-13: 9780996025515

2. Hanes, David; Salgueiro, Gonzalo; Grossetete, Patrick; Barton, Robert; Henry, Jerome. IoT Fundamental Networking Technologies, Protocols, and Use Cases for the Internet of Things. 1st Edition, Cisco Press, 20 ISBN-13: 9781587144516

Reference Books:

1. Kamal, Raj. Internet of Things: Architecture and Design Principles. 1st Edition, McGraw Hill Education, 2017. ISBN-13: 9789352605224

2. Serpanos, Dimitrios; Wolf, Marilyn. Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies. 1st Edition, Springer, 2018. ISBN-13: 9783319697147

3. Hersent, Olivier; Boswarthick, David; Elloumi, Omar. The Internet of Things: Key Applications and Protocols. 2nd Edition, Wiley, 2012. ISBN-13: 9781119994350

4. Pfister, Cuno. Getting Started with the Internet of Things. 1st Edition, O'Reilly Media, 2011. ISBN-13: 9781449393571

Text Books

1. Han, Jiawei; Kamber, Micheline; Pei, Jian. Data Mining: Concepts and Techniques. 4th Edition, Morgan Kaufmann (Elsevier), 2022. ISBN: 978-0128117606.

2. Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd Edition, Wiley, 2013. ISBN: 978-1118530801

Reference Books

1. Tan, Pang-Ning; Steinbach, Michael; Karpatne, Anuj; Kumar, Vipin. Introduction to Data Mining. 2nd Edition, Pearson, 2019. ISBN: 978-0133128901.

2. Aggarwal, Charu C. Data Mining: The Textbook. 1st Edition, Springer, 2015. ISBN: 978-3319141411.

3. Inmon, W. H. Building the Data Warehouse. 4th Edition, Wiley, 2005. ISBN: 978-0764599446.

4. Ponniah, Paulraj. Data Warehousing Fundamentals for IT Professionals. 2nd Edition, Wiley, 2010. ISBN: 978-0470462072.

Course Code	Name of the Course	L	T	P	C
XXPC3103	IOT Laboratory	0	0	2	1

Prerequisite: Computer Networks Lab (XXPC2105), Embedded Systems Laboratory, Internet of Things Theory (XXPE301Y)

Course Educational Objectives

CEO1	Provide students with hands-on knowledge of hypervisors, virtual machine provisioning, and private cloud deployment models.
CEO2	Develop the practical ability to orchestrate public cloud platforms, compute instances, networking topologies, and auto scaling structures.
CEO3	Enable students to configure distributed scale-out filesystems, managed NoSQL engines, containerized workflows, and serverless architectures

Course Outcomes: Towards the end of the course students will be able to:

CO1	Apply embedded systems programming constructs to interface analog and digital sensors with microcontroller boards.
CO2	Implement physical actuation controls and interpret signal variations across hardware general-purpose input/output (GPIO) pins.
CO3	Configure and establish reliable serial communication links using industrial UART, I2C, and SPI synchronous protocols.
CO4	Deploy and analyze local and network-layer routing topologies for resource-constrained IoT endpoints using 6LoWPAN and ZigBee topologies.
CO5	Implement high-throughput application telemetry pipelines over MQTT and RESTful architectural patterns to remote cloud hosts.
CO6	Design secure end-to-end IoT solutions utilizing visual analytics dashboards, real-time edge streaming, and cryptographic link tokens.

CO-PO & PSO Mapping

	PROGRAMME OUTCOMES											PSOs	
COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	–	2	–	–	–	–	–	–	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	2
CO3	3	3	2	2	2	–	–	–	–	–	–	3	2
CO4	3	3	3	2	3	–	–	–	1	1	–	3	3
CO5	3	3	3	2	3	–	–	–	1	1	–	3	3
CO6	3	3	3	3	3	–	–	–	1	2	1	3	3

Average	**3.00**	**2.83**	**2.33**	**1.67**	**2.50**	**0.00**	**0.00**	**0.00**	**0.33**	**0.67**	**0.17**	**2.83**	**2.33**

List of Experiments

Experiment 1: Physical Interfacing of Sensors and Microcontroller GPIOs

Introduction to development environments (Arduino IDE, Thonny) and hardware pin architectures (ESP32 / Raspberry Pi Pico). Writing embedded C/Python scripts to interface and capture environment metrics from discrete analog and digital sensors (DHT11/DHT22 temperature-humidity, LDR light sensors).

Experiment 2: Physical Actuation Control and Signal Variations

Programming microcontroller units to trigger and govern physical system feedback configurations. Interfacing and operating digital relay switches, LED indicators, and managing dynamic velocity/position controls of DC and Servo motors through Pulse Width Modulation (PWM) duty cycles.

Experiment 3: Synchronous and Asynchronous Serial Communication Interfacing

Configuring peripheral communication networks between microcontrollers and smart sensors. Practical execution and debugging of asynchronous serial pipelines (UART via USB/TTL chips) and multi-device synchronous buses (I2C for OLED displays, SPI for RFID RC522 card reader modules).

Experiment 4: Operating System Deployment on Edge Gateway Hardware

Flashing and booting isolated operating environments (Raspberry Pi OS Lite) onto microcomputer boards. Establishing secure cryptographic shell sessions (SSH via PuTTY), mapping headless system properties, and setting up network configurations via bash terminal control structures.

Experiment 5: Constrained Network Protocol Integration (6LoWPAN / ZigBee)

Deploying and verifying low-power link-layer mesh and routing topologies. Structuring direct peer-to-peer data pipes using IEEE 802.15.4 standard radio frequencies (XBee transceivers), evaluating routing metrics (RPL logs), and tracking transmission latency under active signal degradation constraints.

Experiment 6: Telemetry Ingestion using Publish-Subscribe (MQTT) Architecture

Setting up a centralized local network message dispatcher (Mosquitto MQTT Broker). Writing script pipelines on sensor nodes to parse telemetry feeds into structured JSON packets and execute persistent publications to specific network topics under different Quality of Service (QoS) levels.

Experiment 7: RESTful Architecture Interfacing and Web Service Endpoints

Developing an application server layer using lightweight web frameworks (Python Flask/FastAPI). Exposing resource paths, managing discrete HTTP transmission methods (GET data logs requests, POST configuration updates), and interfacing sensor node payloads onto relational or file datastores.

Experiment 8: End-to-End Cloud Ingestion Integration

Provisioning virtual machine resource endpoints inside public infrastructure networks (AWS IoT Core / ThingsBoard Cloud). Registering hardware tokens, establishing encrypted transport layers (MQTTS/HTTPS via port 8883), and executing streaming updates to remote cloud registries.

Experiment 9: Real-Time Stream Analytics and Dashboard Orchestration

Building time-series pipeline frameworks to process streaming data logs. Connecting cloud datastore pools with automated dashboard tools (Grafana / ThingsBoard UI), mapping charts, and configuring threshold-driven alerting hooks to issue real-time email notifications.

Experiment 10: IoT Security Hardening and TinyML Inference at the Edge

Implementing device authentication controls, password management, and end-to-end data encryption profiles. Deploying pre-trained lightweight deep learning network models onto edge

devices (TinyML via TensorFlow Lite for Microcontrollers) to perform real-time local classification inference on sensor signals without cloud dependencies.

Text Books

1. Bahga, Arshdeep; Madisetti, Vijay. Internet of Things: A Hands-On Approach. 1st Edition, VPT (Universities Press), 2014. ISBN-13: 978-0996025515.

2. Hanes, David; Salgueiro, Gonzalo; Grossetete, Patrick; Barton, Robert; Henry, Jerome. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. 1st Edition, Cisco Press, 2017. ISBN-13: 978-1587144516.

Reference Books

1. Kamal, Raj. Internet of Things: Architecture and Design Principles. 1st Edition, McGraw Hill Education, 2017. ISBN-13: 978-9352605224.

2. Serpanos, Dimitrios; Wolf, Marilyn. Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies. 1st Edition, Springer, 2018. ISBN-13: 978-3319697147.

3. Hersent, Olivier; Boswarthick, David; Elloumi, Omar. The Internet of Things: Key Applications and Protocols. 2nd Edition, Wiley, 2012. ISBN-13: 978-1119994350.

4. Pfister, Cuno. Getting Started with the Internet of Things. 1st Edition, O'Reilly Media, 2011. ISBN-13: 978-1449393571.

Course Code	Name of the Course	L	T	P	C
XXPC3001	Theory of Computation	3	0	0	3

Course Educational Objectives

CEO1	To introduce the fundamental concepts of formal languages, grammars, and automata theory.
CEO2	To develop the ability to design finite automata, regular expressions, and context-free grammars for given languages.
CEO3	To familiarize students with Turing machines as a formal model of computation.

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Explain the fundamental concepts of alphabets, strings, languages, and finite automata (DFA/NFA), and construct automata for given regular languages.
CO2	Apply regular expressions, the pumping lemma, and closure properties to design and analyze regular languages.
CO3	Design context-free grammars and pushdown automata, and simplify grammars into normal forms (CNF/GNF).
CO4	Design Turing machines for computational problems and relate them to the Church-Turing thesis and recursively enumerable languages.
CO5	Analyze the decidability of problems, including the halting problem, using reduction techniques.
CO6	Classify problems based on computational complexity (P, NP) and evaluate NP-completeness of standard problems.

CO-PO & PSO Mapping

COs	**PROGRAMME OUTCOMES**											**PSO**	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	1							1	1	1	
CO2	3	3	2	1						1	1	1	
CO3	2	3	2	2		1				1			1
CO4	2	3	3	2	1						1		1
CO5	2	2	3	2			1					1	1
CO6	1	2	3	1					1			1	
Avg.													

SYLLABUS

UNIT 1: Introduction to Finite Automata **(10 Hours)**

Alphabets, Strings, Languages- Basic building blocks of computation; formal definitions and examples, **Automata Theory**- Motivation, scope, and role in computer science, **Deterministic Finite Automaton (DFA)**- Definition, transition diagrams, and design techniques, **.Nondeterministic Finite Automaton (NFA)**-- Concept, examples, and expressive power, **Equivalence of DFA and NFA**- Proof of equal computational capability, **NFA with ε-transitions** -Role of epsilon moves and simplification, **Conversion of NFA to DFA**- Subset construction method, **Minimization of DFA**- State reduction techniques for efficiency, **Moore and Mealy Machines--** Definitions, differences, and equivalence, **Applications of Finite Automata** -Lexical analysis, text search, protocol design, and pattern recognition

UNIT 2: Regular Expressions and Regular Languages **(10 Hours)**

Regular Expressions (RE)- Syntax, properties, and practical use in pattern matching, Equivalence of RE and Finite Automata, Kleene's Theorem — showing RE and FA describe the same class of languages, Arden's Theorem- Solving equations involving regular expressions, Pumping Lemma- Technique to prove non-regularity of languages; applications, Closure Properties- Union, concatenation, star, intersection, complement, etc., Regular Grammars Right-linear and left-linear grammars; their role in defining regular languages, Equivalence of Regular Grammars and Finite Automata- Demonstrating mutual convertibility, Myhill-Nerode Theorem - Characterization of regular languages; basis for DFA minimization, Decision Properties- Decidability of emptiness, finiteness, membership, and equivalence..

UNIT 3: Context-Free Grammars and Pushdown Automata **(8 Hours)**

Context-Free Grammars (CFG)- Definition, derivation trees, and ambiguity, Simplification of CFG- Elimination of useless symbols, unit productions, and null productions., Normal Forms -Chomsky Normal Form (CNF) and Greibach Normal Form (GNF) — standardization for parsing and proofs, Pushdown Automata (PDA)- Definition, design, and role in recognizing context-free languages, Equivalence of PDA and CFG- Demonstrating that CFGs and PDAs describe the same class of languages, Deterministic PDA- Definition, limitations, and comparison with nondeterministic PDA, Pumping Lemma for CFL Tool for proving non-context-free languages, Closure Properties Union, concatenation, Kleene star, intersection with regular languages, etc.

UNIT 4: Turing Machines **(8 Hours)**

Turing Machine (TM)- Definition, formal model, and design for simple problems, Variants of TM - Multi-tape, multi-track, and nondeterministic Turing machines, Universal Turing Machine - Concept of a machine that can simulate any other TM, Church-Turing Thesis -Foundational principle of computability, Recursive and Recursively Enumerable Languages- Classification of languages based on decidability, Chomsky Hierarchy- Four levels of languages and automata: Regular, Context-Free, Context-Sensitive, and Recursively Enumerable..

UNIT 5: Decidability and Computational Complexity **(8 Hours)**

Halting Problem- Proof of undecidability; implications for program analysis, Reducibility and Rice's Theorem- Using problem reductions to prove undecidability; Rice's Theorem on properties of languages, Post Correspondence Problem (PCP)- Classic undecidable problem and its role in computability theory, Computational Complexity- Time and space complexity; asymptotic analysis, Complexity Classes P and NP- Polynomial-time solvable problems vs nondeterministic polynomial-time problems, NP-Completeness - Cook's Theorem; concept of NP-complete problems, Examples of NP-Complete Problems - SAT, Vertex Cover, Hamiltonian Cycle, and their significance.

Text Book:

1. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, Cengage Learning, 2013,

ISBN: 978-1133187790.

2. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education, 2006, ISBN: 978-0321455369.

Course Code	Name of the Course	L	T	P	C
XXPE301Y	Cloud Computing	3	0	0	3

Prerequisite: Operating Systems(XXPC2004), Computer Networks(XXPC2005), Database Management Systems (XXPC2002)

Course Educational Objectives

CEO1	Provide a comprehensive conceptual foundation in cloud computing models, architectures, and standard deployment modes.
CEO2	Develop engineering competence in virtualization technologies, resource management, and cloud infrastructure orchestration
CEO3	Enable students to design, deploy, and analyze scalable cloud applications and data storage solutions utilizing modern ecosystem tools.

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Explain the core concepts, evolutionary origins, and fundamental business/technical drivers of cloud computing architectures.
CO2	Analyze the technical inner workings, types, and operational mechanisms of hypervisors and virtualization ecosystems.
CO3	Evaluate SPI (SaaS, PaaS, IaaS) service delivery frameworks and deploy application services across diverse cloud deployment models.
CO4	Design and configure scalable cloud data storage models, distributed file systems, and non-relational database structures.
CO5	Implement robust mechanisms for resource provisioning, load balancing, service-level agreements (SLAs), and security controls.
CO6	Analyze advanced trends including serverless computing, edge/fog integration, containerization (Docker/Kubernetes), and federated architectures

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSO	
	3	1			1					1			3
CO1	3	2	1		2					1			3
CO2	3	2	2		2				1	1			3
CO3	3	2	2	2	3					1			2
CO4	2	3	2	2	3					1			2
CO5	2	3	3	3	3	1			1	2	1		3
CO6	3	1			1					1			3
Avg.													

SYLLABUS

UNIT 1: Introduction to Cloud Computing **(9 Hours)**

Foundations of Cloud Computing- Origins, historical evolution, and definition of cloud computing, Drivers and Benefits- Business and technical drivers, core characteristics, and advantages, Cloud Architecture- Reference model and SPI framework (SaaS, PaaS, IaaS), Deployment Models -Public, private, community, and hybrid cloud models, Challenges and Governance- Barriers, risks, and governance issues in cloud environments.

UNIT 2: Virtualization Infrastructure and Mechanisms **(9 Hours)**

Foundations of Virtualization- Definition, importance, and underlying principles of virtualization, Types of Virtualization- Full virtualization, para virtualization, and hardware assisted virtualization, Hypervisors and VMMs Architecture and types of hypervisors/virtual machine monitors (Type 1 and Type 2: KVM, Xen, VMware, Hyper V), Resource Virtualization- CPU, memory, and I/O virtualization for efficient resource utilization, VM Lifecycle Management- Virtual machine lifecycle, live migration, and resource structures

UNIT 3: Cloud Platforms and Resource Provisioning (9 Hours)

Prominent Cloud Platforms- Mechanisms and architectures of Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure, **Service Portfolios**- Core services: compute, networking, and infrastructure orchestration, **Resource Management**- Provisioning mechanisms, dynamic resource allocation, autoscaling frameworks, and workload management, **SLAs and Governance**- Service Level Agreements (SLAs) and their role in ensuring reliability and compliance.

UNIT 4: Distributed Storage and Data Models (9 Hours)

Cloud Storage Architectures- Data storage designs and architectures in cloud environments, **Distributed File Systems**- Google File System (GFS) and Hadoop Distributed File System (HDFS), **Cloud Storage Paradigms** - Object storage (Amazon S3) and block storage (Amazon EBS), **NoSQL Databases**- Key-value, document, columnar, and graph databases (BigTable, DynamoDB, Cassandra), **Data Replication and Consistency**- Replication strategies and consistency models in distributed systems.

UNIT 5: Cloud Security and Emerging Paradigms (9 Hours)

Cloud Security Architecture- Reference architecture and the shared responsibility model, **Infrastructure and Data Security**- Infrastructure protection, data security and encryption, identity and access management (IAM), **Compliance and Governance**- Legal regulations, compliance frameworks, and governance standards (GDPR, SOC2), **Emerging Paradigms**- Containerization (Docker, Kubernetes), serverless computing (AWS Lambda), edge and fog integration, and privacy-preserving federated cloud learning..

Text Books:

1. Buyya, Rajkumar; Vecchiola, Christian; Selvi, S. Thamarai. Mastering Cloud Computing: Foundations and Applications Programming. 2nd Edition, McGraw Hill Education, 2023. ISBN-13: 978-9355324641

2. Hwang, Kai; Fox, Geoffrey C.; Dongarra, Jack J. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. 2nd Edition, Morgan Kaufmann (Elsevier), 2024. ISBN-13: 978-0128218150

Reference Books:

1. Erl, Thomas; Mahmood, Zaigham; Puttini, Ricardo. Cloud Computing: Concepts, Technology & Architecture. 2nd Edition, Prentice Hall, 2023. ISBN-13: 9780138221089

2. Sosinsky, Barrie. Cloud Computing Bible. 1st Edition, Wiley, 2011. ISBN-13: 978-0470903568

3. Marinescu, Dan C. Cloud Computing: Theory and Practice. 3rd Edition, Morgan Kaufmann (Elsevier), 2022. ISBN-13: 978-0323919104

4. Velte, Anthony T.; Velte, Toby J.; Elsenpeter, Robert. Cloud Computing: A Practical Approach. 1st Edition, McGraw-Hill, 2009. ISBN-13: 978-0071626941

Course Code	Name of the Course	L	T	P	C
XXPC4007	Computer Graphics	3	0	0	3

Prerequisite:	**Discrete Mathematics (XXBS2001)**

Course Educational Objectives

CEO1	To provide students with a comprehensive understanding of the mathematical, algorithmic, and hardware foundations of computer graphics, including coordinate geometry, transformation theory, and the rendering pipeline.
CEO2	To develop students' ability to design and implement algorithms for modeling, rasterizing, shading, and rendering 2D/3D graphical objects using industry-standard graphics APIs and programming tools.
CEO3	To familiarize students with advanced rendering, visualization, and animation techniques, and to enable them to evaluate the performance, realism, and application suitability of modern computer graphics systems.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Analyze the fundamental concepts of raster and vector graphics, video display architectures, and the stages of the 2D/3D graphics rendering pipeline. (Analyze)
CO2	Apply 2D and 3D geometric transformations (translation, rotation, scaling, shearing) and homogeneous-coordinate matrix representations to manipulate graphical objects. (Apply)
CO3	Design and implement scan-conversion and clipping algorithms — including line, circle, and polygon rasterization, area filling, and window-to-viewport clipping. (Design)
CO4	Apply 3D viewing transformations, projections (parallel and perspective), visible-surface detection methods, illumination models, and shading techniques (flat, Gouraud, Phong) to render realistic scenes. (Apply)
CO5	Design curve and surface representations (Bezier, B-spline) and texture-mapping schemes for modelling and enhancing complex 3D objects. (Design)
CO6	Evaluate advanced rendering paradigms (ray tracing, radiosity), animation principles, and emerging applications of computer graphics in gaming, simulation, and visualization. (Evaluate)

CO-PO and CO-PSO Mapping Matrix

COs	Programme Outcomes (POs)											PSOs	
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	3	1	2	2	1	1	1	1	1	1	3	1
CO2	2	2	2	2	3	1	1	1	1	1	1	3	2

CO3	2	2	3	2	3	1	1	1	2	1	1	3	2
CO4	2	2	2	2	3	1	2	1	1	1	1	3	3
CO5	2	2	3	2	3	1	1	1	2	1	1	2	3
CO6	1	3	2	2	3	2	2	2	2	2	2	2	3
Average	**2.0**	**2.3**	**2.2**	**2.0**	**2.8**	**1.2**	**1.3**	**1.2**	**1.5**	**1.2**	**1.2**	**2.7**	**2.3**
Syllabus													

Unit I: Introduction to Computer Graphics (9 Hours)

Survey of computer graphics applications; overview of graphics hardware — raster-scan and random-scan (vector) display systems, CRT and flat-panel display technologies, frame buffers, and video controllers. Graphics software standards and the OpenGL architecture. Output primitives: scan-conversion (rasterization) algorithms for lines (DDA, Bresenham), circles and ellipses (midpoint methods), and polygon filling (scan-line, flood-fill, boundary-fill). Attributes of output primitives — line style, color, and area-fill attributes. Anti-aliasing techniques.

Unit II: 2D Transformations, Viewing, and Clipping (9 Hours)

Two-dimensional geometric transformations — translation, rotation, scaling, reflection, and shearing; homogeneous coordinates and composite transformations. Two-dimensional viewing pipeline — window-to-viewport mapping. Clipping algorithms: Cohen–Sutherland and Liang–Barsky line clipping, Sutherland–Hodgman polygon clipping, and text clipping. Structures and hierarchical modeling of 2D graphical scenes.

Unit III: 3D Transformations, Viewing, and Visible-Surface Detection (9 Hours)

Three-dimensional geometric transformations and 3D homogeneous-coordinate representations. 3D viewing pipeline — viewing coordinate systems, projections (parallel: orthographic and oblique; perspective), and view volumes. Visible-surface (hidden-surface) detection methods — back-face culling, depth-buffer (Z-buffer), painter's (depth-sort) algorithm, and area-subdivision methods. Introduction to 3D object representation schemes (boundary representation, octrees).

Unit IV: Illumination, Shading, Color, and Texture (9 Hours)

Light-source models and basic illumination models — ambient, diffuse (Lambertian), and specular (Phong) reflection. Polygon shading techniques — flat (constant) shading, Gouraud shading, and Phong shading. Color models — RGB, CMY, HSV, and YIQ, and their applications. Texture mapping and bump mapping fundamentals. Curve and surface representations — Bezier curves and surfaces, B-spline curves, and an introduction to NURBS.

Unit V: Advanced Rendering, Animation, and Applications (9 Hours)

Advanced rendering techniques — ray tracing (recursive ray tracing, ray–object intersection) and radiosity for global illumination. Fractals and procedural modeling for natural phenomena. Principles of computer animation — keyframing, interpolation, and motion representation. The modern GPU rendering pipeline and real-time rendering considerations. Emerging trends and applications — computer graphics in gaming, film/visual effects, scientific visualization, CAD, and its relationship to Virtual/Augmented Reality.

Text Books:

1. Donald Hearn, M. Pauline Baker, Warren R. Carithers, Computer Graphics with OpenGL, Pearson, 4th Edition, 2010, ISBN: 9780136053583.
2. **2.** Edward Angel, Dave Shreiner, Interactive Computer Graphics: A Top-Down Approach with WebGL, Pearson/Addison-Wesley, 7th Edition, 2014, ISBN: 9780133574845.

Reference Books:

1. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Morgan McGuire, David F. Sklar, Kurt Akeley, Computer Graphics: Principles and Practice, Addison-Wesley Professional, 3rd Edition, 2013, ISBN: 9780321399526.
2. Steve Marschner, Peter Shirley, Fundamentals of Computer Graphics, A K Peters/CRC Press, 4th Edition, 2015, ISBN: 9781482229394.
3. Matt Pharr, Wenzel Jakob, Greg Humphreys, Physically Based Rendering: From Theory to Implementation, MIT Press, 4th Edition, 2023, ISBN: 9780262048026.

Course Code	Name of the Course	L	T	P	C
XXPC3103	**Cloud Computing Laboratory**	**0**	**0**	**2**	**1**

Prerequisite: Operating Systems Lab (XXPC2104), Computer Networks Lab (XXPC2105), Cloud Computing Theory (XXPE301Y)

Course Educational Objectives

CEO1	Provide students with hands-on knowledge of hypervisors, virtual machine provisioning, and private cloud deployment models.
CEO2	Develop the practical ability to orchestrate public cloud platforms, compute instances, networking topologies, and auto scaling structures.
CEO3	Enable students to configure distributed scale-out filesystems, managed NoSQL engines, containerized workflows, and serverless architectures

Course Outcomes: Towards the end of the course students will be able to:

CO1	Deploy and configure Type-2 hypervisors to provision and control guest virtual operating environments.
CO2	Implement virtual machine instance migration architectures and build centralized network storage access blocks.
CO3	Orchestrate compute, networking, security policies, and programmatic access profiles on major public cloud architectures.
CO4	Design and benchmark distributed data clusters, scale-out file systems, and managed non-relational storage environments
CO5	Configure automated elastic scaling hooks, load balancer metrics, and global monitoring notification pipelines.
CO6	Deploy multi-tier containerized architectures using orchestration frameworks and develop microservice endpoints within serverless application runtimes

CO-PO & PSO Mapping

	PROGRAMME OUTCOMES											PSOs	
COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	–	2	–	–	–	–	–	–	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	2

CO3	3	3	2	2	2	–	–	–	–	–	–	3	2
CO4	3	3	3	2	3	–	–	–	1	1	–	3	3
CO5	3	3	3	2	3	–	–	–	1	1	–	3	3
CO6	3	3	3	3	3	–	–	–	1	2	1	3	3
Average	**3.00**	**2.83**	**2.33**	**1.67**	**2.50**	**0.00**	**0.00**	**0.00**	**0.33**	**0.67**	**0.17**	**2.83**	**2.33**

List of Experiments

Experiment 1: Hypervisor Deployment and Guest OS Virtualization

Installation, configuration, and structural setup of Type-2 Hypervisors (e.g., VirtualBox, VMware Workstation). Provisioning virtual machine instances, allocating virtual hardware parameters (vCPU, vRAM, Disk allocations), installing diverse guest operating systems, and setting up host-only and NAT network profiles.

Experiment 2: Private Cloud Storage and Virtual Machine Live Migration

Configuring shared network data storage systems (NFS/iSCSI target). Interfacing multi-hypervisor nodes onto shared pools, executing virtual instance snapshot trees, and conducting simulated live migrations across hosting fabrics while evaluating performance overheads and network latency metrics.

Experiment 3: Public Cloud Infrastructure Provisioning (IaaS)

Introduction to public cloud control layers (AWS Console/Azure Portal). Configuring secure Virtual Private Clouds (VPC/VNet), setting up subnets, defining security firewall rule lists, and provisioning Linux/Windows elastic compute instances (Amazon EC2 / Azure VM).

Experiment 4: IAM Security Management and Programmatic Access Profiles

Implementing Identity and Access Management (IAM) infrastructure. Creating custom administrative groups, assigning granular role permissions policies, enforcing multi-factor token setups, and setting up programmatic access secret keys to interact with infrastructure via Cloud Command Line Interfaces (CLI).

Experiment 5: Scalable Distributed File Systems Setup (HDFS)

Installation and deployment of a multi-node Hadoop Distributed File System (HDFS) cluster in a pseudo-distributed or fully distributed infrastructure configuration. Interfacing data partitions, testing automated file block replication mechanics, and running core shell management parameters.

Experiment 6: Data Modeling in Managed Cloud NoSQL Services

Design, structural modeling, and implementation of highly available data models within non-relational cloud systems (e.g., Amazon DynamoDB, Azure CosmosDB, or Apache Cassandra). Executing high-throughput read/write query maps, configuring partition keys, and evaluating eventual consistency latency bounds.

Experiment 7: Load Balancing and Multi-Tier Application Architecture

Deploying application servers behind public Application Load Balancers (ALBs). Implementing cross-zone request routing, designing health-check mechanisms, and verifying fault-tolerance profiles by inducing simulated backend system crashes.

Experiment 8: Elastic Autoscaling Framework Orchestration

Configuring metric-driven dynamic scale-out and scale-in autoscaling groups. Establishing infrastructure triggers linked to system performance parameters (e.g., CPU utilization thresholds, request queues), and testing real-world auto-scaling under synthetic stress testing toolnets (e.g., Apache Bench).

Experiment 9: Application Containerization and Microservices Orchestration

Developing custom microservice application builds and drafting declarative configurations (Dockerfiles). Building isolated images, executing local runtime targets via Docker Engine, and pushing production builds into managed container registry networks (Amazon ECR / Docker Hub).

Experiment 10: Serverless Function Implementation and Analytics Pipelines

Writing and deploying event-driven applications on serverless computing architectures (e.g., AWS Lambda, Azure Functions). Configuring cloud storage bucket triggers (Amazon S3 upload hooks), executing programmatic logs data transformations, and monitoring metrics with centralized analytic dashboards (CloudWatch).

Text Books

1. Buyya, Rajkumar; Vecchiola, Christian; Selvi, S. Thamarai. Mastering Cloud Computing: Foundations and Applications Programming. 2nd Edition, McGraw Hill Education, 2023. ISBN-13: 978-9355324641.

2. Hwang, Kai; Fox, Geoffrey C.; Dongarra, Jack J. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. 2nd Edition, Morgan Kaufmann (Elsevier), 2024. ISBN-13: 978-0128218150.

Reference Books

1. Erl, Thomas; Mahmood, Zaigham; Puttini, Ricardo. Cloud Computing: Concepts, Technology & Architecture. 2nd Edition, Prentice Hall, 2023. ISBN-13: 978-0138221089.

2. Marinescu, Dan C. Cloud Computing: Theory and Practice. 3rd Edition, Morgan Kaufmann (Elsevier), 2022. ISBN-13: 978-0323919104.

3. Sosinsky, Barrie. Cloud Computing Bible. 1st Edition, Wiley, 2011. ISBN-13: 978-0470903568.

4. Velte, Anthony T.; Velte, Toby J.; Elsenpeter, Robert. Cloud Computing: A Practical Approach. 1st Edition, McGraw-Hill, 2009. ISBN-13: 978-0071626941.

Course Code	Name of the Course	L	T	P	C
XXPC3101	**Introduction to Machine Learning Laboratory (IML Lab)**	**0**	**0**	**2**	**1**

Prerequisite: Introduction to Machine Learning (XXPC3002)

Course Educational Objectives

CEO1	To provide students with hands-on experience in implementing supervised learning algorithms for regression and classification tasks using Python-based machine learning libraries.
CEO2	To develop practical skills in constructing, tuning, and evaluating machine learning models, including decision trees, instance-based methods, and probabilistic classifiers.
CEO3	To enable students to implement unsupervised learning and ensemble learning techniques for analyzing and solving real-world, data-driven problems.

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Implement and evaluate simple and multivariate linear regression models using standard performance metrics.
CO2	Build logistic regression and classification models and assess them using confusion matrix based performance measures.
CO3	Construct and prune decision trees, and apply K-Nearest Neighbours and Naive Bayes classifiers to classification problems.
CO4	Apply unsupervised learning methods such as K-Means, K-Medoids, and Hierarchical Clustering to organize and interpret unlabeled data.
CO5	Develop ensemble learning models using Bagging, Boosting, and Random Forests to improve predictive performance.
CO6	Integrate various machine learning techniques into practical, end-to-end workflows to solve domain-specific problems.

CO-PO & PSO Mapping

COs	**PROGRAMME OUTCOMES**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	3	-	-	2	-	-	-	-	-	-	3	2
CO2	3	3	2	2	3	-	-	-	-	-	-	3	3
CO3	3	3	2	2	2	-	-	-	1	-	1	3	2
CO4	3	2	3	2	2	-	-	-	-	-	-	3	3
CO5	2	3	3	2	3	-	-	-	1	-	1	3	3
CO6	2	2	2	1	2	1	1	1	1	-	2	3	3
AVG	**2.7**	**2.7**	**2.0**	**1.5**	**2.3**	**0.2**	**0.2**	**0.2**	**0.5**	**-**	**0.7**	**3.0**	**2.7**

SYLLABUS

Experiment 1: Implement Simple and Multivariate Linear Regression from scratch and using Scikit-learn; evaluate using MSE, RMSE, MAE, R^2, and Adjusted R^2.

Experiment 2: Implement Logistic Regression for binary classification and study the effect of L1 (Lasso) and L2 (Ridge) Regularization.

Experiment 3: Evaluate classification models using Confusion Matrix, Accuracy, Precision, Recall, F1-score, and ROC Curve.

Experiment 4: Implement Decision Tree construction algorithms - ID3, C4.5, and CART - and visualize the resulting trees.

Experiment 5: Apply pre-pruning and post-pruning techniques to Decision Trees and compare their effect on model performance.

Experiment 6: Implement the K-Nearest Neighbours (KNN) classifier and study the effect of the value of K and distance metrics on accuracy.

Experiment 7: Implement a Naive Bayes Classifier (Gaussian/Multinomial) and apply it to a real-world classification problem.

Experiment 8: Detect multi-collinearity among features using correlation analysis and Variance Inflation Factor (VIF).

Experiment 9: Implement K-Means Clustering and determine the optimal number of clusters using the Elbow Method and Silhouette Score.

Experiment 10: Implement Clustering around Medoids (K-Medoids/PAM) and compare results with K-Means Clustering.

Experiment 11: Implement Agglomerative Hierarchical Clustering and visualize the results using Dendograms.

Experiment 12: Implement a K-d Tree for efficient nearest-neighbour search on multi-dimensional data.

Experiment 13: Implement Bagging and Random Forest classifiers and compare their performance with a single Decision Tree.

Experiment 14: Implement Boosting techniques - AdaBoost and Gradient Boosting - for classification problems.

Experiment 15: Case Study: Implement XGBoost for a multi-class classification problem and compare its performance with other ensemble methods.

Text Book:

1. Ethem Alpaydin, Introduction to Machine Learning, 4th Edition, MIT Press, 2020,

ISBN: 9780262043793

2. Stephen Marsland, Machine Learning: An Algorithmic Perspective, 2nd Edition, CRC Press, 2023, ISBN: 9781138583405

Reference Books:

1. Peter Flach, Machine Learning: The Art and Science of Algorithms That Make Sense of Data, 1st Edition, Cambridge University Press, 2012, ISBN: 9781107096394

2. Y.S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning from Data, 1st Edition, AML Book Publishers, 2012, ISBN:9781600490064.

Course Code	Name of the Course	L	T	P	C
XXPE302Y	Management Principles	3	0	0	3

Prerequisite:

Course Educational Objectives

CEO1	To provide students with a foundational understanding of classical and modern management thoughts, functions, and contemporary competencies required to lead in digital and global contexts.
CEO2	To develop students' ability to formulate strategic, tactical, and operational plans and apply analytical, data-driven, and AI-assisted decision-making frameworks to solve organizational problems.
CEO3	To familiarize students with modern agile organizational designs, strategic Human Resource Management (HRM), leadership theories, and advanced IT-enabled controlling systems to run efficient and compliant enterprises.

Course Outcomes: Towards the end of the course students will be able to:

CO1	**Explain** the evolution of management thoughts, the core functions of management, and adapt Mintzberg's managerial roles and essential skills to global and digital enterprises.
CO2	**Formulate** flexible, strategic, and tactical plans and apply data-driven, AI-assisted decision-making models to resolve complex organizational problems.
CO3	**Analyze** different organizational structures (functional, flat, hybrid) and examine modern, strategic HRM practices tailored for remote and hybrid work environments.
CO4	**Evaluate** classical and contemporary leadership theories and apply diverse motivational strategies to foster a culture of mentoring and innovation.
CO5	**Design** and implement effective control systems utilizing real-time monitoring, predictive analytics, budgetary frameworks, and quality standards (TQM, Six Sigma).
CO6	**Assess** contemporary management issues, cybersecurity/IT-enabled controls, and organizational governance practices to ensure ethical compliance and operational excellence.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
CO1	1	-	-	-	-	1	1	1	2	2	1	1	1
CO2	2	2	1	2	2	-	-	-	2	1	2	2	1
CO3	1	1	-	-	1	1	-	-	2	2	2	1	1
CO4	-	-	-	-	-	2	-	1	3	3	1	1	2
CO5	2	2	1	1	2	-	1	-	1	-	2	2	1
CO6	1	1	-	-	2	2	2	2	1	1	1	1	2
Average	**1.4**	**1.5**	**1.0**	**1.5**	**1.8**	**1.5**	**1.3**	**1.3**	**1.9**	**1.8**	**1.5**	**1.3**	**1.3**

SYLLABUS

Unit I: Introduction to Management **(9 Hours)**

Definition and Scope- Management as a process of achieving organizational goals through effective use of resources; relevance across industries including IT and digital enterprises, .Core Functions- Planning, Organizing, Leading, and Controlling — integrated with modern practices like agile management and data-driven decision-making, Managerial Roles- Interpersonal, informational, and decisional roles (Mintzberg framework), adapted to global and digital contexts, Essential Skills- Technical, human, and conceptual skills, plus contemporary competencies such as emotional intelligence, cross-cultural communication, and digital literacy, Evolution of Thought- From classical (Taylor, Fayol) and behavioural (Mayo, Maslow) approaches to modern perspectives including systems theory, contingency approach, and contemporary trends like lean management, design thinking, and sustainability-driven leadership

Unit II: Planning and Decision Making **(9 Hours)**

Planning Process- Systematic steps: setting objectives, analyzing environment, formulating strategies, implementing, and monitoring outcomes — aligned with agile and adaptive planning, Types of Plans- Strategic (long-term vision), Tactical (mid-level execution), and Operational (day-to-day activities), with emphasis on flexibility, scenario planning, and digital transformation strategies, Decision-Making Models -Programmed (routine, rule-based) vs. Non-programmed (novel, complex) decisions; inclusion of data-driven, AI-assisted, and evidence-based decision-making frameworks, Problem-Solving Techniques- Root cause analysis, brainstorming, decision trees, SWOT, and design thinking — enhanced by analytics, simulation, and collaborative tools

Unit III: Organization and HRM **(9 Hours)**

Organizational Principles- Core principles of clarity, efficiency, and adaptability; emphasis on agile structures and networked organizations for digital enterprises, Structures and Design- Functional, divisional, matrix, and flat structures; modern trends include hybrid, project-based, and virtual organizations, Delegation and Authority- Effective delegation, empowerment, and accountability; balancing centralization with decentralization in global and remote work contexts, Human Resource Management- Recruitment, selection, training, and performance appraisal — enhanced by digital HR tools, AI-driven talent acquisition, and continuous learning platforms, Strategic HRM- Aligning HR practices with organizational strategy; talent management, workforce analytics, diversity and inclusion, and employee engagement in the era of remote and hybrid work.

Unit IV: Leadership and Motivation **(9 Hours)**

Leadership Theories- Trait (innate qualities), Behavioural (observable actions), and Situational (context-driven adaptability). Modern extensions include transformational, transactional, and servant leadership, Leadership Styles and Skills- Autocratic, democratic, laissez-faire, and participative styles; contemporary emphasis on emotional intelligence, cross-cultural competence, and digital leadership in remote/hybrid teams, Motivation Theories- Maslow's hierarchy of needs, Herzberg's two-factor theory, McGregor's Theory X & Y, Role of Leaders as Mentors- Leaders as coaches fostering growth, guiding career development, and enabling innovation. Focus on mentoring, feedback culture, and employee empowerment.

Unit V: Controlling **(9 Hours)**

Control Process- Establishing standards, measuring performance, comparing results, and taking corrective action — integrated with real-time monitoring and predictive analytics, Types of Control- Pre-control (before activity), concurrent control (during activity), and post-control (after activity); modern emphasis on proactive and automated controls, Budgetary and Non-Budgetary Controls- Financial controls like budgets and cost analysis, alongside non-financial methods such as quality audits, performance metrics, and balanced scorecards, Effective Control Systems- Key characteristics: accuracy, timeliness, flexibility, and cost-effectiveness; enhanced by digital dashboards and KPI-driven frameworks, Contemporary Issues in Control- Quality Management (TQM, Six Sigma, ISO standards),Benchmarking: (Comparing with industry best practices), IT-Enabled Controls (ERP systems, AI-driven monitoring, cybersecurity, and compliance automation).

Text Books:

6. David S. Bright, Anastasia H. Cortes, Eva Hartmann, K. Praveen Parboteeah, Jon L. Pierce, Principles of Management, OpenStax, 1st edition, 2019, ISBN: 9789353168148
7. Stephen P. Robbins, David A. DeCenzo, Mary Coulter, Fundamentals of Management, Pearson Education, 10th edition, 2016, ISBN: 9780134237473

Reference Books:

1. Harold Koontz, Heinz Weihrich, Mark V. Cannice, Essentials of Management, McGraw Hill Education (India), 11th Edition, 2020, ISBN: 9780070620303
2. R.K. Sapru, Administrative Theories and Management Thought, PHI Learning, 3rd Revised Edition, 2013, ISBN: 9788120347342
3. D. Chandra Bose, Principles of Management and Administration, PHI Learning, 2nd Edition, 2012, ISBN: 9788120345812

4. Henry Mintzberg, The Nature of Managerial Work, Harper & Row, 1st Edition (Classic), 1980, ISBN: 9780060441979.

SEMESTER VI [THIRD YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	PCC	XXPC3004	Environmental Sustainability	3	0	0	3	10	30	60	100
2	PCC	XXPC3005	Generative AI Tools	3	0	0	3	10	30	60	100
3	PCC	XXPC3006	Advanced Database Managent System	3	0	0	3	10	30	60	100
4	PEC	XXPE302Y	Professional Elective-II Option-1: Big Data Analytics Option-2: Management Principle	3	0	0	3	10	30	60	100
5	PEC	XXPE303Y	Professional Elective-III Option-1: Compiler Design Option-2: Digital Marketing & E-Commervce Option-3: Computer Graphics	3	0	0	3	10	30	60	100
6	OEC	XXOE301Y	Open Elective - I	2	0	0	2	10	30	60	100
7	PCC	XXPC3104	Advanced Database Managent System Laboratory	0	0	2	1	60	-	40	100
8	PCC	XXPC3105	Generative AI Tools Laboratory	0	0	2	1	60	-	40	100
9	EEC	XXEE3202	Mini Project	0	0	8	4	-	-	100	100
			Total:	**17**	**0**	**12**	**23**	**180**	**180**	**540**	**900**

Course Code	Name of the Course	L	T	P	C
XXPC3006	Advance Database Management System	3	0	0	3

Prerequisite: Database Management System (XXPC2002)

Course Educational Objectives

CEO1	To provide advanced knowledge of database architectures, distributed databases, query processing, and transaction management for large-scale data systems.
CEO2	To develop the ability to design scalable, secure, and high-performance database solutions using modern database technologies including NoSQL, NewSQL, cloud, graph, and vector databases.
CEO3	To enable students to analyze emerging trends such as AI-enabled databases, data warehousing, big data storage, Blockchain databases, and autonomous database management systems.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Analyze advanced database architectures and storage models for large-scale applications
CO2	Design efficient query processing and optimization techniques for relational and distributed databases.
CO3	Apply distributed transaction processing, concurrency control, replication, and recovery techniques in distributed environments.
CO4	Evaluate modern database technologies including NoSQL, NewSQL, graph, cloud, and vector databases for different application domains.
CO5	Design secure and reliable database systems using authentication, authorization, encryption, auditing, and disaster recovery mechanisms.
CO6	Analyze emerging database technologies including AI-assisted databases, Blockchain databases, autonomous databases, and database support for machine learning applications..

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
CO1	3	2	1	–	–	–	–	–	–	–	–	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	2
CO3	3	3	3	2	1	–	–	–	–	–	–	3	2
CO4	3	3	2	2	1	–	–	–	–	–	–	3	2
CO5	2	2	1	–	2	–	–	–	–	–	–	2	3
CO6	2	2	1	–	3	–	–	–	–	–	–	2	3
Average	2.7	2.5	1.7	1.7	1.8	–	–	–	–	–	–	2.5	2.2

SYLLABUS

UNIT 1: Advanced Database Architecture: (9 Hours)

Review of Database System Architecture, Centralized and Client–Server Database Architecture, Parallel Database Architecture (Shared Memory, Shared Disk, Shared Nothing), Distributed Database Architecture, Database Storage Organization, File Organization, Storage Management, Buffer Management, Disk Scheduling, Database Partitioning, Horizontal and Vertical Partitioning, Sharding, Database Scalability, Indexing Techniques (B+ Tree, Bitmap Index, Hash Index), Storage Optimization Techniques

UNIT 2: Advanced Query Processing and Optimization (9 Hours)

Query Processing and Optimization: Query Processing Architecture, Query Parsing, Query Translation, Query Evaluation, Cost Estimation, Query Execution Plans, Heuristic Query Optimization, Cost-Based Query Optimization, Join Processing Algorithms (Nested Loop Join, Sort-Merge Join, Hash Join), Index Selection, Materialized Views, Query Rewriting, Distributed Query Processing, Query Performance Tuning, SQL Performance Optimization Techniques.

UNIT 3: Distributed Database Systems and Transaction Management (9 Hours)
Distributed Database Systems: Characteristics of Distributed Databases, Data Fragmentation (Horizontal, Vertical, Mixed), Data Allocation, Data Replication, Distributed Transactions, Distributed Concurrency Control, Distributed Deadlock Management, Two-Phase Commit Protocol (2PC), Three-Phase Commit Protocol (3PC), Distributed Recovery, Parallel Transaction Processing, CAP Theorem, BASE Properties, Fault Tolerance, High Availability Techniques.

UNIT4: Emerging Trends in Advanced Database (9 Hours)
In-Memory Databases, Time-Series Databases, Graph Database Applications, Multi-Model Databases, Self-Tuning Databases, Edge Databases, AI-Assisted Database Management, Database Support for Machine Learning Workloads, Federated Databases, Green Database Computing,

UNIT 5: Database Security, and Resilience (9 Hours)
Database Security: Authentication, Authorization, Role-Based Access Control (RBAC), Encryption Techniques, Database Auditing, Backup and Disaster Recovery, Secure Database Design. Emerging Trends: Blockchain Databases, Autonomous Databases, Database Support for Large Language Models (LLMs),

Text Books:

1. **M**. Tamer Özsu and Patrick Valduriez, Principles of Distributed Database Systems, 4th Edition, Springer, 2020. .ISBN: 978-3030262006
2. Pramod J. Sadalage and Martin Fowler, NoSQL Distilled*: A* Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2012, 1st Edition, ISBN: 978-0321826626

Reference Books:

1. Eric Redmond and Jim R. Wilson, Seven Databases in Seven Weeks, 2nd Edition, Pragmatic Bookshelf, 2018,ISBN: 978-1680502534
2. Bill Inmon, Building the Data Warehouse, Fifth Edition, Wiley, 2016. ISBN: 9781118530801

Course Code	Name of the Course	L	T	P	C
XXPC3104	**Advance Database Management System Lab**	**0**	**0**	**2**	**1**

Prerequisite: Advanced Database Management System (**XXPC3006)**

Course Educational Objectives

CEO1	Provide hands-on experience in designing, implementing, and optimizing advanced database systems, including distributed, NoSQL, and cloud-hosted databases.
CEO2	Develop practical proficiency in query optimization, transaction management, indexing, and performance tuning for large-scale, real-world database applications.
CEO3	Enable students to implement and evaluate modern database technologies, including graph, vector, and AI-integrated databases, along with database security and recovery mechanisms.

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Apply advanced indexing, partitioning, and storage organization techniques to optimize retrieval performance in large relational databases.
CO2	Design efficient query execution strategies using cost-based optimization, materialized views, and query rewriting for complex SQL workloads.
CO3	Analyze distributed transaction management by implementing concurrency control, two-phase commit, and replication protocols in a simulated distributed database environment.
CO4	Evaluate and implement NoSQL, NewSQL, and graph database solutions for handling unstructured, semi-structured, and highly connected data.
CO5	Design secure and resilient database systems incorporating authentication, role-based access control, encryption, auditing, and backup/disaster-recovery procedures.
CO6	Analyze and implement emerging database technologies, including vector databases, in-memory databases, and AI-assisted/autonomous database management tools, for practical application scenarios.

CO-PO & PSO Mapping

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	1	3	–	1	1	1	–	–	3	2
CO2	3	3	2	2	3	–	1	1	1	–	–	3	2
CO3	3	3	3	2	3	–	1	1	2	–	1	3	3
CO4	2	2	3	2	3	–	1	1	1	–	–	3	3
CO5	2	2	2	1	3	1	2	3	1	–	–	2	3
CO6	3	2	3	2	3	1	2	2	1	–	1	3	3
Average	**2.67**	**2.33**	**2.50**	**1.67**	**3.00**	**0.33**	**1.33**	**1.50**	**1.17**	**0.00**	**0.33**	**2.83**	**2.67**

List of Experiments

Experiment 1: Advanced Indexing and Storage Organization

Design and implement B+ Tree, bitmap, and hash indexes on large relational tables in PostgreSQL/MySQL. Compare query performance before and after indexing, and implement horizontal and vertical partitioning/sharding on a sample dataset, measuring the impact on storage and retrieval efficiency.

Experiment 2: Query Execution Plans and Cost-Based Optimization

Generate and interpret query execution plans (EXPLAIN/EXPLAIN ANALYZE) for complex multi-table SQL queries. Implement and compare join algorithms (nested loop, sort-merge, hash join), and apply query rewriting and materialized views to optimize execution cost.

Experiment 3: Materialized Views and SQL Performance Tuning

Create and refresh materialized views for analytical workloads. Profile slow queries using database performance tools, apply index selection strategies, and benchmark performance improvements on large synthetic datasets.

Experiment 4: Distributed Database Fragmentation and Replication

Set up a simulated distributed database environment. Implement horizontal, vertical, and mixed data fragmentation, configure data allocation and replication across nodes, and evaluate read/write consistency trade-offs.

Experiment 5: Distributed Concurrency Control and Commit Protocols

Implement distributed concurrency control mechanisms (locking, timestamp ordering) and simulate the Two-Phase Commit (2PC) and Three-Phase Commit (3PC) protocols across distributed nodes, analyzing failure handling and recovery.

Experiment 6: NoSQL Document and Key-Value Store Implementation

Design and implement a data model in a document-oriented database (e.g., MongoDB). Perform CRUD operations, aggregation pipelines, and indexing, and compare schema flexibility against a relational equivalent.

Experiment 7: Wide-Column and Graph Database Applications

Configure a wide-column store (e.g., Apache Cassandra) evaluating partition keys and eventual consistency. Model and query connected data using a graph database (e.g., Neo4j) with Cypher, applied to a real-world use case such as social network or recommendation analysis.

Experiment 8: Database Security and Access Control

Implement authentication and Role-Based Access Control (RBAC) on a multi-user database. Configure column/row-level encryption, enable database auditing, and design and execute a backup and disaster-recovery plan.

Experiment 9: In-Memory and Vector Database Implementation

Configure an in-memory data store (e.g., Redis) for caching and session management, benchmarking latency against disk-based storage. Implement a vector database or vector-search extension (e.g., pgvector) for similarity search over embeddings.

Experiment 10: Mini Project — AI-Assisted / Autonomous Database Application

Design and deploy an end-to-end mini project integrating a relational or NoSQL backend with an AI-assisted feature (e.g., natural-language-to-SQL query interface, automated index/query tuning, or a retrieval-augmented application using a vector store), and present a performance and design evaluation report.

Text Books

1. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, Database System Concepts, 7th Edition, McGraw-Hill Education, 2019. ISBN: 978-9352902117.

2. Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, 1st Edition, O'Reilly Media, 2017. ISBN: 978-1449373320.

Reference Books

1. Eric Redmond and Jim R. Wilson, Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement, 2nd Edition, Pragmatic Bookshelf, 2018. ISBN: 978-1680502534.

2. Ian Robinson, Jim Webber, and Emil Eifrem, Graph Databases: New Opportunities for Connected Data, 2nd Edition, O'Reilly Media, 2015. ISBN: 978-1491930892.

3. Jeff Carpenter and Eben Hewitt, Cassandra: The Definitive Guide, 3rd Edition, O'Reilly Media, 2020. ISBN: 978-1492097143.

Course Code	Name of the Course	L	T	P	C
XXPE302Y	**Big Data Analytics**	**3**	**0**	**0**	**3**

Prerequisite: Introduction to Machine Learning (XXPC3002)

Course Educational Objectives

CEO1	Understand Big Data concepts, architecture, and analytics.
CEO2	Analyze distributed frameworks, Hadoop ecosystem, and NoSQL data management.
CEO3	Apply analytical and machine learning techniques on Big Data.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Understand Big Data concepts, characteristics, and applications.
CO2	Explain scalability, parallel computing, and statistical basics.
CO3	Understand Hadoop architecture and HDFS processing.
CO4	Analyze Big Data using NoSQL and distributed storage.
CO5	Apply MapReduce, Hive, Pig, and Spark for analytics.
CO6	Interpret machine learning and advanced analytics (stream, graph, text).

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	1									2	1
CO2	3	2		1								2	1
CO3	2	3	1	1								3	2
CO4	2	3	1	1								3	2
CO5	2	3	2	1								3	3
CO6	2	3	1	2								3	3
AVG	**2.33**	**2.67**	**1.2**	**1.2**								**2.67**	**2**

SYLLABUS

Unit 1: Introduction to Big Data Analytics (8 Hours)

Introduction to Big Data- Definition, scope, and importance, Big Data Analytics- Techniques, goals, and value creation, Evolution of Big Data: From traditional data management to modern analytics, Need for Big Data- Drivers such as volume, velocity, variety, and value, Big Data Characteristics: 5Vs (Volume, Velocity, Variety, Veracity, Value), Challenges in Big Data- Scalability, integration, privacy, and security, Big Data Types-Structured, semi-structured, unstructured, Parallel Computing Paradigms- Cloud, grid, and cluster computing, Scalability- Horizontal vs vertical scaling, Big Data Architecture: Layers, components, and workflows, Data Sources: Sensors, social media, enterprise systems, IoT, Data Quality: Accuracy, consistency, completeness, Pre-processing Techniques: Cleaning, normalization, transformation, Storage Mechanisms: Distributed file systems, NoSQL databases, data lakes, Applications of Big Data- Healthcare, finance, retail, manufacturing, smart cities, Analysis vs Reporting, Sampling, distributions, re-sampling, inference, Prediction Error- Bias, variance, overfitting

Unit II: Hadoop and Distributed Data Processing (10 Hours)

Introduction to Hadoop-Motivation, design philosophy, and use cases, **Apache Hadoop**-Open-source framework and ecosystem overview, **Hadoop Architecture** -Master-slave design, NameNode, DataNode, JobTracker, TaskTracker, Data Analysis Approaches, **Analyzing Data with Unix Tools**-Traditional

command-line utilities, **Analyzing Data with Hadoop**- MapReduce paradigm and workflows, **Hadoop Streaming**-Using non-Java languages for MapReduce, **Hadoop Ecosystem-** Hive, Pig, HBase, Spark, Mahout, Zookeeper, Oozie, **IBM Big Data Strategy-** Enterprise adoption and integration, **HDFS Design**-Distributed storage principles, **HDFS Concepts**- Blocks, replication, fault tolerance, **Command Line Interface**-Basic operations and commands, **Hadoop File System Interfaces**- APIs for data access, **Data Flow in Hadoop**- Input, shuffle, output stages, **Data Ingest with Flume**- Streaming log data collection, **Data Ingest with Sqoop-** Import/export between Hadoop and RDBMS, **Hadoop Archives**- Efficient storage and retrieval, **Compression**- Techniques for reducing storage and transfer costs, **Serialization**-Writable interface, custom serialization, **Avro**- Data serialization system for schema evolution, **File-Based Data Structures**- Sequence files, MapFiles, Parquet, **Emerging Alternatives**- Spark, Flink, and cloud-native solutions

Unit III: NoSQL and Big Data Management (8 Hours)

Introduction to NoSQL Databases- Motivation, differences from RDBMS, advantages, **Schema-less Design**-Flexibility, dynamic schema evolution, **Flexible Data Models**-Adaptability for diverse data types, **Key-Value Databases-** Simple lookups, caching, scalability, **Document Databases**- JSON-like storage, hierarchical data (e.g., MongoDB), **Column-Family Databases**- Wide-column stores for analytics (e.g., Cassandra, HBase), **Graph Databases**- Relationship-centric queries (e.g., Neo4j), **NoSQL Architecture Patterns**- Sharding, replication, CAP theorem trade-offs, **Distributed Storage**- Horizontal scaling, fault tolerance, **MongoDB**- Document-oriented storage, aggregation framework, **Cassandra**- High availability, decentralized architecture, **Handling Big Data with NoSQL-** Use cases, performance considerations, **Emerging Trends**-Multi-model databases, cloud-native NoSQL, serverless data management

Unit IV: Big Data Processing (10 Hours)

Map Tasks- Input splitting, mapping functions, **Reduce Tasks**- Aggregation and summarization, **Job Scheduling**-Resource allocation, task prioritization, **Shuffle and Sort**- Data reorganization between map and reduce phases, **Partitioning**- Key-based distribution of data, **Combiner**-Local aggregation to reduce network overhead, **Task Execution Steps**- End-to-end workflow, **Hive Architecture**- Components, metastore, execution engine, **HiveQL**-SQL-like query language for Hadoop, **Hive Data Types**- Primitive and complex types, **Hive File Formats**-Text, ORC, Parquet, **Hive vs RDBMS**: Differences in schema, query execution, scalability, **Pig Latin**- High-level scripting language for data flows, **Pig Data Model**- Atoms, tuples, bags, maps, **Pig Scripting**: Procedural data manipulation, **Pig Data Processing Operations**: Load, filter, group, join, store, **Spark Architecture**- Driver, executors, cluster manager, **RDDs**- Resilient Distributed Datasets, transformations, actions, **Spark SQL-**Structured queries, integration with Hive, **DataFrames**- Schema-based distributed collections, **ETL with Spark**: Extract, transform, load pipelines, **Data Analysis Operations**-Aggregations, joins, streaming analytics, **Spark MLlib**- Machine learning algorithms and pipelines, **Data Visualization**: Integration with BI tools, dashboards, **Reporting Techniques**- Real-time and batch reporting, **Security in Big Data Processing**-Authentication, encryption, role-based access

Unit V: Machine Learning & Advanced Big Data Analytics (10 Hours)

Machine Learning in Big Data- Enhancing predictive analytics, pattern recognition, and automation, **Integration with Big Data**- Leveraging distributed frameworks for scalable ML, **Regression**- Linear, logistic, and advanced regression models, **Classification**- Decision trees, random forests, SVMs, neural networks, **Clustering Methods**-K-Means, hierarchical clustering, **Association Rule Mining**- Apriori algorithm for market basket analysis, **Spark Streaming**- Processing live data streams, window operations, fault tolerance, **Graph Models**- Nodes, edges, properties, **Social Network Analysis**- Community detection, centrality measures, influence modeling, **Text Analytics**- Sentiment analysis, topic modeling, NLP techniques, **Web Analytics**: Clickstream analysis, user behavior tracking, **Social Media Analytics**- Trend detection, opinion mining, influencer analysis, **Ethics in Big Data Analytics**- Bias, fairness, transparency, responsible AI

Text Books:

1. Raj Kamal and Preeti Saxena, Big Data Analytics: Introduction to Hadoop Spark and Machine Learning, 1st Edition, McGraw Hill Education India, 2022, ISBN: 9353164974
2. Seema Acharya & Subhashini Chellappan, Big Data and Analytics, 2nd Edition, Wiley India, 2020, ISBN: 9788126588367

Reference Books:

1. Viktor Mayer-Schönberger & Kenneth Cukier, Big Data: A Revolution That Will Transform How We Live, Work, and Think, 1st Edition, 2014, Harper Business, ISBN: 9780544227750
2. Tom White, Hadoop: The Definitive Guide, 4th Edition, O'Reilly Media, 2015, ISBN: 1491901632

Course Code	Name of the Course	L	T	P	C
XXPE303Y	**Compiler Design**	**3**	**0**	**0**	**3**

Course Educational Objectives

CEO1	To provide students with a foundational understanding of the structure, phases, and role of compilers in translating high-level programming languages into machine code.
CEO2	To develop students' ability to design lexical analyzers, parsers, and semantic analyzers using formal language theory and automata concepts.
CEO3	To familiarize students with intermediate code generation, optimization techniques, and runtime environments for efficient program execution.

Course Outcomes: Towards the end of the course students will be able to

CO1	Explain the role of compilers, phases of compilation, and the importance of lexical, syntax, and semantic analysis in program translation.
CO2	Design and implement lexical analyzers and parsers using regular expressions, finite automata, and context-free grammars.
CO3	Apply semantic analysis techniques including symbol table management, type checking, and scope resolution.
CO4	Generate intermediate code representations and apply optimization techniques for efficient execution.
CO5	Implement code generation strategies for target machines, including register allocation and instruction scheduling.
CO6	Evaluate advanced compiler issues such as error recovery, runtime environments, and the use of modern compiler construction tools (e.g., Lex, Yacc, LLVM).

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
CO1	1	-	-	-	-	1	1	1	2	2	1	1	1
CO2	2	2	1	2	2	-	-	-	2	1	2	2	1
CO3	1	1	-	-	1	1	-	-	2	2	2	1	1
CO4	-	-	-	-	-	2	-	1	3	3	1	1	2
CO5	2	2	1	1	2	-	1	-	1	-	2	2	1
CO6	1	1	-	-	2	2	2	2	1	1	1	1	2
Average	**1.4**	**1.5**	**1.0**	**1.5**	**1.8**	**1.5**	**1.3**	**1.3**	**1.9**	**1.8**	**1.5**	**1.3**	**1.3**

SYLLABUS

Unit I: Introduction and Lexical Analysis (9 Hours)

Compiler vs. Interpreter -Differences in translation, execution speed, and error handling; relevance in modern languages (e.g., Java's hybrid model, Python's interpreter), **Phases of Compilation**- Front-end (lexical, syntax, semantic analysis) and back-end (optimization, code generation); integration with modern compiler frameworks like LLVM and GCC, **Lexical Analysis**- Role of the lexer in tokenizing source code; handling identifiers, keywords, literals, and operators, **Regular Expressions and Automata**- Use of regex for token patterns; finite automata (NFA/DFA) for efficient lexical recognition, **-** Implementation using Lex/Flex; automation of token recognition and error detection.

Unit II: Syntax (9 Hours)

Analysis **Context-Free Grammars**- Foundation for defining programming language syntax; parse trees used to represent hierarchical structure, **Ambiguity in Grammars**- Issues caused by multiple parse trees; techniques for grammar refinement and disambiguation., **Top-Down Parsing**- Recursive descent and LL(1) parsing; predictive parsing with lookahead for efficiency, **Bottom-Up Parsing** -Shift-reduce parsing; LR(0), SLR, and LALR parsers widely used in modern compiler construction, **Error Recovery Techniques**- Panic-mode recovery, phrase-level recovery, and error productions; modern parser generators (Yacc/Bison) integrate robust error handling.

Unit III: Semantic Analysis and Syntax-Directed Translation (9 Hours)

Syntax-Directed Definitions-- Formal rules linking grammar with semantic actions; translation schemes for generating intermediate code, **Type Checking and Type Systems**-- Static vs. dynamic type checking; strong vs. weak typing; handling polymorphism and modern language features, **Symbol Table Management**-- Efficient storage and retrieval of identifiers; hash tables, linked structures, and scope resolution, **Scope Handling**- Block structures, nested scopes, lexical vs. dynamic scoping; relevance in modern languages like Java, Python, and C++, **Storage Organization and Activation Records**- Memory layout for variables, procedures, and recursion; activation records, stack management, and parameter passing.

Unit IV: Intermediate Code Generation (9 Hours)

Abstract Syntax Trees (AST)- Structured representation of source code; basis for semantic checks and intermediate code generation, **Directed Acyclic Graphs (DAGs)** -Optimization of expressions by eliminating redundant computations; useful in common subexpression elimination, **Three-Address Code**- Representation formats: quadruples, triples, indirect triples; widely used for intermediate representation in compilers, **Translation of Expressions and Control Flow**- Generating code for arithmetic, Boolean expressions, loops, and conditional statements; handling procedure calls and parameter passing, **Backpatching**- Technique for handling forward jumps in control flow; essential for Boolean expressions and conditional constructs, **Boolean Expression Handling**- Short-circuit evaluation, relational operators, and efficient translation into intermediate code.

Unit V: Code Optimization and Code Generation (9 Hours)

Optimization Techniques- Constant folding, dead code elimination, loop optimization, and strength reduction; modern emphasis on SSA (Static Single Assignment) form for advanced optimizations, **Data Flow Analysis** -Liveness analysis, reaching definitions, and available expressions; foundation for register allocation and code motion, **Code Generation**- Register allocation strategies, instruction selection, and scheduling; balancing efficiency with hardware constraints, **Modern Compiler Frameworks**- Introduction to GCC and LLVM; role of intermediate representations, JIT compilation, and cross-platform portability

Text Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools (Dragon Book), Pearson/Addison Wesley, 2nd Edition, 2006, ISBN: 9780321486813
2. Torben Ægidius Mogensen, Introduction to Compiler Design, Springer, 3rd Edition, 2024, ISBN: 9783031464591

Reference Books:

1. Y.N. Srikant, Priti Shankar, The Compiler Design Handbook: Optimizations and Machine Code Generation, CRC Press, 2nd Edition, 2008, ISBN: 9781420043822
2. Keith D. Cooper, Linda Torczon, Engineering a Compiler, Morgan Kaufmann, 2nd Edition, 2011, ISBN: 9780120884780
3. Andrew W. Appel, Modern Compiler Implementation in C, Cambridge University Press, Revised Edition, 2004, ISBN: 9780521602724

Steven Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufmann, 1st Edition, 1997, ISBN: 9781558603202.

Course Code	Name of the Course	L	T	P	C
XXPE303Y	**Digital Marketing and E-commerce**	**3**	**0**	**0**	**3**

Course Educational Objectives

CEO1	To provide students with a foundational understanding of digital marketing evolution, strategies, and e-commerce fundamentals in a global and data-driven context.
CEO2	To develop analytical and strategic skills for planning, executing, and measuring digital campaigns using AI, big data, and emerging platforms.
CEO3	To familiarize students with consumer behavior in digital spaces, ethical practices, and trust-building mechanisms for sustainable e-commerce.

Course Outcomes: Towards the end of the course students will be able to

CO1	Explain the evolution, scope, and key differences between traditional and digital marketing, including omnichannel strategies.
CO2	Analyze online consumer behavior, customer journey mapping, and ethical issues in digital commerce.
CO3	Design and implement social media and content marketing strategies, leveraging influencers and viral campaigns.
CO4	Apply emerging digital marketing tools such as email, mobile, video, programmatic advertising, and AI-driven personalization.
CO5	Evaluate e-commerce business models, electronic payment systems, and cybersecurity frameworks for compliance and trust.
CO6	Assess contemporary digital marketing issues, cybersecurity challenges, and governance practices to ensure ethical compliance and operational excellence in e-commerce.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	2	1	-	-	-	1	-	-	1	2	1	1	1
CO2	1	3	-	2	1	1	-	2	1	1	-	2	1
CO3	1	2	3	1	2	-	-	-	2	3	2	2	2
CO4	2	2	2	2	3	-	-	-	2	2	2	2	3
CO5	2	2	1	1	-	1	1	1	-	-	2	1	2
CO6	1	2	-	-	2	2	2	3	1	1	1	1	2
Average	**1.5**	**2.0**	**2.0**	**1.5**	**2.0**	**1.3**	**1.5**	**2.0**	**1.4**	**1.8**	**1.6**	**1.5**	**1.9**

SYLLABUS

Unit I: Fundamentals of Digital Marketing (9 Hours)

Evolution of Digital Marketing- Growth from traditional advertising to data-driven, AI-powered, and omnichannel strategies, **Traditional vs. Digital Marketing**- Key differences in reach, personalization, interactivity, and measurement; digital marketing enables real-time engagement and analytics, **Digital Marketing Channels**- Core platforms: SEO, SEM, social media, email, and mobile; integration with influencer marketing, content marketing, and video platforms, **Digital Strategy and Planning-** Setting objectives, identifying target audiences, selecting channels, and measuring ROI with analytics tools; emphasis on agile and adaptive strategies.

Unit II: Consumer Behaviour in Digital Space (9 Hours)

Online Consumer Behaviour- How digital platforms shape decision-making; influence of reviews, social proof, and personalized recommendations, **Impact of Digital Technology**- Mobile-first browsing, AI-driven personalization, subscription models, and the rise of omnichannel shopping, **Customer Journey Mapping**- Identifying touchpoints from awareness to post-purchase; use of personas, analytics, and predictive modelling to optimize engagement, **Privacy, Ethics, and Trust**- Data protection, GDPR compliance, ethical marketing, and building consumer trust through transparency and secure transactions.

Unit III: Social Media and Content Marketing (9 Hours)

Social Media Platforms- Role of Facebook, Instagram, LinkedIn, YouTube, and Twitter in brand building; emphasis on emerging platforms like TikTok and Threads for short-form engagement, **Content Creation and Storytelling-** Crafting compelling narratives, visual content, and interactive formats; focus on authenticity and user-generated content, **Influencer and Viral Marketing-** Leveraging influencers, micro-influencers, and community-driven campaigns; strategies for virality through trends, memes, and shareable content, **Analytics and ROI Measurement**- Using KPIs, engagement metrics, conversion tracking, and advanced tools like Google Analytics, Meta Insights, and AI-driven dashboards to measure impact.

Unit IV: Emerging Digital Marketing Platforms (9 Hours)

Email Marketing- Personalized campaigns, automation, and segmentation; integration with CRM and AI-driven targeting, **Mobile Marketing** App-based engagement, push notifications, location-based services, and mobile-first design strategies, **Video Marketing-** Short-form videos, live streaming, interactive video ads; platforms like YouTube, TikTok, and Instagram Reels driving engagement, **Programmatic Advertising**- Automated ad buying, real-time bidding, and display ads optimized through AI and big data analytics, **AI and Personalization**-Machine learning for predictive targeting, chatbots, recommendation engines, and hyper-personalized customer experiences, **Immersive Technologies**-- Virtual reality (VR), augmented reality (AR), and gamification enhancing interactive brand experiences.

Unit V: E-Commerce Fundamentals (9 Hours)

Introduction to E-Commerce- Scope and applications across industries; integration with digital marketing and global trade, **Business Models**- B2B, B2C, C2C, and C2B; modern extensions like D2C (Direct-to-Consumer), subscription-based, and platform-driven ecosystems, **Electronic Payment Systems**- Digital wallets, UPI, cryptocurrency, and contactless payments; emphasis on security and convenience, **E-Banking**- Online banking services, mobile banking apps, and integration with fintech solutions, **Cybersecurity and Regulations**-- Data protection, encryption, fraud prevention, and compliance with global frameworks (GDPR, PCI DSS, IT Act in India).

Text Books:

1. Dave Chaffey, Fiona Ellis-Chadwick, Digital Marketing: Strategy, Implementation and Practice, Pearson Education, 9th Edition, 2025, ISBN: 9781292738086
2. Kenneth C. Laudon, Carol Guercio Traver, E-Commerce 2023–2024: Business, Technology, Society, Pearson Education, 18th Edition, 2023, ISBN: 9781292449722

Reference Books:

1. Philip Kotler, Hermawan Kartajaya, Iwan Setiawan, Marketing 5.0: Technology for Humanity, Wiley, 1st Edition, 2021, ISBN: 9781119668510
2. Tracy L. Tuten, Michael R. Solomon, Social Media Marketing, SAGE Publications, 3rd Edition, 2020, ISBN: 9781529731990.

3. Jonah Berger, Contagious: How to Build Word of Mouth in the Digital Age. Simon & Schuster, Reprint Edition, 2016, ISBN: 9781451686586
4. P.T. Joseph, S.J., E-Commerce: An Indian Perspective, PHI Learning, 7th Edition, 2023, ISBN: 9789391818913

Course Code	Name of the Course	L	T	P	C
XXPC3004	**Environmental Sustainability**	**3**	**0**	**0**	**3**

Course Educational Objectives (CEOs)

CEO-1	To build foundational environmental awareness among engineering students by developing an understanding of ecosystems, natural resources, environmental pollution, and the concept of ecological and carbon footprints, enabling them to recognize the environmental consequences of human and technological activity.
CEO-2	To develop competence in Green Computing principles, sustainable data center design, and energy-efficient software engineering practices, enabling students to design, code, and operate computing systems that minimize energy consumption and carbon emissions.
CEO-3	To equip students with the ability to apply modern technologies such as Artificial Intelligence, the Internet of Things, and Cloud Computing to environmental monitoring, climate prediction, and sustainable governance, positioning Computer Science as an active enabler of climate action and the Sustainable Development Goals.

Course Outcomes: Towards the end of the course, students will be able to

CO-1	**Analyze** ecosystem dynamics, biodiversity loss, and human-induced environmental impacts, and relate them to global sustainability challenges.
CO-2	**Apply** Green Computing principles to the design of energy-efficient hardware, virtualization strategies, and sustainable software engineering practices.
CO-3	**Evaluate** the complete e-waste lifecycle and design effective recycling, recovery, and circular-economy strategies for electronic hardware.
CO-4	**Design** smart environmental monitoring systems using IoT sensor networks and AI/ML models for real-time air, water, and climate data analysis.
CO-5	**Assess** the carbon footprint and life-cycle environmental impact of computing infrastructure, including cloud and data center operations.
CO-6	**Formulate** IT governance policies and organizational strategies aligned with global sustainable development goals (SDGs) and ESG frameworks.

COs	**PROGRAMME OUTCOMES (PO1–PO11)**											**PSOs**	
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO-1	2	2	-	1	-	2	3	2	-	1	-	1	1
CO-2	2	2	3	1	2	1	3	1	-	-	1	3	2

CO-3	1	2	2	2	1	2	3	3	1	1	1	2	1
CO-4	2	2	3	2	3	1	3	1	2	1	1	3	3
CO-5	2	3	2	2	2	2	3	2	1	1	2	2	2
CO-6	1	1	1	1	1	3	3	3	2	2	2	1	2
Average	**1.7**	**2.0**	**1.8**	**1.5**	**1.5**	**1.8**	**3.0**	**2.0**	**1.0**	**1.0**	**1.2**	**2.0**	**1.8**

SYLLABUS

Unit I: Fundamentals of Environmental Science and Ecological Footprint

Multidisciplinary nature of environmental studies; ecosystems, structure and function, energy flow, and biodiversity conservation. Natural resources — forest, water, energy, and mineral resources — and their sustainable utilization. Environmental pollution: air, water, soil, and noise pollution, causes and control measures. Concept of Ecological Footprint and Carbon Footprint — definitions, calculation methods, and their relevance to individuals, organizations, and IT infrastructure. Overview of the United Nations Sustainable Development Goals (SDGs) and their linkage to technology-driven development.

Unit II: Green Computing and Sustainable Software Engineering

Introduction to Green Computing — goals, approaches, and green metrics (PUE, DCiE, carbon intensity). Energy-efficient hardware design: low-power processors, server consolidation, and virtualization. Sustainable data centre design: cooling technologies, renewable-powered data centres, and energy management systems. Sustainable Software Engineering: principles of green coding, algorithmic efficiency, energy profiling of applications, and the Green Software Foundation's Software Carbon Intensity (SCI) specification. Cloud computing sustainability — carbon-aware computing, workload scheduling, and server less architectures for reduced energy footprint.

Unit III: E-Waste Management and Circular Economy in IT

Electronic waste: sources, classification, and growth trends in the IT and electronics industry. Hazardous constituents of e-waste (heavy metals, flame retardants) and associated health and environmental risks. E-waste lifecycle management: collection, dismantling, recycling, and safe disposal technologies. Circular Economy models for computing hardware — refurbishment, remanufacturing, and product-as-a-service approaches. Extended Producer Responsibility (EPR) frameworks and national/international e-waste regulations and compliance standards.

Unit IV: AI, IoT, and Cloud for Environmental Monitoring

IoT architectures and sensor networks for environmental monitoring — air and water quality, soil moisture, and weather parameters. Machine Learning and AI models for climate prediction, disaster forecasting, and early-warning systems. Integration of remote sensing and Geographic Information Systems (GIS) with AI for land-use and deforestation analysis. Applications in smart grids, precision agriculture, and wildlife conservation. Cloud-based platforms for aggregating, analyzing, and visualizing large-scale environmental data streams.

Unit V: Sustainable IT Policy, Carbon Accounting, and Climate-Tech Governance

Carbon accounting frameworks for IT and cloud infrastructure (Scope 1, 2, and 3 emissions). Life Cycle Assessment (LCA) methodology applied to computing systems and digital products. Environmental, Social, and Governance (ESG) reporting for technology organizations. IT governance frameworks for sustainability audits, compliance, and green certifications (ISO 14001, Energy Star).

Case studies of smart cities and climate-tech governance; alignment of digital policy with global climate agreements and the SDGs.

Text Books:

1. Erach Bharucha, Textbook of Environmental Studies for Undergraduate Courses, Universities Press, 4th Edition, 2025, ISBN: 9789393330901
2. Anne Currie, Sarah Hsu, Sara Bergman, Building Green Software: A Sustainable Approach to Software Development and Operations, O'Reilly Media, 1st Edition, 2024, ISBN: 9781098150624.

Reference Books:

1. Toby Velte, Anthony Velte, Robert C. Elsenpeter, Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line, McGraw-Hill, 1st Edition, 2008, ISBN: 9780071599238.
2. Joel J. P. C. Rodrigues, Parul Agarwal, Kavita Khanna (Eds.), IoT for Sustainable Smart Cities and Society, Springer, 1st Edition, 2022, ISBN: 9783030895563.
3. Ashutosh Kumar Dubey, Abhishek Kumar, Sushil Kumar Narang, Moonis Ali Khan, Arun Lal Srivastav (Eds.), Visualization Techniques for Climate Change with Machine Learning and Artificial Intelligence, Elsevier, 1st Edition, 2022, ISBN: 9780323997140.

Course Code	Name of the Course	L	T	P	C
XXOE301Y	Ethical Hacking Essentials	2	0	0	2

Course Educational Objectives	
CEO1	To provide students with fundamental knowledge of ethical hacking concepts, cyber security principles, and various types of cyber attacks.
CEO2	To develop the ability to identify system vulnerabilities, perform footprinting, scanning, and basic penetration testing techniques using appropriate tools.
CEO3	To equip students with knowledge of network security, web application security, cryptography, and legal/ethical issues related to cyber security.
Course Outcomes: Towards the end of the course students will be able to :	
CO1	Understand the fundamental concepts of ethical hacking, types of hackers, and cyber security principles.
CO2	Apply footprinting and social engineering techniques for information gathering.
CO3	Analyze network scanning, enumeration, and vulnerability assessment methods.
CO4	Demonstrate knowledge of system hacking techniques, password cracking, and malware analysis.
CO5	Evaluate network and web application attacks such as DoS, SQL injection, and session hijacking.
CO6	Understand cryptography concepts, security mechanisms, and legal/ethical issues in cyber security.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	1	-	-	-	-	-	-	-	-	2	1
CO2	3	3	2	2	-	-	-	-	-	-	-	3	2
CO3	3	3	3	2	2	-	-	-	-	-	-	3	3
CO4	3	3	3	2	1	-	-	-	-	-	-	3	2
CO5	3	2	3	3	2	-	-	-	-	-	-	3	3
CO6	3	2	2	2	3	-	-	-	-	-	-	3	3
AVG	3	2.5	2.3	2.0	1.6	-	-	-	-	-	-	2.8	2.3

SYLLABUS

UNIT I – Introduction to Ethical Hacking (10 Hours)

Basics of hacking- definition and concept, types of hackers (white hat, black hat, grey hat). Ethical hacking concepts: scope and importance, difference between ethical hacking and cybercrime. Cyber security fundamentals: CIA triad (confidentiality, integrity, availability), security policies and mechanisms. Phases of hacking: reconnaissance, scanning, gaining access, maintaining access, covering tracks. Network attacks: DoS (Denial of Service), DDoS (Distributed Denial of Service).

UNIT II - Footprinting & Social Engineering (8 Hours)

Footprinting Concepts- Definition and objectives of footprinting, types of footprinting: passive and active footprinting. Footprinting Techniques: Search engines, WHOIS lookup, DNS interrogation. Information Gathering: Email tracking, website footprinting, IP tracking. Social Engineering Concepts: Human-based attacks, computer-based attacks. Social Engineering Attacks: Phishing, pretexting, baiting, shoulder surfing.

UNIT III - Scanning, Enumeration & Vulnerability Analysis (8 Hours)

Network Scanning- Objectives and methodology, host discovery techniques. **Port Scanning Techniques:** TCP scan, UDP scan, SYN scan. **Scanning Tools:** Nmap basics, introduction to vulnerability scanners. **Enumeration:** NetBIOS enumeration, SNMP enumeration, user and group identification. **Vulnerability Analysis:** Types of vulnerabilities, vulnerability assessment methods.

UNIT IV- System Hacking & Malware (10 Hours)

System Hacking Concepts- Gaining access to systems, privilege escalation**. Password Cracking Techniques:** Brute force attack, dictionary attack**. Malware Concepts:** Definition and characteristics**. Types of Malware:** Viruses**,** worms, trojans**,** rootkits, spyware**. Keyloggers and Backdoors:** Working and impact**. Malware Detection and Prevention:** Antivirus tools**,** security practices**.**

UNIT V - Network Security, Web Attacks & Cryptography (10 Hours)

Session Hijacking- Concept and types. **Web Application Attacks:** SQL Injection, Cross-Site Scripting (XSS). **Basics of Cryptography:** Encryption and decryption, symmetric and asymmetric keys, Digital signatures and hashing concepts. **Network Security Mechanisms:** Firewalls, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Authentication and authorization concepts, HTTPS/SSL basics. **Cyber Laws and Ethics:** IT Act basics, legal and ethical responsibilities.

Text Books:

1. Matt Walker, CEH All-in-One Exam Guide, 5th Edition, McGraw-Hill, 2022, ISBN: 9781264269952
2. Omar Santos, Michael Gregg, CEH Cert Guide, 5th Edition, Pearson IT Certification, 2025, ISBN: 9780135485880
   EC-Council, Ethical Hacking & Countermeasures, 12th Edition, EC-Council Academia, 2022, ISBN: 9798885931014

Reference Books:

1. Phineas Ochse, Web App Hacker's Handbook, 2024, Independently Published, 2024, ISBN: 9798334969186
2. William Stallings, Network Security Essentials, 6th Edition, Pearson+, 2022, ISBN: 9780134527338
3. Kennedy, O'Gorman, Kearns, Aharoni, Graham, Metasploit Guide, 2nd Edition, No Starch Press, 2025, ISBN: 9781718502987
4. Georgia Weidman, Penetration Testing, 1st Edition, No Starch Press, 2014, ISBN: 9781593275648
5. 5. Stallings & Brown, Computer Security, 5th Edition, Pearson, 2024, ISBN: 9781292729015
6. Jon Erickson, Hacking: Art of Exploitation, 2nd Edition, No Starch Press, 2023–25, ISBN: 9789399595779

Course Code	Name of the Course	L	T	P
XXPC3005	Generative AI Tools	3	0	0

Prerequisite: Introduction to Machine Learning(XXPC3002)

Course Educational Objectives

CEO1	Apply generative AI tools and prompt engineering strategies across real-world tasks involving text, image, video, and automated workflow generation.
CEO2	Develop AI-assisted software solutions by leveraging code generation, debugging, documentation, and rapid prototyping tools to build functional applications.
CEO3	Construct end-to-end data engineering pipelines and present data-driven insights through visualization, AI-generated narratives, and integrated capstone projects.

Course Outcomes: Towards the end of the course, students will be able to:

CO1	Utilize leading Generative AI tools (ChatGPT, Gemini, Claude, Copilot, Notebook LM, Sora, Gamma, In Video) for text, image, video, and productivity tasks.
CO2	Design and apply effective prompts using zero-shot, few-shot, chain-of-thought, and role-based prompting strategies for diverse AI applications.
CO3	Employ AI-assisted coding tools (GitHub Copilot, ChatGPT, Claude, Lovable) for code generation, debugging, review, documentation, and rapid software prototyping.
CO4	Build and automate basic data engineering pipelines including ETL processes, data transformation, and visualization using AI-powered tools.
CO5	Design and execute no-code AI workflow automation pipelines using tools such as n8n, KNIME, and Zapier for event-driven process automation.
CO6	Integrate knowledge across all units to independently define, design, and deliver a capstone project that applies GenAI tools, prompt engineering, AI coding, and data engineering to solve a real-world problem.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES									PSOs	
	1	2	3	4	5	6	7	8	9	10	2
CO1	2	1	-	-	3	-	-	-	2	2	2
CO2	3	2	2	-	2	-	-	-	1	1	3
CO3	3	3	3	2	3	-	-	-	1	1	3

CO4	2	3	3	2	3	-	-	-	1	1	3
CO5	2	2	3	3	3	1	-	-	2	1	3
CO6	3	3	3	3	3	2	1	1	3	2	3
AVG	**2.50**	**2.33**	**2.80**	**2.50**	**2.83**	**1.50**	**1.00**	**1.00**	**1.67**	**1.33**	**2.83**

SYLLABUS

UNIT I - Generative AI Tools and Applications (10 hours)

AI Platform Exploration- Hands-on introduction to ChatGPT, Gemini, Claude, and Perplexity. Compare response quality, reasoning capability, and use-case suitability across platforms for a common set of tasks (summarization, Q&A, creative writing).Productivity and Research AI Tools- Explore Microsoft Copilot and NotebookLM for document summarization, meeting notes, and research synthesis. Upload PDFs and academic papers; generate structured insights and citations.AI Image and Creative Media Generation- Use Adobe Firefly to generate and edit images with text prompts. Explore style transfer, generative fill, and text-to-image features. Compare outputs with variations in prompt specificity. AI Video Generation- Produce short AI-generated videos using Sora, Veo, and Runway. Experiment with text-to-video, image-to-video, and video-editing prompts. Evaluate output quality, length, and motion coherence.AI Presentation and Content Creation- Build a professional presentation on an AIML topic using Gamma. Create and edit a short explainer video using InVideo. Compare AI-driven slide generation with manual design. No-Code AI Workflow Automation- Design and execute automated workflows using n8n, KNIME, and Zapier. Build a pipeline that triggers on an event (e.g., form submission), processes data with an AI step, and sends a notification output.

UNIT II - Prompt Engineering (10 Hours)

Prompt Engineering Fundamentals: Study and apply prompt design principles. Compare outputs from vague vs. well-structured prompts. Explore the effect of context, constraints, tone, and format instructions on AI responses. Zero-Shot, Few-Shot and Chain-of-Thought Prompting: Implement zero-shot classification, few-shot translation/summarization, and chain-of-thought reasoning tasks. Measure and compare accuracy and coherence across prompting strategies. Role-Based and Persona Prompting: Assign roles to the AI model (e.g., senior developer, data analyst, medical expert) and observe how persona influences output quality, depth, and tone. Apply to real-world scenarios such as code review and report writing. AI Workflow Optimization and Advanced Prompting: Build effective prompts for content generation, analytical reporting, and code assistance. Explore system prompts, iterative refinement, and multi-turn conversation strategies for complex task completion.

UNIT III AI-Assisted Coding (10 Hours)

AI Pair Programming and Code Generation: Use GitHub Copilot, ChatGPT, or Claude to generate Python/JavaScript functions from natural language descriptions. Evaluate correctness, style adherence, and edge-case handling of generated code. AI-Powered Code Debugging and Review: Submit buggy code samples to AI tools and analyze the suggested fixes. Apply AI-generated code review suggestions to improve readability, efficiency, and security of an existing codebase. Software Documentation Using AI: Auto-generate docstrings, README files, API documentation, and inline comments for a provided Python module using AI tools. Compare AI-generated documentation with manually written equivalents. Rapid Prototyping with AI — Web App using Lovable: Build a functional web application prototype using Lovable (AI-powered full-stack app builder). Translate a product brief into a working UI with backend logic using natural language instructions, with minimal manual coding.

UNIT IV- Data Engineering Fundamentals (10 Hours)

Data Pipeline Design and ETL Concepts: Design a simple ETL (Extract, Transform, Load) pipeline using Python (Pandas) and KNIME. Extract data from a CSV/API source, apply transformation rules, and load the cleaned data into an output file or database table. Data Collection, Processing and Transformation: Collect real-world data from a public API or web source. Apply data cleaning, normalization, and feature engineering steps. Document the pipeline with AI-assisted code generation and automated inline comments. Visualization, Reporting and Data Storytelling: Build a data visualization dashboard using

Python (Matplotlib/Seaborn/Plotly) and integrate with an AI narrative tool to auto-generate a written summary of trends, outliers, and business insights from the dataset.

UNIT V - Industrial Interaction (5 Hours)

Applications of Gen AI Tools- AI in Manufacturing, AI in Healthcare, AI in Education, AI in finance, Human AI Collaboration Real Time Use case Discussion- Industrial Case Studies, Real world Examples of Gen AI adoption in Logistics, Energy, and Agriculture

Text Books:

1. Chris Fregly and Antje Barth, Generative AI on AWS: Building Context-Aware Multimodal Reasoning Applications, 1st Edition, O'Reilly Media, 2024. ISBN: 9781098159221
2. Chip Huyen, AI Engineering: Building Applications with Foundation Models, 1st Edition, O'Reilly Media, 2025,ISBN: 9781098166304

Reference Books:

1. James Phoenix and Mike Taylor, Prompt Engineering for Generative AI, 1st Edition, O'Reilly Media, 2024,ISBN: 9781098153434
2. Jay Alammar and Maarten Grootendorst, Hands-On Large Language Models, 1st Edition, O'Reilly Media, 2024,ISBN: 9781098150969

Course Code	Name of the Course	L	T	P	C
XXPC3105	**Generative AI Tools Lab**	**0**	**0**	**2**	**1**
Prerequisite:	**Generative AI Tools (XXBS2001)**				
Course Educational Objectives					
CEO1	Provide hands-on practical exposure to leading Generative AI platforms and prompt engineering techniques for real-world text, image, video, and workflow-automation tasks.				
CEO2	Develop practical proficiency in AI-assisted software development — code generation, debugging, documentation, and rapid prototyping — using contemporary GenAI coding tools.				
CEO3	Enable students to design, build, and evaluate end-to-end data engineering and no-code AI workflow automation pipelines, culminating in an integrated capstone project that solves a real-world problem.				
Course Outcomes: Towards the end of the course, students will be able to:					
CO1	Apply leading Generative AI platforms (ChatGPT, Gemini, Claude, Perplexity, Microsoft Copilot, NotebookLM) to perform text, research, and productivity tasks such as summarization, Q&A, and document synthesis.				
CO2	Design and apply zero-shot, few-shot, chain-of-thought, and role-based prompting strategies to optimize AI-generated outputs across diverse real-world scenarios.				
CO3	Apply AI-assisted coding tools (GitHub Copilot, ChatGPT, Claude, Lovable) to generate, debug, document, and rapidly prototype functional software applications.				
CO4	Analyze and construct automated data engineering pipelines — covering ETL, data transformation, and visualization — using AI-powered tools to derive data-driven insights.				
CO5	Design and evaluate no-code AI workflow automation pipelines (n8n, KNIME, Zapier) for event-driven business process automation.				
CO6	Evaluate and integrate GenAI tools, prompt engineering, AI-assisted coding, and data engineering techniques to independently design, implement, and present a capstone project addressing a real-world problem.				

CO-PO & PSO Mapping

COs \ POs & PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	1	-	-	3	-	-	1	2	2	-	2	3
CO2	2	2	1	-	3	-	-	1	1	2	-	2	3
CO3	2	2	3	1	3	-	-	1	2	1	-	3	2

CO4	2	3	2	2	3	1	2	1	1	2	1	2	3
CO5	1	2	3	1	3	1	1	1	2	1	1	3	2
CO6	3	3	3	2	3	2	2	2	3	3	2	3	3
Average	**2.00**	**2.17**	**2.40**	**1.50**	**3.00**	**1.33**	**1.67**	**1.17**	**1.83**	**1.83**	**1.33**	**2.50**	**2.67**

List of Experiments
Experiment 1: Multi-Platform Generative AI Exploration Hands-on comparison of ChatGPT, Gemini, Claude, and Perplexity for summarization, Q&A, and creative writing tasks. Use Microsoft Copilot and NotebookLM to synthesize research papers and generate structured insights with citations.
Experiment 2: AI Image and Video Generation Generate and edit images using Adobe Firefly (text-to-image, generative fill, style transfer). Produce short AI videos using Sora/Veo/Runway (text-to-video and image-to-video) and evaluate output quality and coherence.
Experiment 3: AI Presentation Design and No-Code Workflow Automation Build an AI-generated presentation using Gamma and a short explainer video using InVideo. Design and execute an event-driven automation pipeline using n8n/KNIME/Zapier that processes data with an AI step and sends a notification.
Experiment 4: Prompt Engineering Fundamentals Compare outputs from vague vs. well-structured prompts. Implement zero-shot classification, few-shot translation/summarization, and chain-of-thought reasoning tasks; measure accuracy and coherence across strategies.
Experiment 5: Role-Based Prompting and Advanced Prompt Optimization Apply role/persona-based prompting (e.g., senior developer, data analyst) to code review and report-writing tasks. Explore system prompts, iterative refinement, and multi-turn conversation strategies.
Experiment 6: AI Pair Programming, Debugging, and Code Review Use GitHub Copilot, ChatGPT, or Claude to generate Python/JavaScript functions from natural-language descriptions. Submit buggy code samples, analyze AI-suggested fixes, and apply AI-generated review suggestions for readability, efficiency, and security.
Experiment 7: AI-Assisted Documentation and Rapid Prototyping Auto-generate docstrings, README files, and API documentation for a Python module using AI tools. Build a functional web-application prototype using Lovable, translating a product brief into a working UI with backend logic.
Experiment 8: ETL Pipeline Design with AI-Assisted Tools Design a simple ETL pipeline using Python (Pandas) and KNIME: extract data from a CSV/API source, apply transformation rules, and load cleaned data into an output file or database table.
Experiment 9: Data Collection, Cleaning, and AI-Assisted Documentation

Collect real-world data from a public API or web source; apply cleaning, normalization, and feature-engineering steps. Document the pipeline using AI-assisted code generation and automated inline comments.

Experiment 10: Data Visualization, Storytelling, and Capstone Mini-Project

Build a data-visualization dashboard (Matplotlib/Seaborn/Plotly) integrated with an AI narrative tool to auto-generate insights. Independently design, implement, and present an end-to-end capstone project integrating GenAI tools, prompt engineering, AI coding, and data engineering to solve a real-world problem.

Text Books:

1. Chris Fregly and Antje Barth, Generative AI on AWS: Building Context-Aware Multimodal Reasoning Applications, 1st Edition, O'Reilly Media, 2024. ISBN: 9781098159221.

2. Chip Huyen, AI Engineering: Building Applications with Foundation Models, 1st Edition, O'Reilly Media, 2025. ISBN: 9781098166304.

Reference Books:

1. James Phoenix and Mike Taylor, Prompt Engineering for Generative AI, 1st Edition, O'Reilly Media, 2024. ISBN: 9781098153434.

2. Jay Alammar and Maarten Grootendorst, Hands-On Large Language Models, 1st Edition, O'Reilly Media, 2024. ISBN: 9781098150969.

3. John Berryman and Albert Ziegler, Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications, 1st Edition, O'Reilly Media, 2024. ISBN: 9781098156152.

Course Code	**Name of the Course**	L	T	P	C	
XXOE301Y	**Introduction to Quantum Technology**	**3**	**0**	**0**	**3**	

Course Educational Objectives	
CEO1	Students will be able to analyze and design quantum circuits using quantum gates
CEO2	Students be capable of analyzing the efficiency and complexity of quantum algorithms
CEO3	They will be able to apply different quantum algorithm and programming to solve the societal problems.
Course Outcomes: Towards the end of the course, students will be able to:	
CO1	Describe quantum mechanics concepts and quantum computer systems
CO2	Explain and apply linear algebra operations in quantum computing
CO3	relate the role of quantum technology in secure computing
CO4	test and use modern tools for industry 4.0
CO5	Able to independently analyze and find solutions in quantum computing.
CO6	Design and evaluate quantum programs for known algorithms

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3					3		3							
CO2	3	3	3	3											
CO3	3	2	3	3											
CO4	3	3	2		2										
CO5	3	3	3		2		1								
CO6	3	2	3	3		1		1							

Avg.	3	2.6	2.8	3	2	2	1	2							

SYLLABUS

Unit 1: Basics of Quantum Computing **(9 Hours)**

Fundamentals of Quantum Computing-Basic Quantum Mechanics – superposition, entanglement, measurement, Classical vs Quantum Systems – differences in computation and information representation, Quantum Supremacy – milestones and current state, Quantum Computer Architectures – superconducting qubits, trapped ions, photonic systems, Quantum Applications – cryptography, optimization, simulation, machine learning, Mathematical Foundations-Complex Numbers, .Linear Algebra – vectors, matrices, inner products, orthonormality, Vector Spaces & Basis – spanning sets, dimensions, Gram-Schmidt orthogonalization, Bra-Ket Formalism – Dirac notation for quantum states, Key Inequalities – Cauchy-Schwarz and triangle inequality in quantum proofs, Qubits & Quantum States-Quantum Bits (Qubits) – representation and measurement, Bloch Sphere – geometric visualization of a qubit, Multiple Qubits – tensor products, entanglement, Quantum State Representation – linear combinations, superposition, and orthogonality

Unit 2: Quantum Gates & State Representations **(9 Hours)**

Quantum Gates- Single-Qubit Gates – Pauli-X, Y, Z, Hadamard, Phase, Rotation gates, Multi-Qubit Gates – CNOT, Toffoli, SWAP, controlled rotations, Universal Gate Sets – gate combinations for universal quantum computation, Tensor Products & Composite States-Tensor Products – representing multi-qubit systems using vector and matrix tensor products, Composite States – entanglement, separability, and inner precomputations, Density Operators & Mixed States-Density Operator Basics – pure vs mixed states, key properties, Characterizing Mixed States – statistical mixtures, decoherence, Trace & Partial Trace – reduced density matrices for subsystems, Bloch Vector Representation – geometric visualization of density operators.

Unit 3: Quantum Algorithms **(9 Hours)**

Foundations of Quantum Algorithms, Classical Computation on Quantum Computers – hybrid approaches combining classical pre-processing with quantum execution., Complexity Classes – relationship between classical (P, NP) and quantum (BQP, QMA) complexity, b- exploiting superposition for simultaneous evaluation, Core Quantum Algorithms-Deutsch's Algorithm – the simplest quantum speedup, Deutsch–Jozsa Algorithm – demonstrating exponential separation from classical, Simon's Algorithm – periodicity problem, precursor to Shor's, Shor's Algorithm – integer factorization, breaking RSA cryptography, Grover's Algorithm – quadratic speedup for unstructured search, Quantum Information Applications, Quantum Error Correction – stabilizer codes, fault tolerance, Entanglement Applications – resource for algorithms and communication, Quantum Teleportation – transmitting quantum states via entanglement, Super dense Coding – doubling classical communication capacity with entanglement.

Unit 4: Quantum Programming & Implementations **(9 Hours)**

Quantum Programming & Computation, Quantum Programming Languages – Qiskit, Cirq, Quipper, and emerging frameworks, Probabilistic vs Quantum Computations – differences in randomness, superposition, and measurement outcomes, Quantum Cryptography – quantum key distribution (QKD), security advantages, Quantum Information Theory – entropy, information transfer, and quantum communication, Quantum Technology Foundations, Quantum Measurement – collapse of wavefunction, measurement postulates, No-Cloning Theorem – fundamental limits on copying quantum states, Scalability Challenges – noise, decoherence, and hardware limitations in real systems, Quantum Computing Implementations, Quantum Annealing – optimization problems, D-Wave systems, Superconducting Qubits – IBM Q, Google Sycamore, Ion Trap Quantum Computers – Honeywell, IonQ implementations, Photonic Quantum Systems – emerging optical approaches for scalability

Unit 5: Quantum Applications (9 Hours)

Industry & Societal Applications, Healthcare & Genomics – personalized medicine, genetic sequencing, medical imaging, Drug Discovery & Material Science – molecular simulation, catalyst design, advanced materials, Finance & Security – portfolio optimization, fraud detection, quantum-safe cryptography, Transportation & Logistics – route optimization, supply chain management, Global & Strategic Applications, Climate Modelling & Environmental Analysis – accurate simulations for sustainability, Quantum Warfare & Défense – secure communication, strategic simulations, Post-Quantum Cryptography – algorithms resistant to quantum attacks, AI & Data-Driven Applications, Machine Learning & Artificial Intelligence – quantum-enhanced ML models, optimization, Machine Vision & Imaging – advanced pattern recognition, medical diagnostics, Personalized Medicine – tailoring treatments using quantum simulations.

Text Books:

1. Osama M. Raisuddin, Suvranu De, Quantum Computing for Engineers, 1st edition, (2026), Springer publications, ISBN: 978-3-031-09789-1

2. Bernhardt, C. Quantum computing for everyone, (2019). 1st edition, MIT Press, ISBN: 9780262039253

Reference Books:

1. Eleanor Rieffel & Wolfgang Polak, Quantum Computing: A Gentle Introduction, MIT Press, 1st edition, (2011), ISBN: 978-0-262-01506-6.

2. David McMahon, Quantum computing explained, Wiley-interscience, John Wiley & Sons, (2008), ISBN: 978-0-470-17911-6

3. Noson S. Yanofsky, Mirco A. Mannucci, Quantum computing for computer scientists, Cambridge University Press, 1st edition, (2008), ISBN: 978-0-521-87996-5

4. Explorations in Quantum Computing, C. P. Williams and S. H. Clearwater, Springer, New York, 2nd edition, 2018, ISBN: 978-3-319-92434-1

SEMESTER VII [FOURTH YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	PEC	XXPE401Y	Professional Elective-IV Option-1: Micro Service Architechture Option-2: DevOps and CI/CD	3	0	0	3	10	30	60	100
2	PEC	XXPE402Y	Professional Elective-V Option-1: Natural Language Processing Option-2: Edge Computing & IoT Security	3	0	0	3	10	30	60	100
3	MDM	XXMD4001	Multi Disciplinary Minors - I	3	0	0	3	10	30	60	100
4	OEC	XXOE401Y	Open Elective - II	2	0	0	2	10	30	60	100
5	HSMC	HSHS4001	Entrepreneurship and Start-ups	2	0	0	2	10	30	60	100
6	EEC	XXEE4201	Technical Seminar	0	0	2	1	-	-	100	100
7	EEC	XXEE4202	Summer Internship-II	-	-	-	1	60	-	40	100
			Total:	**13**	**0**	**2**	**15**	**110**	**150**	**440**	**700**

Course Code	Name of the Course	L	T	P	C
XXPE402Y	**Edge Computing and IoT Security**	**3**	**0**	**0**	**3**

Course Educational Objectives

CEO1	To provide a comprehensive understanding of the foundational concepts, communication models, and architectures of Internet of Things (IoT) and Edge Computing.
CEO2	To analyze security threats, vulnerabilities, and privacy challenges across edge devices, networks, and cloud integrations.
CEO3	To explore advanced defense mechanisms including cryptography, Blockchain, AI-driven threat response, zero-trust architectures, and emerging technologies in edge environments.

Course Outcomes: Towards the end of the course students will be able to

CO1	Explain the concepts, architectures, and communication protocols of IoT and Edge Computing environments.
CO2	Identify security risks, attack surfaces, and threat landscapes across IoT devices and networks.
CO3	Implement authentication, encryption, and access control models for data and network security.
CO4	Apply hardware-level security mechanisms, zero-trust frameworks, and privacy-preserving techniques to edge systems.
CO5	Integrate AI and blockchain technologies to design resilient security mechanisms for smart applications.
CO6	Evaluate emerging security paradigms (5G, post-quantum cryptography) and design secure IoT/Edge solutions for industrial and real-world domains.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	-	-	-	-	-	-	-	-	-	3	1
CO2	3	3	2	-	2	-	-	-	-	-	-	3	2
CO3	3	3	3	2	2	1	-	-	-	-	-	3	2
CO4	3	3	3	2	3	2	1	-	-	-	-	3	3
CO5	3	3	3	3	3	2	2	2	-	-	1	3	3
CO6	3	3	3	3	3	2	2	2	2	2	2	3	3
Average	3	2.83	2.8	2.5	2.6	1.75	1.5	2	2	2	1.5	3	2.33

SYLLABUS

Unit 1: Fundamentals of IoT and Edge Computing (9 Hours)

IoT Overview – architecture, protocols, and real-world applications, Edge Computing Concepts – role of edge, need for low-latency processing, comparison with cloud computing, IoT Communication Models

– MQTT, CoAP, HTTP, and emerging lightweight protocols, Edge Devices – sensors, gateways, edge servers, and integration with AI accelerators.

Unit 2: IoT Security Challenges (9 Hours)

Threat Landscape – vulnerabilities, attack surfaces, and evolving risks, Authentication & Access Control – secure device onboarding, identity management, and role-based access, Data Security – encryption, secure storage, and transmission safeguards, Network Security – intrusion detection, firewalls, VPNs, and anomaly monitoring

Unit 3: Edge Security Mechanisms (9 Hours)

Secure Edge Architectures – trusted execution environments, secure boot, hardware root of trust, Blockchain for IoT – decentralized trust, smart contracts, lightweight consensus mechanisms, AI for Security – anomaly detection, predictive analytics, adaptive threat response, Privacy-Preserving Techniques– differential privacy, federated learning.

Unit 4: Applications and Case Studies (9 Hours)

Smart Cities – secure surveillance, traffic optimization, and resilient energy grids, Healthcare IoT – patient monitoring, medical device security, and telemedicine platforms, Industrial IoT – SCADA protection, predictive maintenance, and secure automation, Autonomous Systems – connected vehicles, drones, and safety-critical edge deployments

Unit 5: Emerging Trends and Future Directions (9 Hours)

5G and IoT – ultra-low latency, network slicing, and secure edge integration, Zero Trust Architectures – continuous verification, micro-segmentation, and adaptive access control, Quantum Security – post-quantum cryptography and quantum-resistant protocols. Future Challenges – scalability, interoperability, and sustainable IoT deployments.

Text books:

1. Bahga, A., & Madisetti, V. K. (2014). Internet of things: A hands-on approach, 1st Edition, Universities Press, ISBN: 978-8173719338
2. Shi, W., & Dustdar, S. (2020). Foundations of edge computing, 1st Edition, . Wiley, ISBN: 978-1119551713

Reference Books:

1. Roman, R., Zhou, J., & Lopez, J. (2013). Security and privacy in Internet of things: Challenges and solutions, 1st Edition.. Springer, ISBN: 978-3642336269
2. Suthar, R., & Patel, P. (2021). IoT security: Advances in authentication, 1st Edition. CRC Press, ISBN: 978-0367331587
3. Minoli, D. (2019). IoT security: Advances in authentication and access control, 1st Edition, Wiley, ISBN: 978-1119456735
4. Zhang, Y., & Green, R. (2019). Security and privacy for mobile edge computing, 1st Edition, Springer, ISBN: 978-3030168357

Course Code	Name of the Course	L	T	P	C
XXMD4001	COMPUTER VISION	3	0	0	3

Course Educational Objectives

CEO1	Provide fundamental understanding of image processing and computer vision concepts, including image formation, representation, and basic processing techniques.
CEO2	Develop the ability to analyze and extract meaningful features from images and videos using computer vision algorithms and geometric models.
CEO3	Enable students to apply computer vision techniques and machine learning approaches to solve real-world problems such as object detection, tracking, and 3D reconstruction.

Course Outcomes: Towards the end of the course, students will be able to:	
CO1	Explain fundamental concepts of image processing and computer vision, including image representation, color models, and real-world applications.
CO2	Apply image formation models, camera geometry, and projection techniques for 2D–3D scene understanding
CO3	Analyze image features using edge detection, corner detection, and feature extraction techniques to represent objects.
CO4	Implement motion estimation, optical flow, and 3D reconstruction techniques for analyzing dynamic scenes.
CO5	Evaluate segmentation and shape representation techniques for object detection and recognition.
CO6	Design computer vision solutions using machine learning and deep learning techniques for real-world applications.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	1								1		3	2
CO2	3	2	1		1							3	2
CO3	3	2	2	1	2							3	2
CO4	3	2	2	2	3							3	3
CO5	2	3	2	2	3							2	3
CO6	2	3	3	2	3	1			1	1	1	3	3
AVG													

SYLLABUS

Unit 1 – Computer Vision **(10 Hours)**

Introduction to Computer Vision- Fundamentals, scope, and applications in robotics, healthcare, surveillance, and autonomous systems, Image Formation and Representation Camera models, image acquisition, pixel representation, colour spaces (RGB, HSV, YCbCr), Image Processing Basics Filtering, convolution, edge detection (Sobel, Canny), thresholding, histogram equalization, Feature Extraction Corners (Harris), blobs (LoG, DoG), SIFT, SURF, ORB descriptors, Pattern Recognition Template matching, statistical classifiers, introduction to machine learning in vision, Applications Object detection, face recognition, medical imaging, autonomous navigation

Unit II- Imaging Systems & 3D Reconstruction (8 Hours)

Monocular Imaging Systems- Principles of single-camera vision, orthographic vs. perspective projection, **Camera Models & Calibration** Pinhole camera model, intrinsic/extrinsic parameters, calibration techniques (Zhang's method), **Binocular Imaging & Multiple View Geometry** Epipolar geometry, fundamental matrix, applications in stereo vision, **Depth Estimation Techniques**- Stereo vision, shape from shading, depth from defocus, photometric stereo,.**3D Scene Reconstruction** Structure-from-motion (SfM), multi-view stereo, point cloud generation, surface reconstruction.

Unit III – Image Preprocessing & Feature Extraction (8 Hours)

Image Preprocessing and Representation- Continuous vs. discrete representation, grayscale and color models, normalization, noise reduction, **Edge Detection Techniques** Gradient-based methods (Sobel, Prewitt), advanced detectors (Canny), **Corner Detection** Harris corner detector, Shi-Tomasi method, applications in tracking and mapping, **Feature Extraction Methods** Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB), Histogram of Oriented Gradients (HOG),.**Feature Matching and Object Representation**- Brute-force matching, FLANN-based matching, object descriptors, representation for recognition tasks.

Unit IV – Motion Analysis & 3D Vision (10 Hours)

Motion Estimation and Optical Flow- Lucas–Kanade and Horn–Schunck methods, dense vs. sparse optical flow, applications in video analysis, **Regularization Theory in Motion Computation** Variational approaches, smoothness constraints, handling noise and ambiguities in motion fields, **Structure from Motion and Tracking-** Recovering 3D structure from image sequences, feature tracking across frames, bundle adjustment,.**Epipolar Geometry and Stereo Correspondence-** Fundamental and essential matrices, epipolar constraints, stereo matching algorithms,.**3D Reconstruction and Motion Applications**- Multi-view reconstruction, dynamic scene analysis, applications in robotics, AR/VR, and autonomous navigation.

Unit V – Segmentation, Shape & Recognition (10 Hours)

Image Segmentation- Contour-based methods, region-based approaches, thresholding, clustering (k-means, mean-shift), **Active Contours and Level Sets-** Snakes model, energy minimization, level set evolution for flexible boundary detection, **Shape Representation-** Fourier descriptors, wavelet transforms, medial axis, applications in shape matching., **Object Recognition Techniques-** Template matching, statistical classifiers, Bag-of-Features model, machine learning approaches (SVM, Random Forests),.**Deep Learning Basics for Vision** Convolutional Neural Networks (CNNs), layers (convolution, pooling, fully connected), training basics, applications in image classification and detection.

Text Books:

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2022, ISBN: 9780137848560

2. Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Edition, Springer, 2023, ISBN: 9783030343743

Reference Books:

1. Richard Hartley, Andrew Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, ISBN: 9780521540513

2. Gary Bradski, Adrian Kaehler, Learning OpenCV, 1st Edition, O'Reilly Media, 2008, ISBN: 9780596516130

3. David A. Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2nd Edition, Pearson Education, 2023, ISBN: 9781292455563

4. Mohamed Elgendy, Deep Learning for Vision Systems, 1st Edition, Manning Publications, 2020, ISBN: 9781617296192

Course Code	Name of the Course	L	T	P	C
XXPE401Y	Micro Services Architecture	3	0	0	3

Prerequisite: Object-Oriented Programming through Java (XXES2001), Database Management Systems (XXPC2002), Computer Networks (XXPC2005)	
Course Educational Objectives	
CEO1	To provide students with a comprehensive understanding of microservices architectural principles, distributed systems concepts, and how they compare with monolithic and service-oriented architectures.
CEO2	To develop students' ability to design, decompose, and implement scalable, independently deployable microservices using modern communication protocols, containerization, and orchestration technologies.
CEO3	To familiarize students with production-grade engineering concerns in microservices — including distributed data management, resilience, security, and observability — preparing them for careers in cloud-native software engineering.
Course Outcomes: Towards the end of the course students will be able to	
CO1	Analyze the limitations of monolithic architectures and apply domain-driven design principles to identify appropriate microservice boundaries and decomposition strategies. (Analyze)
CO2	Design RESTful, gRPC-based, and event-driven APIs for synchronous and asynchronous inter-service communication in a distributed application. (Design)
CO3	Apply distributed data management patterns — including Database-per-Service, Saga, CQRS, and event sourcing — to maintain data consistency across microservices. (Apply)
CO4	Design and implement containerized microservices using Docker, and deploy and orchestrate them at scale using Kubernetes. (Design)
CO5	Evaluate resilience and fault-tolerance strategies (circuit breaker, retries, bulkhead, rate limiting) and observability practices (logging, metrics, distributed tracing) for production microservice systems. (Evaluate)
CO6	Analyze security, API gateway, and service mesh architectures to design secure, scalable, and governable microservice ecosystems. (Analyze)

CO-PO & PSO Mapping

COs	**PROGRAMME OUTCOMES**											**PSOs**	
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	2	3	1	1	-	-	1	1	1	1	3	2
CO2	1	1	3	1	3	-	-	-	1	1	1	3	3
CO3	2	2	3	2	2	1	-	-	1	1	1	2	3

CO4	1	1	3	2	3	1	1	-	2	1	1	3	3
CO5	2	2	2	2	3	1	1	2	2	1	1	2	2
CO6	1	2	2	1	2	1	1	3	1	1	2	2	3
Average	1.5	1.7	2.7	1.5	2.3	1.0	1.0	2.0	1.3	1.0	1.2	2.5	2.7

SYLLABUS

Unit I: Introduction to Microservices Architecture **(9 Hours)**

Monolithic vs. Microservices- Characteristics, benefits, and drawbacks of monolithic, SOA, and microservices architectural styles; when to (and when not to) adopt microservices. Core Characteristics- Independent deployability, single responsibility, decentralized data management, and organizing around business capabilities. Domain-Driven Design (DDD)- Bounded contexts, ubiquitous language, aggregates, and their use in identifying service boundaries. Decomposition Strategies- Decomposing by business capability and subdomain; strangler fig pattern for migrating legacy monoliths. Twelve-Factor App Principles- Configuration, dependencies, and stateless processes as a foundation for cloud-native services.

Unit II: Service Design and Communication Synchronous Communication **(9 Hours)**

RESTful API design, richardson maturity model, versioning strategies, and gRPC for high-performance inter-service calls. Asynchronous/Event-Driven Communication- Message brokers (Kafka, RabbitMQ), publish-subscribe vs. point-to-point messaging, event notification, and event-carried state transfer. API Gateway Pattern- Request routing, composition, protocol translation, and cross-cutting concerns (authentication, rate limiting) at the edge. Service Discovery- Client-side vs. server-side discovery, service registry patterns (e.g., Eureka, Consul). Contract Testing- Consumer-driven contracts to safely evolve independently deployed services.

Unit III: Data Management in Microservices Database-per-Service Pattern **(9 Hours)**

Decentralized data ownership and the challenges of maintaining referential integrity across services. Distributed Transactions- Limitations of two-phase commit in microservices; the Saga pattern (choreography-based and orchestration-based) for managing long-running business transactions. CQRS (Command Query Responsibility Segregation)- Separating read and write models to optimize for scalability and query flexibility. Event Sourcing- Storing state as a sequence of events; rebuilding state and enabling audit trails. API Composition- Aggregating data from multiple services for client-facing queries.

Unit IV: Containerization and Orchestration Containerization Fundamentals **(9 Hours)**

Docker images, containers, Dockerfiles, and container registries; benefits over virtual machines for microservice packaging. Container Orchestration- Kubernetes architecture (pods, deployments, services, ConfigMaps, Secrets), self-healing, and horizontal autoscaling. CI/CD for Microservices- Continuous integration and continuous deployment pipelines tailored to independently releasable services. Service Mesh- Sidecar proxy pattern, traffic management, and observability using tools such as Istio or Linkerd. Infrastructure as Code- Managing microservice infrastructure using declarative configuration.

Unit V: Resilience, Security, and Observability Resilience Patterns .(9 Hours)

Circuit breaker, retry with backoff, bulkhead isolation, timeouts, and rate limiting to prevent cascading failures. Observability- The three pillars: centralized logging, metrics collection (Prometheus/Grafana), and distributed tracing (Jaeger/Zipkin) across service boundaries. Security in Microservices- Authentication and authorization (OAuth2, JWT), service-to-service (mutual TLS) security, and API gateway-based access control. Governance and Compliance- API management, versioning policies, and monitoring for regulatory compliance in distributed systems. Contemporary Issues- Serverless and Function-as-a-Service as an evolution of microservices; challenges of microservices at hyperscale.

Text Books:

1. Sam Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, 2nd Edition, 2021, ISBN: 9781492034025
2. Chris Richardson, Microservices Patterns: With Examples in Java, Manning Publications, 1st Edition, 2018, ISBN: 9781617294549

Reference Books:

1. Kasun Indrasiri, Prabath Siriwardena, Microservices for the Enterprise: Designing, Developing, and Deploying, Apress, 1st Edition, 2018, ISBN: 9781484238578
2. Brendan Burns, Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services, O'Reilly Media, 1st Edition, 2018, ISBN: 9781491983645
3. Eric Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison-Wesley Professional, 1st Edition, 2003, ISBN: 9780321125217

Course Code	Name of the Course	L	T	P	C
XXPE402Y	Natural Language Processing	3	0	0	3

Course Educational Objectives

CEO1	To analyze the leading trends and systems in natural language processing.
CEO2	Make them understand the concepts of morphology, syntax, semantics and pragmatics of the language and that they are able to give the appropriate examples that will illustrate the above-mentioned concepts.
CEO3	To recognize the significance of pragmatics for natural language understanding Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Understand approaches to syntax and semantics in NLP.
CO2	Understand approaches to discourse, generation, dialogue and summarization within NLP.
CO3	Understand current methods for statistical approaches to machine translation.
CO4	Understand machine learning techniques used in NLP, including the probabilistic context-free grammars and unsupervised methods, as applied within NLP.
CO5	Design NLP models for tasks like text classification, sentiment analysis, and information retrieval.
CO6	Evaluate different NLP techniques and models based on performance metrics and real-world applications.

CO-PO & PSO Mapping

COs	PROGRAMME OUTCOMES											PSOs	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3	2	1		2							3	2
CO2	3	3	2		2				1			3	2
CO3	3	3	2	1	2							3	3
CO4	3	2	3	2	3							3	3
CO5	3	3	3	2	3				2			3	3
CO6	3	3	2	3	2				2	1		3	3

AVG	**2.83**	**2.67**	**2.17**	**1.33**	**2.33**	**0**	**0**	**0**	**0.83**	**0.17**	**0**	**3.00**	**2.67**

SYLLABUS

Unit-I: Introduction to Natural Language Processing **(10 Hours)**

Foundations: Need for NLP – importance of processing human languages in AI systems**,** Challenges in NLP – ambiguity, variability, context-dependence, computational complexity**,** History of NLP – rule-based systems → statistical NLP → deep learning approaches. Phases of NLP-Language Modelling – predicting word sequences, probability-based models, Grammar-Based Models – context-free grammars, government & binding theory, lexical functional grammar, Statistical Modelling – probabilistic approaches, n-gram models**,** Modern Language Models – neural language models, transformers (introductory overview for next units), Morphological Analysis – handling word forms and inflections, Syntax & Parsing – dependency parsing, constituency parsing basics, Semantics & Pragmatics – meaning representation, discourse context, Applications of NLP – chatbots, machine translation, sentiment analysis, speech recognition

Unit II: Word Level Analysis **(10 Hours)**

Regular Expressions – pattern matching for text processing, **Finite State Automata** – modelling word recognition, **Morphological Parsing** – analyzing word structure (stems, affixes), **Human Morphological Parsing** – cognitive perspective on word formation**, Spelling Error Detection & Correction** – edit distance, probabilistic models, **Words & Word Classes** – nouns, verbs, adjectives, etc, **Part-of-Speech Tagging** – rule-based, stochastic, and modern approaches, **Handling Unknown Words** – smoothing, sub word models. Syntax-Level Analysis: **Context-Free Grammar (CFG)** – formal rules for sentence structure, **Phrase & Sentence Constructions** – syntactic categories and structures, **Parsing Techniques** –Top-down parsing, Bottom-up parsing, Early parser, CYK parser, **Probabilistic Parsing** – estimating rule probabilities, PCFGs, **Challenges with PCFGs** – ambiguity, data sparsity syntactic relations between words, **Constituency vs Dependency Parsing** – comparative approaches.

Unit III. Semantic Analysis **(10 Hours)**

Meaning Representation – capturing semantics in computational form**, Characteristics of Representation Languages** – expressiveness, ambiguity handling**, Syntax-driven Semantic Analysis** – linking grammar and meaning**, Semantic Grammars** – rule-based semantic structures. **Lexical Semantics** – meaning of words and word classes, **Relations among Lexemes** – synonymy, antonymy, hyponymy, meronymy, **WordNet** – lexical database for semantic relations, **Internal Structure of Words** – morphology and semantics. **Ambiguity** – lexical, syntactic, semantic, **Word Sense Disambiguation** – approaches and challenges, **Selectional Restrictions** – semantic constraints for disambiguation, **Robust WSD** – statistical and machine learning approaches, **Context-based WSD** – discourse and contextual cues**, Knowledge Sources** – dictionaries, corpora, ontologies, **WSD Evaluation** – metrics and benchmarks. Information Retrieval & Discourse**: Information Retrieval** – indexing, ranking, relevance, **Other IR Tasks** – query expansion, document classification, **Discourse Analysis** – coherence, cohesion, discourse context, **Discourse Context & Word Knowledge** – role in semantic interpretation.

Unit IV: Natural Language generation **(8 Hours)**

Architecture of Generators – pipeline for text generation, **Surface Realization** – converting abstract representations into natural text, **Discourse Planning** – structuring coherent multi-sentence output, **Generation Approaches** – template-based, phrase-based, feature-based methods, **Knowledge-Based NLG** – rule-driven and symbolic approaches, **Modern Neural NLG** – sequence-to-sequence models, transformers, GPT-style generation. Machine Translation (MT): MT Systems – architecture and workflow, Challenges in MT – linguistic diversity, ambiguity, cultural context, Core Concepts – transfer metaphor, interlingua idea (meaning-based translation), Approaches to MT- Direct translation, Rule-based translation, Transfer-based translation, Interlingua-based translation, Statistical MT (SMT), Example-based MT (EBMT), Semantic/knowledge-based MT, Neural MT (NMT) – encoder-decoder, attention, transformer-based

Unit V: Advanced NLP and Applications (8 Hours)

Deep Learning in NLP: Neural Architectures – RNN, LSTM, GRU for sequential data, Transformers – attention mechanism, encoder-decoder models, BERT and Variants – contextual embeddings, fine-tuning for tasks. Core Applications- Text Classification – spam detection, topic categorization, Sentiment Analysis – polarity detection, opinion mining, Named Entity Recognition (NER) – extracting people, places, organizations, Question Answering Systems – fact-based QA, reading comprehension models, Chatbots & Conversational AI – dialogue systems, intent recognition, response generation. Recent Trends & Challenges- Large Language Models (LLMs) – GPT, T5, multilingual transformers, Multimodal NLP – combining text with vision and speech, Low-Resource NLP – handling underrepresented languages, Ethical & Social Challenges – bias, fairness, privacy, explainability, Efficiency & Deployment – model compression, distillation, edge deployment

Text Books:

1. Daniel Jurafsky & James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Pearson Education, 2nd Edition, 2009, **ISBN:** 9780131873216

2. Hobson Lane, Maria Dyshel, Cole Howard, Hannes Hapke, Natural Language Processing in Action, Second Edition, Manning Publications, 2025, ISBN: 9781617299445

Reference Books:

1. Yoav Goldberg, Neural Network Methods in Natural Language Processing, Morgan & Claypool Publishers, revised edition 2022, Synthesis Lectures on Human Language Technologies, **ISBN:** 9781627052986
2. Akshi Kumar, Language Intelligence: Expanding Frontiers in Natural Language Processing, Wiley–IEEE Press, **E-book:** December 2024, **Hardcover:** January 2025, ISBN: 978-1-394-29727-6
3. J. Allen, Natural Language Understanding, Pearson Education, second edition, 2008, ISBN: 9780805303346.

Course Code	Name of the Course	L	T	P	C
XXMD4001	**AR/VR Technology**	3	0	0	3
Prerequisite:	Object-Oriented Programming Through JAVA (XXES2001)				
Course Educational Objectives					
CEO1	To provide students with a comprehensive understanding of the fundamental concepts, taxonomy, and evolution of Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR), grounded in the principles of human visual and spatial perception that underlie immersive experiences.				
CEO2	To develop students' ability to design, model, and render immersive 3D environments, and to apply tracking, sensing, and interaction technologies in building responsive AR/VR/MR systems using industry-standard engines and SDKs.				
CEO3	To familiarize students with advanced AR/VR/MR application development practices, 3D user-interface design, and emerging cross-domain applications, while enabling them to evaluate the usability, ethical, and accessibility implications of immersive technologies.				
Course Outcomes: Towards the end of the course student will be able to:					
CO1	Analyze the fundamental concepts, taxonomy (Milgram's Reality-Virtuality continuum), and evolution of AR, VR, and MR, along with the human perceptual and vestibular principles that govern immersion and presence.				

CO2	Design stereoscopic 3D graphics pipelines and rendering techniques — including viewing transformations, depth cues, and display optics — suited to head-mounted and other immersive display systems.
CO3	Apply tracking and sensing technologies (inertial, optical, marker-based, markerless, and SLAM-based) along with haptic and gesture-based input devices to build responsive AR/VR systems.
CO4	Design and develop interactive AR/VR/MR applications using industry-standard engines and SDKs such as Unity3D, Unreal Engine, ARKit, ARCore, Vuforia, and OpenXR.
CO5	Evaluate 3D user-interface techniques — navigation, selection, manipulation, and system control — and assess usability, comfort, and cybersickness factors that determine the quality of an immersive experience.
CO6	Analyze emerging AR/VR/MR applications across domains such as education, healthcare, gaming, industrial training, and digital twins, and evaluate the associated ethical, privacy, and accessibility considerations.

CO-PO & PSO Mapping

COs	**Programme Outcomes (POs)**											**PSOs**	
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	3	1	1	2	1	1	1	1	1	1	3	2
CO2	2	2	3	2	3	1	-	1	1	1	1	3	3
CO3	2	2	2	2	3	1	1	1	1	1	1	2	3
CO4	1	2	3	2	3	1	-	1	1	1	1	3	3
CO5	1	2	2	2	2	1	1	2	2	1	1	2	2
CO6	1	3	1	1	2	2	1	3	2	2	2	2	3
Average	1.5	2.3	2.0	1.7	2.5	1.2	1.0	1.5	1.3	1.2	1.2	2.5	2.7

SYLLABUS

Unit I: Foundations of AR, VR, and MR **(9 Hours)**

Introduction to Augmented Reality, Virtual Reality, and Mixed Reality — definitions, history, and evolution. Milgram's Reality-Virtuality continuum and taxonomy of immersive technologies. Human perception for immersive systems — visual perception, depth cues, the vestibular and proprioceptive systems, and their role in presence and immersion. Classification of display systems — head-mounted displays (HMDs), CAVE systems, handheld/mobile AR, and projection-based displays. Overview of the current AR/VR/MR industry, hardware ecosystem, and application landscape.

Unit II: 3D Computer Graphics and Rendering for Immersive Systems **(9 Hours)**

Coordinate systems, transformations, and the 3D graphics pipeline for immersive rendering. Viewing frustums, projection, and stereoscopic rendering for binocular displays. Optics of head-mounted displays — lenses, field of view (FOV), interpupillary distance (IPD), and distortion correction. Real-time rendering considerations — latency, frame rate, motion-to-photon delay, and foveated rendering. Spatial (3D) audio and its contribution to immersion. Performance optimization techniques for real-time immersive graphics.

Unit III: Tracking, Sensing, and Interaction Devices **(9 Hours)**

Degrees of freedom (3-DoF and 6-DoF) tracking systems. Inertial Measurement Units (IMUs) and sensor fusion. Optical, marker-based, and markerless tracking; fiducial markers and natural feature tracking. Simultaneous Localization and Mapping (SLAM) for spatial tracking. Depth sensing technologies — structured light, time-of-flight (ToF), and LiDAR. Hand, eye, and gesture

tracking; haptic feedback devices and controllers. Calibration and registration challenges in AR/VR systems.

Unit IV: AR/VR/MR Application Development (9 Hours)

Application development using Unity3D and Unreal Engine for immersive experiences. AR development kits — Apple ARKit, Google ARCore, and Vuforia; the OpenXR and WebXR standards for cross-platform development. Principles of 3D user-interface (3D UI) design — navigation and wayfinding, selection and manipulation techniques, system control, and symbolic input in immersive environments. Prototyping and iterative design workflows for AR/VR applications. Identifying and mitigating cybersickness and user-comfort issues.

Unit V: Advanced Topics and Emerging Applications (9 Hours)

Mixed Reality and diminished reality; collaborative and social VR environments. Digital twins and the industrial metaverse. AR/VR/MR applications in education and training, healthcare and surgical simulation, gaming and entertainment, and industrial design/manufacturing. Ethical, privacy, security, and accessibility considerations in immersive technologies. Emerging trends — AI-driven content generation, neural rendering, spatial computing, and the broader Extended Reality (XR) ecosystem.

Text Books:

1. Alan B. Craig, Understanding Augmented Reality: Concepts and Applications, Morgan Kaufmann, 1st Edition, 2013, ISBN: 9780240824086
2. Steven M. LaValle, Virtual Reality, Cambridge University Press, 1st Edition, 2023, ISBN: 9781107198937

Reference Books:

1. Jason Jerald, The VR Book: Human-Centered Design for Virtual Reality, ACM Books / Morgan & Claypool Publishers, 1st Edition, 2015, ISBN: 9781970001129
2. Joseph J. LaViola Jr., Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, Ivan Poupyrev, 3D User Interfaces: Theory and Practice, Addison-Wesley Professional, 2nd Edition, 2017, ISBN: 9780134034324
3. Jonathan Linowes, Krystian Babilinski, Augmented Reality for Developers: Build Practical Augmented Reality Applications with Unity, ARCore, ARKit, and Vuforia, Packt Publishing, 1st Edition, 2017, ISBN: 9781787286436

Course Code	Name of the Course	L	T	P	C
XXPE401Y	DevOps and CI/CD	3	0	0	3
Prerequisite:	Ethical Software Engineering (XXPC2006), Operating Systems (XXPC2004), Computer Networks (XXPC2005), Cloud Computing (XXPE301Y)				
Course Educational Objectives					
CEO1	To provide students with a comprehensive understanding of DevOps culture, principles (CALMS), and the end-to-end software delivery lifecycle that bridges development and IT operations.				

CEO2	To develop students' ability to design, automate, and manage build–test–release pipelines using industry-standard version control, continuous integration, and continuous delivery/deployment tools and practices.
CEO3	To familiarize students with infrastructure automation, containerization, orchestration, monitoring, and site reliability engineering practices required to build and operate scalable, secure, production-grade systems.

Course Outcomes: Towards the end of the course students will be able to:

CO1	Analyze the limitations of traditional, siloed SDLC/waterfall practices and apply DevOps culture and the CALMS framework (Culture, Automation, Lean, Measurement, Sharing) to a software delivery workflow. (Analyze)
CO2	Design Git-based version control workflows and automated Continuous Integration (CI) pipelines — including build, static analysis, and automated testing stages — using tools such as Jenkins and GitHub Actions/GitLab CI. (Design)
CO3	Apply configuration management and Infrastructure as Code (IaC) practices using tools such as Ansible and Terraform to provision, configure, and manage immutable, reproducible environments. (Apply)
CO4	Design containerized Continuous Delivery/Deployment (CD) pipelines using Docker and Kubernetes, incorporating GitOps workflows for automated, auditable releases. (Design)
CO5	Evaluate monitoring, logging, and observability strategies (Prometheus, Grafana, the ELK stack, distributed tracing) to assess and ensure the reliability and performance of deployed applications. (Evaluate)
CO6	Analyze DevSecOps practices and deployment strategies (blue-green, canary, rolling updates) to design secure, resilient, and progressively-releasable production systems. (Analyze)

CO-PO & PSO Mapping

COs	**Programme Outcomes**											**PSOs**	
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	3	1	1	2	1	1	2	2	1	1	2	2
CO2	2	2	3	1	3	-	1	1	2	1	1	3	2
CO3	2	2	3	2	3	1	2	1	2	1	2	3	3
CO4	1	2	3	2	3	1	2	1	2	1	1	3	3
CO5	2	3	2	2	3	1	2	2	2	2	1	2	3
CO6	1	3	2	1	2	2	1	3	1	1	2	2	2
Average	**1.7**	**2.5**	**2.3**	**1.5**	**2.7**	**1.2**	**1.5**	**1.7**	**1.8**	**1.2**	**1.3**	**2.5**	**2.5**

SYLLABUS

Unit I: Foundations of DevOps (9 Hours)

Evolution from Waterfall to Agile to DevOps; DevOps culture and the CALMS framework (Culture, Automation, Lean, Measurement, Sharing). DevOps Lifecycle — Plan, Code, Build, Test, Release, Deploy, Operate, Monitor. Version Control with Git — repositories, branching models (Gitflow, trunk-based development, feature branching), pull requests, and code review workflows. Collaboration and Communication — breaking down silos between Dev, QA, and Ops teams; DevOps maturity models and adoption challenges.

Unit II: Continuous Integration (CI) (9 Hours)

Principles of Continuous Integration — frequent commits, automated builds, and fast feedback loops. Build Automation — using Maven/Gradle and build servers. CI Tools and Pipelines — Jenkins architecture (master-agent model, plugins, Jenkinsfile), GitHub Actions, and GitLab CI/CD; Pipeline-as-Code concepts. Automated Testing in CI — unit, integration, and static code analysis (SonarQube) as pipeline gates. Artifact Management — binary repositories (Nexus, JFrog Artifactory) and dependency/version management.

Unit III: Configuration Management and Infrastructure as Code (IaC) (9 Hours)

Configuration Management Fundamentals — desired-state configuration, idempotency, and drift detection using Ansible playbooks, roles, and inventories (with a comparative overview of Chef and Puppet). Infrastructure as Code — declarative provisioning with Terraform (providers, resources, modules, state management). Immutable Infrastructure — golden images, immutable servers vs. configuration drift. Cloud Provisioning Basics — provisioning compute, storage, and networking resources on AWS/Azure/GCP through IaC.

Unit IV: Continuous Delivery, Containerization, and Orchestration (9 Hours)

Continuous Delivery vs. Continuous Deployment — deployment pipelines and release automation. Containerization — Docker images, containers, Dockerfiles, multi-stage builds, and container registries. Container Orchestration — Kubernetes architecture (pods, deployments, services, ConfigMaps, Secrets, Helm charts), self-healing, and autoscaling. GitOps — declarative, Git-driven continuous deployment using tools such as ArgoCD/Flux. Deployment Strategies — blue-green, canary, and rolling deployments for zero/low-downtime releases.

Unit V: Monitoring, Security, and Site Reliability (9 Hours)

Observability — the three pillars: centralized logging (ELK/EFK stack), metrics collection and visualization (Prometheus, Grafana), and distributed tracing (Jaeger/Zipkin). DevSecOps — shifting security left, static/dynamic application security testing (SAST/DAST) in the pipeline, secrets management, and container/image scanning. Site Reliability Engineering (SRE) — Service Level Indicators/Objectives/Agreements (SLIs/SLOs/SLAs), error budgets, incident management, and post-mortems. Chaos Engineering and Contemporary Trends — resilience testing, and the evolution of DevOps toward Platform Engineering and MLOps.

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis, Nicole Forsgren, The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, IT Revolution Press, 2nd Edition, 2021, ISBN: 9781950508402

2. Jez Humble, David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley Professional, 1st Edition, 2010, ISBN: 9780321601919

Reference Books:

1. Brendan Burns, Joe Beda, Kelsey Hightower, Lachlan Evenson, Kubernetes: Up and Running: Dive into the Future of Infrastructure, O'Reilly Media, 3rd Edition, 2022, ISBN: 9781098110208
2. Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy, Site Reliability Engineering: How Google Runs Production Systems, O'Reilly Media, 1st Edition, 2016, ISBN: 9781491929124
1. 3. Yevgeniy Brikman, Terraform: Up & Running: Writing Infrastructure as Code, O'Reilly Media, 3rd Edition, 2022, ISBN: 9781098116743

Course Code	Name of the Course	L	T	P	C
XXMD4002	Blockchain Technology	3	0	0	3

Course Educational Objectives	
CEO1	To provide students with a comprehensive understanding of the fundamental concepts, architecture, cryptographic foundations, and evolution of Blockchain and distributed ledger technologies.
CEO2	To develop students' ability to design, model, and implement decentralized applications, smart contracts, and consensus mechanisms using industry-standard Blockchain platforms and development tools.
CEO3	To familiarize students with advanced Blockchain topics — including scalability, interoperability, and security — and enable them to evaluate real-world enterprise applications along with the legal, ethical, and regulatory implications of Blockchain adoption.
Course Outcomes: Towards the end of the course students will be able to:	
CO1	Analyze the fundamental concepts, architecture, and evolution of Blockchain technology, along with the cryptographic primitives (hash functions, digital signatures, Merkle trees) that underpin distributed ledgers.
CO2	Analyze and compare consensus mechanisms — Proof of Work, Proof of Stake, and PBFT-based protocols — and their role in achieving distributed agreement in permissioned and permissionless Blockchain networks.
CO3	Design and develop smart contracts using Solidity and build decentralized applications (DApps) deployable on Ethereum and other Blockchain platforms. (Design)

CO4	Apply Blockchain development frameworks and tools — such as Remix, Truffle, Hardhat, Ganache, MetaMask, and Hyperledger Fabric — to build, test, and deploy blockchain-based solutions.
CO5	Evaluate the scalability, security vulnerabilities, and privacy challenges of Blockchain systems, including 51% attacks, double-spending, and smart-contract exploits such as reentrancy.
CO6	Analyze emerging Blockchain applications across domains such as decentralized finance (DeFi), supply chain, healthcare, and government, and evaluate the associated legal, ethical, and regulatory implications.

CO-PO and CO-PSO Mapping

COs	**Programme Outcomes (POs)**											**PSOs**	
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	3	1	2	1	1	-	1	1	1	-	3	2
CO2	2	3	2	3	1	-	-	1	1	1	-	3	3
CO3	2	2	3	2	3	1	-	1	2	1	1	3	3
CO4	1	2	3	2	3	1	1	1	2	1	1	3	3
CO5	2	3	2	3	2	2	1	2	1	1	1	2	3
CO6	1	2	1	2	1	3	2	3	2	2	2	2	3
Avg.	1.8	2.5	2.0	2.3	1.8	1.3	0.7	1.5	1.5	1.2	0.8	2.7	2.8

SYLLABUS

Unit I: Introduction to Blockchain Technology and Cryptographic Foundations (9 Hours)

Distributed ledger technology — centralized, decentralized, and distributed systems. History and evolution of Blockchain, from Bitcoin to Ethereum and beyond. Blockchain architecture — blocks, chains, nodes, hashing, and timestamping. Cryptographic foundations — cryptographic hash functions (SHA-256), Merkle trees, public-key cryptography, and digital signatures. Types of Blockchain networks — public, private, permissioned, and consortium Blockchain. Blockchain versus traditional and distributed databases.

Unit II: Consensus Mechanisms and Blockchain Networks (9 Hours)

The distributed consensus problem, the Byzantine Generals Problem, and the CAP theorem in the Blockchain context. Proof of Work (PoW) — mining, difficulty adjustment, and block propagation. Proof of Stake (PoS) and its variants — Delegated PoS (DPoS) and Leased PoS (LPoS). Practical Byzantine Fault Tolerance (PBFT) and other BFT-based consensus protocols. Blockchain forks — soft forks and hard forks. Peer-to-peer networking in Blockchain and node

types (full nodes, light nodes, miners/validators). Bitcoin's UTXO model versus Ethereum's account-based model.

Unit III: Smart Contracts and Decentralized Application Development (9 Hours)

Introduction to smart contracts — concept, lifecycle, and the Ethereum Virtual Machine (EVM). Solidity programming — data types, functions, modifiers, events, and inheritance. Smart contract design patterns and security best practices. Development and testing tools — Remix IDE, Truffle, Hardhat, Ganache, and MetaMask. Deploying and interacting with decentralized applications (DApps); front-end integration using Web3.js/Ethers.js. Token standards — ERC-20 (fungible tokens), ERC-721 and ERC-1155 (non-fungible tokens).

Unit IV: Enterprise Blockchain, Interoperability, and Scalability (9 Hours)

Permissioned blockchain frameworks — Hyperledger Fabric architecture, chaincode, ordering service, and channels. Introduction to Corda and enterprise blockchain use cases. Blockchain interoperability — cross-chain communication, atomic swaps, sidechains, and bridges. Scalability solutions — sharding and Layer-2 protocols (state channels, rollups, and the Lightning Network). Blockchain oracles and their role in connecting on-chain and off-chain data.

Unit V: Blockchain Security, Applications, and Emerging Trends (9 Hours)

Security vulnerabilities and attacks — 51% attacks, double-spending, Sybil attacks, reentrancy, and front-running. Fundamentals of smart contract auditing and formal verification. Privacy-enhancing techniques — zero-knowledge proofs, zk-SNARKs, ring signatures, and mixing services. Applications across domains — decentralized finance (DeFi), supply chain management, healthcare, digital identity, voting systems, and government records. Legal, regulatory, and ethical considerations in Blockchain adoption. Emerging trends — Web3, DAOs, NFTs, central bank digital currencies (CBDCs), and the metaverse economy.

Text Books:

1. Andreas M. Antonopoulos and David A. Harding, Mastering Bitcoin: Programming the Open Blockchain, O'Reilly Media, 3rd Edition, 2023, ISBN: 9781098150099
2. Andreas M. Antonopoulos and Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, 1st Edition, 2018, ISBN: 9781491971949

Reference Books:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 1st Edition, 2016, ISBN: 9780691171692
2. Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress, 1st Edition, 2017, ISBN: 9781484226032
3. Vikram Dhillon, David Metcalf, and Max Hooper, Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You, Apress, 2nd Edition, 2021, ISBN: 9781484265338

SEMESTER VIII [FOURTH YEAR]											
Sl. No.	**Course Category**	**Course Code**	**Course Title**	**Hours per week**			**Credits**	**Marks for various Exams**			
				L	**T**	**P**		**C. A.**	**M. S.**	**E. S.**	**Total**
1	PE	XXPE403Y	Professional Elective-IV Option-1: Data Visualization Option-2: Smart Device Architecture	3	0	0	3	10	30	60	100
2	MDM	XXMD4002	Multi Disciplinary Minors - II	3	0	0	3	10	30	60	100
3	HSM	HSHS4002	Intellectual Property Rights (IPR)	2	0	0	2	10	30	60	100
4	EEC	XXEE4203	Capstone/Major Project	0	0	12	6	-	-	100	100
			Total:	**8**	**0**	**12**	**14**	**30**	**90**	**280**	**400**

Course Code	Name of the Course	L	T	P	C
XXPE403Y	**Data Visualization**	3	0	0	3
Prerequisite:	Introduction to Machine Learning(XXPC3002), Database Management Systems (XXPC2002), Python Programming (XXES2001)				

CEO1	To provide students with a comprehensive understanding of the principles of visual perception, graphical encoding, and design theory that underlie effective data visualization.
CEO2	To develop students' ability to select, design, and implement static, interactive, and web-based visualizations for diverse data types using industry-standard libraries and business intelligence tools.
CEO3	To familiarize students with advanced visualization techniques for high-dimensional, geospatial, network, and large-scale data, and to build the analytical judgment needed to evaluate the accuracy, clarity, and ethics of a visualization.

CO1	Analyze principles of visual perception, cognition, and graphical encoding (Gestalt principles, pre-attentive attributes, data-ink ratio) to critique and select appropriate chart types for a given dataset.
CO2	Design effective static and interactive visualizations using grammar-of-graphics based tools such as ggplot2, Matplotlib/Seaborn, and D3.js, applying sound principles of color, layout, and typography.
CO3	Apply data preparation techniques — cleaning, aggregation, binning, and dimensionality reduction (PCA, t-SNE) — to transform raw data into visualization-ready form.
CO4	Design interactive dashboards and data-driven narratives using business intelligence platforms such as Tableau/Power BI to support exploratory and explanatory analytics.
CO5	Evaluate visualization techniques for multivariate, geospatial, and network data with respect to accuracy, scalability, and usability, and identify instances of misleading or unethical visual representation.
CO6	Analyze the architectural and perceptual challenges of visualizing large-scale and streaming data, and apply appropriate scalable rendering and visual-analytics strategies.

COs	Programme Outcomes (POs)											PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	2	2	1	1	3	1	2	2	2	1	1	2	2
CO2	2	3	3	1	3	1	1	1	2	1	1	3	2
CO3	2	2	3	2	3	1	1	1	2	1	2	3	3
CO4	1	2	3	2	3	1	2	1	2	1	1	3	3
CO5	2	3	2	2	3	1	2	3	2	2	1	2	3
CO6	1	2	2	1	3	2	2	2	1	1	2	2	2
Average	**1.7**	**2.3**	**2.3**	**1.5**	**3.0**	**1.2**	**1.7**	**1.7**	**1.8**	**1.2**	**1.3**	**2.5**	**2.5**

Unit I: Foundations of Data Visualization **(9 Hours)**

Why visualize data — history and importance of visual communication. Data types (nominal, ordinal, interval, ratio) and data abstraction. Human visual perception and cognition — Gestalt principles, pre-attentive attributes, and working memory limits. Visual encoding — marks and channels (position, length, angle, area, color, shape); ranking channel effectiveness. Data-ink ratio and chart-junk. Chart taxonomy — bar, line, scatter, pie, and their appropriate use. Introduction to visualization ethics — misleading axes, truncated scales, and deceptive design.

Unit II: Visualization Design and the Grammar of Graphics **(9 Hours)**

The Grammar of Graphics — aesthetics, geometries, statistical transformations, scales, and facets. Implementation using ggplot2 (R) and Matplotlib/Seaborn (Python). Static chart families — distribution plots (histogram, box plot, violin plot), relationship plots (scatter, bubble, heatmap), and comparison plots (grouped/stacked bar, small multiples). Color theory — sequential, diverging, and qualitative color scales; accessibility and color-blind-safe palettes. Typography and layout for figures. Storytelling with data — structuring a narrative, annotation, and audience-appropriate design.

Unit III: Interactive and Web-Based Visualization **(9 Hours)**

Fundamentals of D3.js — DOM manipulation, SVG primitives, data binding (enter-update-exit), and scales/axes. Building interactivity — tooltips, hover effects, brushing and linking, zooming and panning. Dashboard design principles — layout, information hierarchy, and interaction design for exploratory analysis. Business intelligence tools — Tableau and Power BI for rapid dashboard construction, calculated fields, filters, and parameters. Alternative interactive libraries — Plotly and Bokeh for Python-based interactive charts.

Unit IV: Multivariate, High-Dimensional, and Geospatial Visualization **(9 Hours)**

Visualizing multivariate data — parallel coordinates plots, scatterplot matrices (SPLOM), and radar charts. Dimensionality reduction for visualization — Principal Component Analysis (PCA), t-SNE, and UMAP. Hierarchical and part-whole data — treemaps, sunburst diagrams, and dendrograms. Network and graph visualization — node-link diagrams, force-directed layouts, and adjacency matrices; tools such as Gephi and D3-force. Geospatial visualization — choropleth maps, cartograms, and point-density maps using Leaflet, Kepler.gl, and GIS fundamentals.

Unit V: Big Data Visualization, Visual Analytics, and Emerging Trends (9 Hours)

Challenges of visualizing large-scale and high-velocity data — sampling, binning, aggregation, and server-side rendering strategies. Streaming and real-time visualization for monitoring and dashboards. Visual analytics — combining automated analysis with interactive visual exploration for decision support. Visualization for machine-learning interpretability — feature-importance plots, SHAP/partial-dependence visualizations, and confusion-matrix heatmaps. Emerging directions — immersive visualization with AR/VR, and rigorous evaluation of visualizations through user studies and perceptual experiments.

Text Books:

1. Tamara Munzner, Visualization Analysis and Design, A K Peters/CRC Press, 1st Edition, 2014, ISBN: 9781466508910
2. Claus O. Wilke, Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures, O'Reilly Media, 1st Edition, 2019, ISBN: 9781492031086

Reference Books:

1. Edward R. Tufte, The Visual Display of Quantitative Information, Graphics Press, 2nd Edition, 2001, ISBN: 9780961392147
2. Scott Murray, Interactive Data Visualization for the Web: An Introduction to Designing with D3, O'Reilly Media, 2nd Edition, 2017, ISBN: 9781491921289
3. Alberto Cairo, The Truthful Art: Data, Charts, and Maps for Communication, New Riders, 1st Edition, 2016, ISBN: 9780321934079

Course Code	Name of The Course	L	T	P	C
XXPE4033Y	Smart Device Architecture	3	0	0	3

Prerequisite: Digital Electronics(XXPC2001), Computer Organization and Architecture(XXPC2003)

Course Educational Objectives

CEO1	To provide students with a strong foundation in the architecture, components, and working principles of smart and embedded devices used in modern IoT-based systems.44
CEO2	To develop students' ability to analyze and design smart device architectures involving microcontrollers, sensors, actuators, and communication interfaces.
CEO3	To enable students to apply concepts of power management, security, and edge intelligence in building efficient, real-world smart device solutions.

Course Outcomes: Towards the end of the course students will be able to :

CO1	Comprehend the fundamental architecture and building blocks of smart devices and embedded systems.
CO2	Analyze the working of microprocessors, microcontrollers, and System-on-Chip (SoC) architectures used in smart devices.

CO3	Design sensor and actuator interfacing mechanisms for real-time data acquisition and control.
CO4	Evaluate wired and wireless communication protocols used for connectivity in smart devices.
CO5	Apply power management and optimization techniques to enhance the energy efficiency of smart devices.
CO6	Integrate security mechanisms and edge intelligence into smart device architectures for reliable IoT deployment.

CO-PO & PSO Mapping

COs	**PROGRAMME OUTCOMES**											**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**
CO1	3	2										3	2
CO2	3		3	2	3							3	3
CO3	3	3		3	3							3	3
CO4	2		3		3							3	2
CO5				3		2						3	3
CO6		3					3					3	3
AVG	2.8	2.7	3	2.7	3	2	3					3	2.7

SYLLABUS

UNIT I - Introduction to Smart Devices **(10 hours)**

Evolution of smart devices, embedded systems vs. smart devices, characteristics of smart devices, applications in IoT, wearables, smart homes, and industrial automation. Basic architecture of a smart device: sensing layer, processing layer, communication layer, and actuation layer. Introduction to Real-Time Operating System (RTOS) concepts and their role in smart devices.

UNIT II - Processing Units in Smart Devices **(10 hours)**

Microprocessors vs. microcontrollers, System-on-Chip (SoC) architecture, overview of ARM Cortex-M/A architectures, memory hierarchy (Flash, RAM, EEPROM), General Purpose Input/Output (GPIO), timers and interrupts, introduction to FPGAs in smart devices. Case study of popular development platforms: Arduino, Raspberry Pi, ESP32.

UNIT III - Sensors, Actuators and Interfacing **(10 hours)**

Types of sensors (temperature, proximity, motion, biometric, environmental) and actuators (motors, relays, servo motors), Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC), communication interfaces: I2C, SPI, UART, signal conditioning and calibration basics.

UNIT IV - Connectivity and Communication Protocols (8 hours)

Wired vs. wireless communication, Bluetooth and BLE, Wi-Fi, Zigbee, LoRa, NFC, introduction to MQTT, CoAP, and HTTP for IoT, network topologies for smart device deployment, and gateway architecture.

UNIT V - Power Management, Security and Edge Intelligenc (8 hours)

Power management techniques - sleep modes, dynamic voltage and frequency scaling, energy harvesting basics; security fundamentals for smart devices - authentication, encryption basics, secure boot; introduction to edge computing and on-device AI/ML inferencing; Over-the-Air (OTA) update mechanisms.

Text Books:

1. Marilyn Wolf, "Computers as Components: Principles of Embedded Computing System Design", 5th Edition, Morgan Kaufmann, 2022,ISBN: 9780128214312
2. Raj Kamal, "Internet of Things: Architecture and Design Principles", 2nd Edition, McGraw Hill Education, 2021,ISBN: 9789390727384

Reference Books:

1. Frank Vahid, Tony Givargis, "Embedded System Design: A Unified Hardware/Software Introduction", Wiley, 2010.ISBN: 9788126508372
2. Adrian McEwen, Hakim Cassimally, "Designing the Internet of Things", Wiley, 1st Edition 2013, ISBN: 9781118430620