B.Tech. in Computer Science and Engineering (Curriculum and Syllabus - 2024-Scheme)

National Institute of Electronics and Information Technology, Aurangabad
A constituent campus of NIELIT Deemed to be University

(An Autonomous Scientific Society of Ministry of Electronics and Information Technology, Government of India)
Dr Babasaheb Ambedkar Marathwada University Campus, Chhatrapati Sambhajinagar (Aurangabad), Maharashtra 431004

B. TECH – COMPUTER SCIENCE AND ENGINEERING

National Institute of Electronics and Information Technology, Aurangabad

Introduction

The B.Tech program in Computer Science Engineering is designed to provide students with a strong foundation in computer science principles and practices, preparing them for the ever-evolving technological landscape. This program integrates theoretical knowledge with practical skills in software development, data structures, algorithms, computer networks, artificial intelligence, and more. Students will gain the ability to innovate and solve complex problems, making them industry-ready professionals and lifelong learners in the field of computer science.

Program Education Objectives (PEO)

- **PEO1: Core Competence**

Graduates will have a solid foundation in computer science and engineering principles, enabling them to analyse, design, and implement complex software and hardware systems.

- **PEO2: Technical Proficiency**

Graduates will develop technical expertise in various domains of computer science, including algorithms, data structures, machine learning, and cybersecurity, allowing them to solve real-world problems effectively.

- **PEO3: Professional Development**

Graduates will demonstrate professional skills such as teamwork, communication, and leadership, enabling them to excel in multidisciplinary teams and adapt to global work environments.

- **PEO4: Lifelong Learning**

Graduates will commit to continuous learning and professional development to stay current with emerging technologies and trends in computer science and related fields.

- **PEO5: Social Responsibility**

Graduates will exhibit ethical decision-making, social responsibility, and environmental consciousness in their professional careers, contributing positively to society.

Program Outcomes (PO)

- **PO1: Engineering Knowledge**

Apply knowledge of mathematics, science, engineering fundamentals, and computer science specialization to solve complex engineering problems.

- **PO2: Problem Analysis**

Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

- **PO3: Design and Development**

Design and develop software and hardware systems that meet specified needs while considering public health and safety, societal, and environmental factors.

- **PO4: Modern Tool Usage**

Utilize modern engineering tools, software, and hardware effectively for modelling, simulation, and analysis of complex engineering activities.

- **PO5: Ethics and Professionalism**

Apply ethical principles and commit to professional ethics, responsibilities, and norms of the computer science engineering practice.

Course Category Wise Credit Distribution:

Course Category	Category Code	Credits
Humanities and Social Sciences	HS	10
Basic Science	BS	16
Engineering Science	ES	08
Program Core	PC	71
Professional Elective	PE	16
Open Elective	OE	06
Project work, Seminar etc	EEC	38
Audit	AU	0
Total Credit		165

Semester wise Structure and Curriculum
for
B.Tech. in Computer Science and Engineering (CSE)

Semester I

3-Week Orientation Programme

S.No	Course Code	Course Title	L	T	P	Credits
1.	BS101	Engineering Physics	2	0	2	3
2.	ES101	Engineering Drawing	1	0	4	3
3.	BS102	Engineering Mathematics-I	3	1	0	4
4.	ES102	Electrical Engineering	2	0	2	3
5.	PC101	Python Programming	2	0	4	4
6.	BS103	Engineering Chemistry	2	1	0	3

Total Credits: 20

Semester II

S.No	Course Code	Course Title	L	T	P	Credits
1.	BS201	Engineering mathematics-II	2	1	0	3
2.	PE201	Professional Elective I	2	1	0	3
3.	HS201	Sports and Yoga	0	1	4	3
4.	PC201	Electronic devices and Circuits	2	0	2	3
5.	ES201	Workshop technology	0	1	2	2
6.	HS202	Communication skills	2	1	2	3
7.	HS203	Design Thinking	0	0	2	1

Total Credits: 18

List of Professional Electives are given in Annexure I

Semester III

S.No	Course Code	Course Title	L	T	P	Credits
1.	BS301	Engineering mathematics-III	2	1	0	3
2.	PC301	C Programming	2	0	4	4
3.	PC302	Data Structure and Algorithm	2	0	4	4
4.	PC303	Object Oriented Programming Using Java	3	0	2	4
5.	PC304	Digital Logic and Circuits	2	1	2	4
6.	HS301	Constitution of India	2	1	0	3
						Total Credits: 22

Semester IV

S.No	Course Code	Course Title	L	T	P	Credits
1.	PC401	Computer Architecture and Organisation	3	0	2	4
2.	PC402	Discrete Mathematics	3	1	0	4
3.	PC403	Design and Analysis of Algorithms	2	1	2	4
4.	PC404	Database Management System	2	1	2	4
5.	PE401	Professional Elective II	2	1	0	3
6.	PE402	Professional Elective III	2	1	0	3
						Total Credits: 22

Semester V						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	PC501	Operating Systems	3	0	2	4
2.	PC502	Machine learning	3	0	2	4
3.	PC503	Software Engineering	3	1	0	4
4.	PC504	Probability and Statistics	3	1	0	4
5.	PC505	Theory of Computation	3	1	0	4
						Total Credits: 20

Semester VI						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	PC601	Computer Network and Cyber Security	2	0	4	4
2.	PC602	Compiler Design	3	0	2	4
3.	PC603	Data Mining	2	1	2	4
4.	PE601	Professional Elective IV	2	0	4	4
5.	PE602	Professional Elective V	2	0	2	3
6.	EEC601	Mini project and Seminar	0	0	16	8
						Total Credits: 27

Semester VII						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	OE701	Open Elective 1	--	--	--	3
2.	OE702	Open Elective 2	--	--	--	3
3.	EEC701	Dissertation Phase-I	0	0	20	10
Total						16
Professional Electives and Open Electives may include relevant online courses from platforms such as SWAYAM, MOOC, NPTEL, NIELIT NSQF, or any other UGC/AICTE-approved sources in the field of Computer Science/Electronics/Electrical Engineering, subject to approval by the Dean (Academics) at the time of registration.						

Semester VIII						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	EEC801	Dissertation Phase-II	0	0	40	20
Total						20

- ***Main emphasis should be on Project Based Learning / Experiential Learning.***
- ***There should be an option to delay internship semester to $7^{th}/8^{th}$ Semester as per institute convenience and availability of internship slots for different group of students.***

Core Subject Syllabus

Course title:	ENGINEERING PHYSICS	Sub code	BS101			
		Structure	L	T	P	C
			2	0	2	3

Course Objective:	• To equip the students with an understanding of the "Scientific Methods"so that they can use the training beneficially in their higher pursuits. • This course gives a balance account of the fundamentals of Physics aswell as some of recent developments in this area best suited to the Engineering applications in different branches.
Course Outcome:	• The student will be able to understand many modern devices andtechnologies based on lasers and optical fibres. • Student can also appreciate various material properties which are usedin engineering applications and devices. • Master fundamental principles of physics applicable to engineering. • Apply physics concepts to solve complex engineering problems. • Develop proficiency in experimental techniques and data analysis. • Integrate physics knowledge across engineering disciplines for problem-solving. • Enhance communication and teamwork skills for effective collaboration in engineering projects.

Content

Module 1: Interference and Diffraction:

Interference in thin film of uniform thickness and non-uniform thickness, Newton's rings, Michel son's interferometer, Fabry-Perot interferometer. Fresnel and Fraunhofer diffraction, Fraunhofer diffraction at circular aperture, plane diffraction grating, determination of wavelength using plane diffraction grating, dispersive power of grating, resolving power of grating.

Module 2: Electrostatics:

Gauss's law and its applications, Divergence and Curl of Electrostatic fields, Electrostatic Potential, Boundary conditions, Work and Energy, Conductors, Capacitors, Laplace's equation, Method of images, Boundary value problems in Cartesian Coordinate Systems, Dielectrics, Polarization, Bound Charges, Electric displacement, Boundary conditions in dielectrics, Energy in dielectrics, Forces on dielectrics.

Module 3: Magnetostatics

Lorentz force, Biot-Savart and Ampere's laws and their applications, Divergence and Curl of Magneto static fields, Magnetic vector Potential, Force and torque on a magnetic dipole,

Module 4: Dielectrics materials:

Magnetic materials, Magnetization, Bound currents, Boundary conditions. Diamagnetic materials, Paramagnetic materials, Ferromagnetic materials, origin of magnetization, Types of magnetic materials-hard materials and soft materials. Dielectrics-Introduction, dielectric constant, polarization, induced dipoles, permanent dipoles,polar and non-polar dielectrics, polarization-an atomic view, types of polarization.

Module 5: Classical Mechanics

Review of Newtoninan Mechanics in rectilinear coordinate system, motion in plane polar coordinates. Conservation Principles. Collision problems and centre of mass frame. Rotation about fixed axis. Non-inertial frames and pseudo forces, rigid bossy systems.

Module 6: Quantum Mechanics/ Physics:

Two−slit experiment. Dual nature of light; Compton Effect; De-Broglie hypothesis; Davisson-Germer Experiment; Phase and group velocities; Uncertainty principle; Wave-function; Schrodinger wave equation; Particle in a finite and infinite potential well; Tunnel effect. Superposition Principle, Continuity Equation for probability density; Normalization . Expectation values .Eigen values and eigen functions Stationary states, Bound states, Applications in one dimension: Particle in a box, 1-D Finite Potential well, Harmonic oscillator.

Reference Books:

1) Fiber optic Communication-D.C.Agarwal. Wheeler Publication, New Delhi

2) Solid state electronic devices-Streetman, Prentice Hall India, New Delhi
3) Electronic devices and circuits-Allen Mottershade, Prentice Hall India, New Delhi
4) Fiber optic communication-Keiser. Mc Graw Hill Publication
5) A course in Electrical Engineering Materials – S.P.Seth,P.V.Gupta, Dhanpat Rai Publication,New Delhi.
6) Engineering physics-Gaur and Gupta, S.Chand Publication
7) Engineering physics-Avadhanalu and Kshirsagar, S.Chand Publication

List of Experiments:

1) Determination of radius of curvature of Plano- convex lens by Newton's ring
2) Determination of wavelength by diffraction grating.
3) Study of CRO (amplitude, frequency, phase measurement).
4) Experiments on electromagnetic induction and electromagnetic braking;
5) LC circuit and LCR circuit;
6) Resonance phenomena in LCR circuits;
7) Magnetic field from Helmholtz coil;
8) Measurement of Lorentz force in a vacuum tube.
9) To study different types of Optical fibres.

Course title:	**ENGINEERING DRAWING**	**Sub code**	ES101			
		Structure	**L**	**T**	**P**	**C**
			1	0	4	3

Course Objective:

- To impart and include proper understanding of the theory of projection.
- Improve the visualization skills.
- To enable the students with various concepts like dimensioning, conventions and standards related to working drawing in order to become professionally efficient.
- To impart the knowledge on understanding and drawing of simple residential/ office building.

Course Outcome:

- Develop proficiency in reading and interpreting engineering drawings and diagrams.
- Acquire skills in creating detailed drawings of electronic components, circuits, and systems.
- Learn industry-standard drafting techniques and conventions for technical drawings.
- Gain knowledge of CAD (Computer-Aided Design) software tools for electronic drafting and modeling.
- Enhance visualization and spatial reasoning abilities essential for engineering design and communication.

Content

Module 1 Introduction to engineering drawing:

Principles of engineering graphics and their significance – drawing instruments and their use – conventions in drawing – lettering – BIS conventions. Dimensioning rules, geometrical construction. Curves used in engineering practice and their constructions: Conic Sections, Special Curves-Cycloids, Epicycloids, Hypocycloids.

Module 2: Projections of Straight Lines:

Projections of points in four quadrants, projections of points in reference plane, line parallel to both the plane, line parallel to one plane and perpendicular to the other, line inclined to one plane and parallel to the other, line inclined to both the reference planes, traces of line, use of traces of line in obtaining projections (all four quadrants should be considered).

Module 3: Projections of planes and solids

Projections of regular planes, inclined to both planes. Projections of regular solids inclined to both planes. Introduction to solids: prisms, pyramid, cylinder, cone, cube, tetrahedron, sphere, projections of above solids with axis inclined to one plane, projections of above solids with axis inclined to both the planes, projection of composite solids (different arrangement of spheres with above solids).

Module 4: Development of Surfaces:

Development of surfaces of right, regular solids – development of prisms, cylinders, pyramids, cones and their parts

Module 5: Orthographic Projections & Isometric views:

Principles of orthographic projections – conventions – first and third angle projections. Projections of points and lines inclined to both the planes. Orthographic projections of different machine parts, sectional orthographic projections. Introduction to pictorial views, isometric projections and isometric views (Isometric and non-isometric planes).

Reference Books:

1) Bhatt N. D., Panchal V. M., "Engineering Drawing", Charotar Publishing House.
2) Dhabhade M. L., "Engineering Graphics", Vol.-I and Vol.-II, Vision Publications, Pune.
3) Mathur, Laxminarayan, "Elements of Engineering Drawing", Jain Publications, New Delhi.

List of Experiments:

1) Introduction to BIS SP – 46 – 1988.
2) Explanation of various drawing instruments, symbols, RF, Dimensioning, etc.
3) Conversion of pictorial views to orthographic / profile views.
4) Projection of points and lines.
5) Projections of planes.
6) Projections of lines and planes using Auxiliary planes.
7) Projections of solids.
8) Section and development of solids.
9) Intersection of solids.
10) Isometric views.
11) Practice of scales, Representative Factor and dimensioning on some practical exemplaryfigure.

Course title:	**ENGINEERING MATHAMATICS-I**	**Sub code**	BS102			
		Structure	**L**	**T**	**P**	**C**
			3	1	0	4

Course Objective:	• To expose student to understand the basic importance of Differential calculus, Integral calculus, Infinite series and Matrix theory in science and engineering.
Course Outcome:	• The terminal objectives of the course are that, on successfulcompletion of teaching-learning and evaluation activities, a student would be able to identify and analyse the problems by applying the fundamental principles of engineering mathematics. • Develop a strong understanding of calculus and its applications in engineering. • Gain proficiency in differential equations and their relevance to engineering systems. • Apply mathematical methods to analyze and model engineering phenomena. • Acquire problem-solving skills essential for advanced engineering courses and

	practical applications.

Contents

Module 1: Differential Calculus:

Functions of single variable: Limit, continuity and differentiability. Mean value theorems: Rolle's theorem, Lagrange's theorem, Cauchy's theorem, Taylor's theorem with remainders, indeterminate forms, curvature, curve tracing.

Module 2: Determinants:

Determinant of a matrix of order one, order two, order three. Properties of determinant, area of triangle, minors and co-factors, adjoint and inverse of a matrix,

Module 3: Integral Calculus:

Fundamental theorem of Integral calculus, mean value theorems, evaluation of definite integrals, Applications in Area, length, volumes and surface of solids of revolutions, Improper integrals: Beta and Gamma functions, differentiation under integral sign.

Module 4: Sequence and Series

Sequences, Infinite series of real and complex numbers, Cauchy criterion, tests of convergence, absolute and conditional convergence, improper integrals, improper integrals depending on aparameter, uniform convergence, power series, radius of convergence.

Module 5: Matrices:

System of linear equations, Augmented matrix, Existence and uniqueness of solution, Gauss elimination method, Elementary row operations, LU decomposition, Row-equivalent systems, Row echelon form, Rank of a matrix, Linear dependence, Consistency of a linear system, Linear combination of solutions, General solution, Types of matrices and their properties, Eigenvalues, Eigen vectors, Eigenvalue problems, Cayley- Hamilton theorem, Similarity of matrices, Diagonalisation, Quadratic form, Reduction to canonical form

Reference Books:

1) Kreyszig, E., Advanced Engineering Mathematics, John Wiley & Sons
2) Piskunov, N., Differential and Integral calculus, Mir publishers Moscow (Vol. 1, Vol. 2)
3) Thomas, G.B. and Finney, R.L, Calculus and Analytic Geometry, Addison Wesley Longman
4) Michael D. Greenberg, Advanced Engineering Mathematics, Pearson Education Pvt. Ltd
5) Jain R.K., Iyengar S.R.K, Advanced Engineering Mathematics, Narosa Publishers

Course title:	**ELECTRICAL ENGINEERING**	**Sub code**	ES102			
		Structure	**L**	**T**	**P**	**C**
			2	0	2	3

Course Objective:	• To enable the students, understand the basic ideas and principles of Electrical Engineering. • To impart knowledge for understanding the details of electrical powersystems, transformers, generators, motors etc.
Course Outcome:	• Acquire the knowledge about circuit analysis by applying KVL KCL and network Theorems. • Analysis of Single Phase AC Circuits, the representation of alternating quantities, and determining the power in these circuits • Understand the different methods for measurement of various electrical quantities. • Acquire knowledge about the constructional details, losses, parameters, and principles of operation of Transformers. • Acquire the knowledge of fundamentals, construction details, working characteristics, and classification of DC motor and induction motor.

Content

Module 1: Electrical Circuit:

DC circuits-Ohm's & Kirchoff's laws, mesh and nodal analysis, circuit theorems; Electro-magnetism, Faraday's & Lenz's laws, induced EMF and its uses; Network Theorems: Superposition Theorem, Thevenin's Theorem, Norton's Theorem, Maximum Power Transfer Theorem Voltage Source (Definition, Characteristics of Practical Source, and Equivalent Current Source), and Star-Delta Transformation.
Magnetic Circuit, Flux, MMF, Reluctance, Analogy with Electric Circuits. Simple Calculations for Composite Magnetic Circuits

Module 2: AC Circuits:

Periodic Function, Average & R.M.S., Values, Steady State Behaviour with Sinusoidal Excitation, Phasor Representation, Reactance & Impedance, Series & Parallel Circuit, Power Factor, Principle of Generation of Single Phase & Three Phase Voltages, Power in Balanced Three Phase AC System

Module 3: DC Machine:

DC Machines covering, working principle of DC machine as a generator and a motor; Types and constructional features; EMF equation of generator, relation between EMF induced and terminal voltage enumerating the brush drop and drop due to armature reaction; DC motor working principle; Back EMF and its significance, torque equation; Types of D.C. motors, characteristics and applications; Necessity of a starter for DC motor;

Module 4: Transformers:

Introduction, Basic Principles, Construction, Phasor Diagram for Transformer under No Load Condition Transformer on Load, Balance of MMF on Sides, Phasor Diagram, Equivalent Circuit, Open Circuit & Short Circuit Test, Voltage Regulation and Efficiency

Module 5: Power Systems& Electrical Machines:

Elementary Idea about Power Generation, Transmission and Distribution. Sources of Electrical Power covering, Introduction to Wind, Solar, Fuel cell, Tidal, Geothermal, Hydroelectric, Thermal-steam, diesel, gas, nuclear power plants; Concept of cogeneration, and distributed generation.

Reference Books:

1) Hughes, Electrical Technology, Pearson Publishers
2) Theraja B.L., Electrical Technology, S. Chand Publishers
3) Kothari D.P. and NagrathI.J., Theoryand Problems of Basic Electrical Engineering, PrenticeHall India
4) Kulshresta D.C., Basic Electrical Engineering, TMH India
5) Mittle and Mittal, Basic Electrical Engineering, TMH, 2005
6) Tarnekar S.G., and Kharbanda P.K., „A Textbook of Laboratory Course in ElectricalEngineering", Chand S., 2006 (For practical)

List of Experiments:

1) Study and verification of Kirchhoff's laws applied to DC circuits.
2) Verification of Thevenin's Theorem.
3) To Verify Maximum Power Transfer theorem.
4) Study of AC series R-L-C circuit.
5) Determination of B-H curve of a magnetic material.
6) Study of AC parallel R-L-C circuits.
7) Study of balanced 3-phase circuits.
8) Determination of voltage regulation and efficiency of a single-phase transformer by directloading.
9) Study of speed control of a DC motor by field current control and by armature voltage control.
10) Study of reversal of direction of rotation of a 3-phase induction motor.

Course Title:	Python Programming	Sub code:	PC101			
		Structure:	L	T	P	C
			2	0	4	4

Course Objective:	• To equip students with the necessary programming skills and proficiency in Python to solve computational problems and develop software applications effectively. • To enhance students' problem-solving abilities by applying Python programming concepts to analyze, design, and implement solutions across various domains including engineering, science, and data analytics.
Course Outcome:	• Understand Python syntax and use of Python flow control and Functions. Develop, run and manipulate program by using core data structures like lists, sets, dictionaries, tuples and use of strings handling methods. Develop, run and manipulate python programs using file operations. • Able to understand Data Wrangling. Manipulate one-dimensional and multi-dimensional Numpy arrays, pandas series and data frames. Perform Data Wrangling, data loading, cleaning, transformation, reshaping, pivoting and merging. • Able to understand Data Aggregation, Group Operations, Time series and various python web scrapping. Design the web Application with the help of python programming for IOT application. • Explore the use Matplot lib package for Data Visualization in python and implement the different techniques for plotting graphs. Comprehend the importance of the exploratory data analysis paradigm, Select appropriate data visualization technique for given data. • Explore the use of python programing for IOT system. Installing OS and Designing Systems using Raspberry pi. Implement Various IoT system using Python Programming and raspberry pi.

Content

Module 1: Introduction to Python

Installing Python. How a Program Works, Using Python, Program Development Cycle, Input, Processing, and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations, Operators. Type conversions, Expressions, More about Data Output. Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables. Repetition Structures: Introduction, while loop, for loop, Calculating a Running Total, Input Validation Loops, Nested Loops. Data types and Expressions: Strings, Assignment and Comments, Numeric Data Types and Character Sets, Expressions, Functions and Modules.

Module 2: Control statements

Definite Iteration, Formatting Text for Output, Selection, Conditional Iteration. File and Exceptions: Introduction to File Input and Output, Using Loops to Process Files, Processing Records, Exceptions. Functions: Introduction, Defining and Calling a Void Function, Designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning Functions-Generating Random Numbers, The math Module, Storing Functions in Modules

Module 3: Strings and Text Files:

Accessing Characters and Substrings in a String, Strings and Number System, String Methods, Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings. Text Files, Data Encryption, Lists, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List Methods and Useful Built-in Functions, Copying Lists, Processing Lists, Two-Dimensional Lists, Tuples Sequences, Tuples. Dictionaries and Sets: Dictionaries, Sets, Serializing Objects. Recursion: Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms

Module 4: – Design with classes

Classes and Objects, Classes and Functions, Classes and Methods, Working with Instances, Inheritance and Polymorphism. Object-Oriented Programming: Procedural and Object-Oriented Programming, Classes, techniques for Designing Classes.

Module 5: Graphical User Interface

Behavior of terminal based programs and GUI-based programs, Coding simple GUI-based programs, other useful GUI resources. GUI Programming: Graphical User Interfaces, Using the tkinter Module, Display text with Label Widgets, Organizing Widgets with Frames, Button Widgets and Info Dialog Boxes, Getting Input with Entry Widget, Using Labels

as Output Fields, Radio Buttons, Check Buttons. Simple Graphics and Image Processing: Overview of Turtle Graphics, Two dimensional Shapes, Colors and RBG System, Image Processing.

Text/Reference Books:

1. Mark Lutz, "Learning Python", O'Reilly Media, 5th Edition, 2016.
2. White, "Hadoop: The Definitive Guide", Third Edition - O'Reilly, 2012.
3. Brandon Rhodes and John Goerzen, "Foundations of Python Network Programming: The Comprehensive Guide to Building Network Applications with Python", Apress, Second Edition, 2016.

List of Experiments:

1. Installation of Python, and learning interactively at command prompt and writing simple programs.
2. Learning the conditions and iterations in Python by writing and running simple programs.
3. Random number generations, and problems based on random numbers.
4. Handling tuples and exercises based on tuples.
5. Functions and files
6. Linear and binary search
7. Handling tokens
8. Finding unique and duplicate items of a list.
9. Matrix addition, multiplications, and unity matrix.
10. Text processing using python 11. Programs related to python libraries like Numpy, Pandas, Scipy etc

Course title:	**ENGINEERING CHEMISTRY**	**Sub code:**	**BS103**			
		Structure:	**L**	**T**	**P**	**C**
			2	1	0	3

Course Objective:

- To present sound knowledge of chemistry fundamentals, enriching students tounderstand the role of Applied Chemistry in the field of science and engineering. To inculcate habit of scientific reasoning to do the task rationally.
- To introduce the students to basic principles of electrochemistry, cell construction and evaluation, electrochemical power sources, the importance of corrosion in metal/alloy and polymer.

Course Outcome:

- Understand the fundamental principles of chemistry and their relevance to engineering applications.
- Apply chemical concepts to analyze and design engineering materials and processes.
- Gain knowledge of corrosion mechanisms and methods for prevention in engineering materials.
- Explore environmental chemistry and its implications for sustainable engineering practices.
- Acquire laboratory skills for conducting chemical experiments and analyzing results in engineering contexts.

Content

Module 1: Atomic and Molecular Structure

Schrodinger equation. Particle in a box solutions and their applications for conjugated molecules and nanoparticles. Forms of the hydrogen atom wave functions and the plots of these functions to explore their spatial variations. Molecular orbitals of diatomic molecules and plots of the multicentre orbitals. Equations for atomic and molecular orbitals. Energy level diagrams of diatomic. Pi-molecular orbitals of butadiene and benzene and aromaticity. Crystal field theory and the energy level diagrams for transition metal ions and their magnetic properties. Band structure of solids and the role of doping on band structures.

Module 2: Electrochemistry

Conductivity of electrolytes- Specific, molar and equivalent conductivity, Nernst equation for electrode potential, EMF series, hydrogen electrode, calomel electrode, glass electrode, Electrolytic and galvanic cells, cell EMF, its measurement and applications, Weston standard cell, reversible and irreversible cells, concentration cell, electrode (hydrogen gas electrode) and electrolyte concentration cell, concentration cell with and without transference.

Module 3: Intermolecular forces and potential energy surfaces

Ionic, dipolar and van Der Waals interactions. Equations of state of real gases and critical phenomena. Potential energy surfaces of H3, H2F and HCN and trajectories on these surfaces.

Module 4: Periodic Properties

Effective nuclear charge, penetration of orbitals, variations of s, p, d and f orbital energies of atoms in the periodic table, electronic configurations, atomic and ionic sizes, ionization energies, electron affinity and electronegativity, polarizability, oxidation states, coordination numbers and geometries, hard soft acids and bases, molecular geometries.

Module 5: Solid State

Types of solids - close packing of atoms and ions - bcc , fcc structures of rock salt - cesium chloride- spinel -normal and inverse spinel's, Stoichiometric Defect, controlled valency & Chalcogen semiconductors, Non-elemental semiconducting Materials, Preparation of Semiconductors-steps followed during the preparation of highly pure materials and further treatments. Semiconductor Devices-p-n junction diode.

Module 6: Polymer

Nomenclature, functionality, classification, methods of polymerization, mechanism of polymerization, molecular weight determination-Viscometry, light scattering methods. Plastics-Moulding constituents of a plastics and moulding of plastics into articles. Important thermoplastics and thermosetting resins- synthesis & applications of PVA, FLUON, PC, Kevlar, ABS polymer, phenolic & amino resins, epoxy resins and polyurethanes. Conductive polymers.

Reference Books

1. P. C. Jain and M. Jain, Engineering Chemistry, Dhanpat Rai Publishing Company, New Delhi, 2005.
2. B.R. Puri, L.R. Sharma, M.S. Pathania, Principles of Physical Chemistry, Vishal Publishing Company, 2008.
3. J. D. Lee, Concise Inorganic Chemistry, 5th Edn., Chapman and Hall, London, 1996.
4. S. S. Dara, S. S. Umare, A Text Book of Engineering Chemistry, S. Chand Publishing, 2011.
5. F.W. Billmayer. Textbook of Polymer Science, 3rd Edn, Wiley. N.Y. 1991.
6. A.R. West, Basic Solid State Chemistry, 2nd edition, John Wiley and Sons, 1999.

Course title:	**ENGINEERING MATHEMATICS-II**	**Sub code:**	**BS201**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**0**	**3**

Course Objective: To provide students with a fundamental understanding of importance of multi variable calculus (Differential calculus & Integral calculus), Vector calculus and ordinary differential equations in engineering.

Course Outcome:

- Understand advanced calculus concepts and their applications in engineering.
- Master probability and statistics principles relevant to engineering analysis.
- Learn differential equations of engineering importance and their solutions.
- Apply mathematical tools to model and analyze complex engineering systems.

Content

Module 1: Calculus of Functions of Several Variables

Limit, continuity and differentiability of functions of several variables, partial derivatives and their geometrical interpretation, Tangent plane and normal line. Euler's theorem on homogeneous, functions, Total differentiation, chain rules, Jacobian, Taylor's formula, maxima and minima, Lagrange's method of undetermined multipliers.

Module 2: Multiple Integrals
Double and triple integrals, change of order of integration, change of variables, application to area,volumes, Mass, Centre of gravity.
Module 3: Vector Calculus
Scalar and vector fields, gradient of scalar point function, directional derivatives, divergence and curl of vector point function, solenoidal and irrotational motion. Vector integration: line, surface and volume integrals, Green's theorem, Stroke's theorem and Gauss divergence theorem (without proof).
Module 4: Ordinary Differential Equations
First order differential equations: Exact equation, Integrating factors, Reducible to exact differential equations, Linear and Bernoulli's form, orthogonal trajectories, Existence and Uniqueness of solutions. Picard's theorem, Picard's iteration method of solution (Statements only). Solutions of second and higher order linear equation with constant coefficients, Linear independence and dependence, Method of variation of parameters, Solution of Cauchy's equation, simultaneous linear equations.
Module 5: Complex Variable-Differentiation
Differentiation, Cauchy-Riemann equations, analytic functions, harmonic functions, finding harmonic conjugate; elementary analytic functions (exponential, trigonometric, logarithm) and their properties; Conformal mappings, Mobius transformations and their properties.
Reference Books

1. Kreyszig, E., Advanced Engineering Mathematics, John Wiley & Sons
2. Piskunov, N., Differential and Integral calculus, Mir publishers Moscow (Vol. 1,Vol. 2)
3. Thomas, G.B. and Finney, R.L, Calculus and Analytic Geometry, AddisonWesley Longman.
4. Michael D. Greenberg, Advanced Engineering Mathematics, Pearson EducationPvt. Ltd
5. Jain R.K., Iyengar S.R.K, Advanced Engineering Mathematics, Narosa Publishers.

Course title:	**Sports and Yoga**	**Sub code:**	**HS201**			
		Structure:	**L**	**T**	**P**	**C**
			0	**1**	**4**	**3**

Course Objective:	• To maintain students' mental and physical wellness upright and develop ability in them to cope up with the stress arising in the life. • To create space in the curriculum to nurture the potential of the students in sports/games/yoga etc. • To introduce a practice oriented introductory course on the subject. More involved / advanced course may come up in subsequent years of study.
Course Outcome:	• Enhance physical fitness, stamina, and overall well-being through regular participation in sports activities. • Develop teamwork, leadership, and communication skills through team sports and group exercises. • Cultivate mindfulness, stress management, and relaxation techniques through the practice of yoga. • Promote a healthy lifestyle and balance between academic studies and physical activity. • Understand the importance of sports and yoga in maintaining mental and physical health for optimal performance in academic and professional pursuits.

Content
Module 1: Introduction to Sports Science and Yoga
Overview of sports science and its applications in engineering, Introduction to yoga, its history, and benefits, Basic principles of sports training and conditioning, Introduction to various yoga asanas (poses) and their significance
Module 2: Anatomy and Physiology in Sports and Yoga
Understanding human anatomy and physiology related to sports performance, Effects of exercise and yoga on various systems of the body, Biomechanics of movement in sports and yoga, Yoga anatomy: Understanding the alignment and engagement of muscles in asanas
Module 3: Sports Training and Techniques
Principles of sports training: Strength, endurance, speed, and flexibility, Training methodologies and periodization in sports, Techniques for enhancing athletic performance, Introduction to sports-specific drills and exercises, Application of yoga asanas for improving athletic performance and preventing injuries
Module 4: Sports Psychology and Mindfulness
Understanding the psychological aspects of sports performance, Mental preparation and goal setting in sports, Stress management techniques for athletes, Introduction to mindfulness and its application in sports and engineering, Incorporating mindfulness practices into daily routine for improved focus and concentration
Module 5: Applied Sports Science and Yoga in Engineering
Integration of sports science principles in engineering design, Biomechanical analysis of sports equipment and technologies, Case studies on the application of sports science in engineering projects, Designing ergonomic workspaces and equipment for athletes and engineers, Practical sessions combining sports activities, yoga, and engineering projects

Reference Books	
	1. "Introduction to Sports Science" by Robert Weinberg and Daniel Gould 2. "Essentials of Strength Training and Conditioning" by NSCA - National Strength & Conditioning Association 3. "Biomechanics of Sport and Exercise" by Peter McGinnis 4. "Yoga Anatomy" by Leslie Kaminoff and Amy Matthews 5. "Anatomy and Physiology for Health Professionals" by Jahangir Moini 6. "Sports Training Principles" by Frank Dick 7. "The Psychology of Enhancing Human Performance" by Frank L. Gardner and Zella E. Moore 8. "Mind Gym: An Athlete's Guide to Inner Excellence" by Gary Mack 9. "Mindfulness in Plain English" by Bhante Henepola Gunaratana 10. "Engineering Biomechanics: Mechanics and Design Applications in Musculoskeletal Systems" by D. Gordon E. Robertson and Graham E. Caldwell
list of Experiments	
	11. Hands-on practice sessions focusing on various yoga asanas such as Sun Salutation, Warrior poses, and balancing poses. 12. Emphasis on correct alignment, breath awareness, and relaxation techniques. 13. Practical training sessions in different sports disciplines like cricket, football, volleyball, or athletics. 14. Focus on improving specific skills such as batting techniques, dribbling skills, or throwing techniques. 15. Conducting fitness assessments including measurements of flexibility, strength, endurance, and cardiovascular fitness using standardized protocols. 16. Analysis of fitness test results to identify areas for improvement and design personalized fitness programs. 17. Interactive sessions exploring the philosophical principles of yoga such as the Eight Limbs of Yoga, Karma Yoga, and Bhakti Yoga. 18. Guided meditation practices to cultivate mindfulness, concentration, and inner peace. 19. Practical sessions on sports nutrition focusing on meal planning, hydration strategies, and nutrient timing for optimal performance and recovery. 20. Hands-on experience in preparing nutritious meals and snacks tailored to athletes' dietary needs.

Course title:	Electronics Devices and Circuits	Sub code:	PC201			
		Structure:	L	T	P	C
			2	0	2	3

Course Objective:	• This course aims to provide students with a foundational understanding of basic semiconductor devices, including their operating principles, characteristics, and diverse applications in electronic circuits and systems. • Students will gain proficiency in analyzing and designing simple diode circuits, enabling them to comprehend the behavior of diodes in various configurations and apply this knowledge to practical circuit design scenarios. • Through this course, students will develop a comprehensive understanding of PN junction behavior within the context of circuit-level analysis. They will explore the pivotal role of PN junctions in the operation of diodes and other active semiconductor devices, facilitating deeper insights into electronic device behavior and performance.
Course Outcome:	• Understand the principles and operating characteristics of electronic devices such as diodes, transistors, and integrated circuits. • Analyze and design electronic circuits using various semiconductor devices for different applications. • Gain proficiency in device modeling, simulation, and characterization techniques. • Explore emerging electronic devices and their potential applications in modern technologies. • Develop troubleshooting skills for diagnosing and rectifying electronic device-related problems in circuits and systems.

Content

Module 1: Basic Understanding of Semiconductor Devices

Introduction to Quantum Theory of Solids: Basic principles of quantum mechanics, Schrodinger equation and its applications, Atoms and formation of energy bands, electrical conduction in solids, density of tates functions, bonding forces and energy bands in solids. Semiconductor in Equilibrium: charge carriers in semiconductors, carrier concentrations, dopant atoms and energy levels, intrinsic and extrinsic semiconductors; charge neutrality, Fermi energy level.

Module 2: Power devices & Switching Devices

Carrier Transport Phenomena: Carrier drift, diffusion, graded impurity distribution, Hall Effect, scattering in semiconductors, velocity- electric field relations, high field transport charge injection and quasi Fermi levels. Non-Equilibrium Excess Carriers in Semiconductors: Carrier generation and recombination, characteristics of excess carriers, excess carrier lifetime, introduction to surface effects.

Module 3: PN Junction diode and Optoelectronics devices

PN junction and hetero-structures: basic structure and principle of operation, pn junction under bias, junction capacitance, steady state conditions, transient and ac conditions, reverse bias breakdown, metal semiconductor junctions.Introduction to Optoelectronic Devices, Overview of optoelectronics ,Historical background, IMPORTANCE AND APPLICATIONS: photodiode, LED, Emerging optoelectronics Technology, Photonic Integrated Circuits.

Module 4: Bipolar junction transistor

Bipolar Junction Transistors: Fundamental operation, amplification with BJTs, generalized biasing and equivalent circuit models, non-ideal effects, switching.

Module 5: Field Effect Transistor

Field – Effect Transistors: Transistor operations. JFET, Metal Semiconductor FET, MISFET, MOSFET and their operations, device characteristics, non-ideal effects, CV characteristics, equivalent circuits, carbon nano tube FET and it's applicationHEMTS.

Reference Books

	1. Electronic devices and Circuit Theory", "R. Boylestad", "Pearson Education", 9thEdition 2. "Electron devices", "S. Poornachandra, Sasikala", "Scitech", 2nd Edition 3. "Electronic Devices and Circuits", "Millman Halkias", "TMH", 2000 4. "Electronic Devices and Circuits", "DavidA.Bell", "PHI", 4thEdition
List of Experiments	
	5. **Characterization of Semiconductor Materials:** Perform experiments to understand the electrical properties of semiconductor materials such as silicon and germanium. Measure parameters like resistivity, mobility, and carrier concentration. 6. **PN Junction Diode Characteristics:** Study the I-V characteristics of a PN junction diode under forward and reverse bias conditions. Determine parameters like threshold voltage, forward and reverse bias currents, and ideality factor. 7. **Diode Rectifier Circuits:** Construct and analyze various diode rectifier circuits such as half-wave, full-wave bridge, and center-tapped full-wave rectifiers. Measure output voltage, ripple factor, and efficiency. 8. **Zener Diode Characteristics:** Investigate the voltage-regulating properties of Zener diodes. Measure the breakdown voltage and dynamic resistance of Zener diodes under different load conditions. 9. **Bipolar Junction Transistor (BJT) Characteristics:** Study the DC and AC characteristics of NPN and PNP bipolar junction transistors. Measure parameters like DC current gain (β), collector current vs. collector-emitter voltage (IC- VCE) characteristics, and output characteristics. 10. **BJT Amplifier Circuits:** Design and analyze common-emitter and common-base amplifier circuits using bipolar junction transistors. Measure parameters like voltage gain, input/output impedance, and frequency response. 11. **Field Effect Transistor (FET) Characteristics:** Investigate the DC and AC characteristics of both JFET and MOSFET transistors. Measure parameters like drain current vs. drain-source voltage (ID-VDS), transconductance, and output conductance. 12. **FET Amplifier Circuits:** Design and analyze common-source and common-drain amplifier circuits using field-effect transistors. Measure parameters like voltage gain, input/output impedance, and frequency response. 13. **Power Devices & Switching Devices:** Experiment with power semiconductor devices such as thyristors (SCRs), power MOSFETs, and IGBTs. Analyze their switching characteristics, turn-on and turn-off times, and power handling capabilities. 14. **Optoelectronic Devices:** Study the characteristics and applications of optoelectronic devices such as light-emitting diodes (LEDs), photodiodes, and phototransistors. Measure parameters like emission wavelength, forward voltage drop, and responsivity.

Course title:	**WORKSHOP TECHNOLOGY**	**Sub code:**	**ES201**			
		Structure:	**L**	**T**	**P**	**C**
			0	**1**	**2**	**2**
Course Objective:	To develop the technical skills of creating entities from raw materials. To give "hands on" training and practice to students for use of various tools, devices, equipment and machines. To develop ability to understand, plan and implement various processes and operations to be performed on the raw material to create object of desired shape and size.					
Course Outcome:	• Acquire practical skills in using workshop tools and machinery for basic metalworking processes. • Understand safety protocols and practices essential for working in workshop environments. • Learn fabrication techniques including welding, machining, and casting for manufacturing components. • Gain knowledge of different materials, their properties, and selection criteria for engineering applications. • Develop problem-solving abilities through hands-on experience in assembling and disassembling mechanical components and systems.					

Content
Module 1: Fitting
Use and setting of fitting tools for chipping, cutting, filing, marking, centre punching, drilling, tapping.Term work to include one job involving following operations: filing to size, drilling and tapping.
Module 2: Carpentry
Use and setting of hand tools like hacksaws, jack planes, chisels and gauges for construction of various joints, wood tuning and modern wood turning methods. Term work to include one carpentry job involving ajoint and report on demonstration of a job involving wood tuning.
Module 3: Electrical and Electronics
Introduction to basic electrical devices and its measurement. Introduction to basic knowledge of electronics component.
Module 4: Welding
Use and setting of tools and equipment's for edge preparation for welding jobs and Arc welding fordifferent job like, Lap welding of two plates, butt welding of plates.
Module 5: Machining, CNC Machines & Foundry.
At least one metal tuning job is to be demonstrated. One job on CNC Lathe and CNC Milling machine to bedemonstrated. At least one demonstration of mould making.
Reference Books
1. S K Hajra, CHoudhury, A K Hajra, CHoudhury, & Nirjhar Roy, Elements ofWorkshop Technology, Vol. I & II. 2. B S Raghuwanshi, A Course in Workshop Technology, Vol. 1 & II. 3. W A .l Chapman, Workshop Technology, Part I, ll & III

list of Experiments
1. Wood sizing exercise in planning, marking, sawing, chiselling and grooving to make 1. Half lap joint 2. Cross lap joint
2. Exercise in arc welding for making 1. Lap joint 2. Butt joint
3. Preparation of sand mould for the following 1. Flange 2. Anvil
4. Preparation of joints, markings, cutting and filling for making 1. V-joint 2. T-joint
5. Making of small parts using sheet metal 1. Tray 2. Funnel

Course title:	COMMUNICATION SKILLS	Sub code:	HS202			
		Structure:	L	T	P	C
			2	1	1	3

Course Objective:	The main aim of the course is to build competence in English grammar and vocabulary and to enhance effective communication by developing Reading, Writing, Listening and Speaking skills of students. Furthermore to develop the level of competence in English required for independent and effective communication for academic and social needs..
Course Outcome:	**Course outcomes:** After completion of course, students would be able to: • Understand various technical writing skills • Apply the technical writing and communication skills in their academic and professional life. • Gain self-confidence with improved command over English. • Understand the technical aspects of communication for better performance in extra-curricular activities, recruitment process and prospective jobs. • Students will be perform better in the Job Interviews.

Content

Module 1: Fundamentals of Communication Skills

Scope and Significance of Communication Skills, Listening, Speaking, Reading and Writing, Technical Communication, Tools of Effective Communication.

Module 2: Writing Skills

Basics of Grammar – Placing of Subject and Verb, Parts of Speech, Uses of Tenses, Active Passive, Narration.

Module 3: Vocabulary Building and Writing

Word Formation & Synonyms, Antonyms, Words Often Confused, One-Word Substitutes, Idioms and Phrasal Verbs, Abbreviations of Scientific and Technical Words.

Module 4: Speaking and Technical Writing Skills

Introduction to Phonetic Sounds & Articulation, Word Accent, Interpersonal Communication, Body Language and Voice Modulation (Para linguistics and Non- Verbal), Negotiation and Persuasion, Group Discussion, Interview Techniques (Telephonic and Video Conferencing).

Writing Skills : Job Application, CV Writing, Business Letters, Report Writing & Structure, E-mail Etiquette, Blog Writing.

Module 5: Engineering Ethics

What is profession? Engineering and Professionalism, Ethics , Morality ,Types of Ethics and Morality ,Engineering Ethics, Responsibility in Engineering, Engineering Standards, The Standard Care, The Positive face and the Negative Face of Engineering Ethics.

Reference Books	
	1. The Essence of Effective Communication", Ludlow R. and Panton F., Pubs: Prentice Hall, 1992 2. "Effective Communication Skills", Kulbhushan Kumar, Khanna Publishing House, 2019. 3. "A University Grammar of English", Quirk R. and Sidney G., 3rd Edition, Pubs: Pearson Education, 2008 4. "High School English Grammar", Wren and Martin, Pubs: S. Chand & Company Ltd, 2007 5. "Essentials of Business Communication", Guffrey M.E., 8th Edition, Pubs: South-Western College Publishing, 2009 6. "Technical Communication: Principles and Practice", Raman M. and Sharma S., 2nd Edition, Pubs: Oxford University Press, 2012 7. "Effective Business Communication", Rodrigues M.V., Pubs: Concept Publishing Company, Delhi, 2003 8. "English Vocabulary in Use", McCarthy M. and Felicity O' Dell, 2nd Edition, Pubs:2010 9. Mike Martin and Roland Schinzinger, "Ethics in Engineering" McGraw Hill
List of Experiments	
	1. Introducing Oneself, Exercise on Parts of Speech & Exercise on Tense. 2. Exercise on Agreement, Narration, Active Passive Voice & Dialogue Conversation. 3. Exercise on Writing Skills and Listening Comprehension (Audio CD). 4. Organize group discussion sessions on engineering-related topics to improve students' ability to express their ideas articulately and persuasively. 5. Assign students to prepare and deliver technical presentations on engineering topics relevant to their specialization. 6. Conduct mock job interviews to help students develop effective communication skills for professional settings.

Course title:	**Design Thinking**	**Sub code:**	**HS203**			
		Structure:	**L**	**T**	**P**	**C**
			0	**0**	**2**	**1**

COURSE OBJECTIVE(S):

The objective of this Course is to provide the new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products which useful for a student in preparing for an engineering career.

Course Outcomes (CO):

Student will able to

1. Compare and classify the various learning styles and memory techniques and Apply them in their engineering education
2. Analyze emotional experience and Inspect emotional expressions to better understand users while designing innovative products
3. Develop new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products
4. Propose real-time innovative engineering product designs and Choose appropriate frameworks, strategies, techniques during prototype development
5. Perceive individual differences and its impact on everyday decisions and further Create a better customer experience

Module 1: Foundations of Learning and Memory

An Insight to Learning

- Understanding the Learning Process
- Kolb's Learning Styles
- Assessing and Interpreting

Remembering Memory

- Understanding the Memory Process
- Problems in Retention
- Memory Enhancement Techniques

Module 2: Emotional Intelligence and Individual Differences

Emotions: Experience & Expression

- Understanding Emotions: Experience & Expression
- Assessing Empathy
- Application with Peers

Celebrating the Difference

- Understanding Individual Differences & Uniqueness
- Group Discussion and Activities to Encourage the Understanding, Acceptance, and Appreciation of Individual Differences

Module 3: Introduction to Design Thinking and Creativity

Basics of Design Thinking

- Definition of Design Thinking
- Need for Design Thinking
- Objective of Design Thinking
- Concepts & Brainstorming
- Stages of Design Thinking Process (explain with examples):
    - Empathize
    - Define
    - Ideate
    - Prototype
    - Test

Being Ingenious & Fixing Problem

- Understanding Creative Thinking Process
- Understanding Problem Solving
- Testing Creative Problem Solving

Module 4: Product Design and Prototyping

Process of Product Design

- Process of Engineering Product Design
- Design Thinking Approach
- Stages of Product Design
- Examples of Best Product Designs and Functions
- Assignment – Engineering Product Design

Prototyping & Testing

- What is Prototype?
- Why Prototype?
- Rapid Prototype Development Process
- Testing
- Sample Example
- Test Group Marketing

Module 5: Customer-Centric Innovation and Final Integration

Design Thinking & Customer Centricity

- Practical Examples of Customer Challenges
- Use of Design Thinking to Enhance Customer Experience
- Parameters of Product Experience
- Alignment of Customer Expectations with Product Design

Feedback, Re-Design & Re-Create

- Feedback Loop
- Focus on User Experience
- Address Ergonomic Challenges
- User Focused Design
- Rapid Prototyping & Testing
- Final Product
- Final Presentation – "Solving Practical Engineering Problem through Innovative Product Design & Creative Solution"

List of Practicals

1. Learning Style Assessment using Kolb's Model
2. Memory Retention & Enhancement Exercise
3. Emotion Mapping and Empathy Test
4. Design Thinking Case Study Analysis
5. Group Brainstorming for Creative Problem Solving
6. Mini Product Design Challenge
7. Low-Fidelity Prototyping Workshop
8. Rapid Testing and Feedback Collection
9. Activity on Understanding Individual Differences
10. Customer Persona Development and Journey Mapping

Course title:	**ENGINEERING MATHEMATICS - III**	**Sub code:**				
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**0**	**3**
Course Objective:	● To Explain the Importance of Numerical Methods in Solving Differential Equations, Integrations, and Algebraic Equations ● To Explain the Significance of Laplace and Z Transforms and Their Use in Solving Differential and Difference Equations. ● To Apply Different Statistical and Curve Fitting Techniques to Gain Insights from Data. ● To Describe the Theory of Complex Variables					

Course Outcome:	• Master advanced mathematical techniques including vector calculus and complex analysis for engineering applications. • Apply Laplace transforms and Numerical methods to solve differential equations arising in engineering problems. • Develop proficiency in numerical methods for solving Complex Variable Integration problems. • Explore probability and statistics concepts relevant to engineering decision-making and data analysis. • Utilize mathematical modeling to analyze and solve real-world engineering problems in various disciplines.

Content	**No. of hours**	**ESE Marks (%)**
Module 1: Numerical Method	8	20
Lagrange's interpolation formula, Numerical differentiation, solution of ordinary differential equation byPicard's method, Taylor 's series method, Euler's modified method and Runge- Kutta method.		
Module 2: Laplace Transform	8	20
Existence theorem - Transform of standard functions – Transform of Unit step function and Dirac delta function – Basic properties - Shifting theorems - Transforms of derivatives and integrals – Transform of periodic functions - Initial and Final value theorem - Inverse Laplace - Convolution theorem– Solving Initial value problems by using Laplace Transform techniques. Solution of LDE by Laplace Transform.		
Module 3: Fourier Series	8	20
Dirichlet's conditions – General Fourier series – Odd and even functions – Half-range Sine and Cosine series – Complex form of Fourier series – Parseval's identity – Harmonic Analysis.		
Module 4: Fourier Transforms	8	20
Fourier integral theorem – Fourier transform pair - Fourier sine and cosine transforms – Properties – Transform of elementary functions - Convolution theorem – Parsevals's identity.		
Module 5: Z transform and difference equations	8	20
Z-transform – Elementary properties – Inverse Z-transform – Convolution theorem – Initial and final value theorems – Formation of difference equation – Solution of difference equation using Z – transform.		

Reference Books

1. G. Balaji, Transforms and Partial Differential Equations, McGraw Hill Education.
2. B.S.Grewal, Higher Engineering Mathematics, Khanna Publicatons, New Delhi
3. Erwin Kreyszing, Advanced Engineering Mathematics, Willey Eastern Ltd.
4. Steven C. Chapra, Raymond P. Canale, Numerical Methods for Engineers, McGraw Hill Education.

Course title:	**C PROGRAMMING**	**Sub code**	PC301			
		Structure	**L**	**T**	**P**	**C**
			2	0	4	4

Course Objective:	• To introduce basics of programming and develop logical thinking of students. • To help students understand how to model real world problems into the software and develop practical programming skills of students. • To implement mathematical statistical, applications into programming.
Course Outcome:	• Master the syntax, semantics, and basic programming constructs of the C languages. • Develop problem-solving skills through algorithmic thinking and programming exercises. • Understand memory management, pointers, and data structures for efficient programming. • Gain proficiency in modular programming, debugging, and testing techniques.

Content

Module 1: Introduction:

Flow charts, data types and storage classes, scope of variables, arithmetic operators, assignment, conditional, arithmetic expressions, enumerated data types, decision making, branching, looping, Switch concept, function and parameter passing, recursive functions, macros.

Module 2: Basic programming algorithms:

Programs to illustrate basic language constructs in C like - Factorial, Sine/cosine and other mathematical series, Fibonacci series, calculating square-root of a number, calculating GCD of 2 integers (Euclid's method and otherwise), Calculating LCM of 2 integers and similar such programs.

Module 3: Arrays and applications in C language:

Introduction to one dimensional and 2-D array with examples. Representing a polynomial using 1-D array and polynomial operations, use of 2-D array to represent a matrix and matrix operations. Character arrays (strings): String related functions (strlen, strcpy, strcat, strcmp, atoi, itoa, reverse, strstr etc) and their function definitions. Searching and Sorting methods: Selection sort, Bubble sort,Insertion sort, Linear and binary search, partitioning an array, merging of 2 sorted arrays. Introduction to "Divide and Conquer" via Mergesort and Quicksort.

Module 4: Structures, Unions and Pointers in C language:

Basic concept, array of structures and its applications. Introduction (declaration and initialization), pointers and arrays, concept of dynamic memory allocation, use of pointers to represent variable- sized 1-D and 2-D arrays, pointers to structures.

Module 5: Advanced Topics in C Programming

Introduction to file operations, opening and closing files, reading from and writing to files, file modes, formatted and unformatted file I/O. Bitwise AND, OR, XOR, NOT, shifting operations, applications of bitwise operators in programming. Use of #include, #define, macros with arguments, conditional compilation, and use of predefined macros. malloc, calloc, realloc, and free; managing dynamic memory in complex data structures. Introduction to linked lists (singly and doubly linked lists), stacks, queues, and basic operations using pointers. Error handling techniques using return codes, errno, and error reporting. Implementing small projects or case studies that integrate the learned concepts, such as a basic text editor or a simple file management system.

Reference Books:

1) Kerninghan; Ritchie, "C programming Language", PHI
2) Theraja B.L., Electrical Technology, S. Chand Publishers
3) Balguruswamy, "Programming in ANSI C", Tata Mcgraw Hill Publishing
4) Kakde and Deshpande, "C and data Structure", Charles River Media Publisher
5) Dromey R G, "How to Solve it by Computer", PHI

List of Experiments:

1) Write a Program to calculate and display the volume of a CUBE having its height (h=10cm),width (w=12cm) and depth (8cm).
2) Write a program to take input of name, roll no and marks obtained by a student in 5 subjectseach have its 100 full marks and display the name, roll no with percentage score secured.
3) Simple Arithmetic Operation.
4) Write a C program to check whether a number is even or odd using ternary operator.
5) Write a C program to find the sum of individual digits of a positive integer.
6) Write a C program to print the numbers in triangular form.
7) Write a C program to find the second largest integer in a list of integers.
8) Write C programs that use both recursive and non-recursive functions.
9) Write a C program to perform arithmetic using Switch Statement.
10) Write a C program to perform factorial.
11) Write a C program to print Fibonacci no.

Course title:	Data Structure and Algorithms	Sub code:	PC302			
		Structure:	L	T	P	C
			2	0	4	4

Course Objective:	• Master fundamental data structures and algorithms to efficiently solve computational problems. • Develop critical thinking and problem-solving skills through the analysis and implementation of algorithms. • Understand the principles behind various data structures and their applications in real-world scenarios. • Enhance coding proficiency by practicing algorithmic problem-solving techniques..
Course Outcome:	• Apply various data structures and algorithms to efficiently solve computational problems. • Analyze the time and space complexity of algorithms to make informed design choices. • Design and implement algorithms for sorting, searching, and manipulating data effectively. • Utilize data structures such as arrays, linked lists, trees, and graphs to model and solve real-world problems.

Content

Module 1: Introduction and basic terminology

Notion of data structures and algorithms, $lllllll$, nn, $2nn$: understanding growth of these functions, and applications (binary search and extensions to similar problems), Worst-case, averagecase time/space complexity and their relative merits, Asymptotic Notation: $OO( )$,$\Omega\Omega( )$

Module 2: Abstract Data-types, Arrays, Linked Lists, Stacks, Queues Dictionary ADT, Trees, Binary Trees

Abstract data-type (ADTs): arrays and linked list ADTs, Stacks, Queues: ADTs and implementations using arrays, linked lists, Doubly linked lists: ADT and implementation, Dictionary ADT: implementation using array, linked lists, binary search, Tree ADT and examples, Implementation of trees and basic traversal algorithms, Binary trees and in order traversal

Module 3: Priority Queues and Heaps

Priority Queue ADT, Definition of heaps, Implementation of Priority Queues using heaps and running time analysis, Implementation of heaps using arrays, Heap-sort

Module 4: Binary Search Trees, AVL Trees, 2-4 trees

Binary Search Trees: definition and some basic algorithms, Implementation of Dictionary ADTs using Binary Search trees and running time analysis, AVL trees: height balance condition, rotations, and implementation of dictionary ADT, 2-4 Trees: Multi-way search trees, implementation of dictionary ADT, Informal discussion of extension to B-trees.

Module 5: **Hash tables, tries**

Map ADT , Hash Tables and implementation of Map using Hash Tables , Design of hash functions , Collision resolution schemes: chaining, open addressing schemes like linear probing, quadratic probing, double hashing, Applications of Hashing: finding duplicates, set intersection, etc. ,Tries: implementation of Map ADT using tries,Compressed tries and suffix tries.

Reference Books

	1	"Data Structures and Algorithms in Java", by Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons; 3rd Edition.
	2	"Data Structures and Algorithms in Python", by Michael T. Goodrich and Robert, Tamassia, Wiley, 1st Edition.
	3	In case any other programming language is used for this course, some other suitable text book may be chosen.
	4	NPTEL video series, Data-structures and Algorithms, Instructor: Naveen Garg
	5	Introduction to Algorithms, 4th Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.

Experiment List

	1. Measure the time complexity of linear search versus binary search on arrays of varying sizes to understand their growth patterns. 2. Implement a stack using arrays and compare its performance with a linked list-based implementation. 3. Explore the efficiency of different collision resolution schemes (chaining, linear probing, quadratic probing) in hash tables. 4. Construct a priority queue using a heap and analyze its time complexity during various operations. 5. Implement a binary search tree and measure its efficiency in comparison to other dictionary ADTs. 6. Study the balance conditions in AVL trees by implementing insertion and deletion operations and analyzing their effects on the tree's height. 7. Compare the efficiency of different traversal algorithms (in-order, pre-order, post-order) on binary trees. 8. Experiment with various hash functions and observe their impact on hash table performance. 9. Implement a trie data structure and explore its efficiency in storing and retrieving data compared to hash tables. 10. Construct a 2-4 tree and analyze its space and time complexity for dictionary operations.

Course title: Object Oriented Programming using Java	**Subject code: PC303**	**L**	**T**	**P**	**C**
		3	**0**	**2**	**4**

Course Objective:	• To introduce the principles of object-oriented programming (OOP) and familiarize students with Java's syntax and features. • To develop the ability to design modular and reusable code using OOP concepts like classes, inheritance, polymorphism, encapsulation, and abstraction. • To provide hands-on experience with Java's standard libraries, APIs, and the Collections Framework for effective data management and manipulation. • To equip students with knowledge of exception handling, multithreading, and file handling to build robust and efficient applications. • To enable students to create graphical user interfaces (GUIs) and understand event-driven programming to enhance user interaction in applications.
Course Outcome:	Upon completing the course, students will be able to: • Apply object-oriented programming concepts to design and develop real-world Java applications. • Implement inheritance, polymorphism, abstraction, and encapsulation effectively to build modular and maintainable code. • Utilize Java's standard libraries and collections framework for data storage, retrieval, and manipulation in various scenarios. • Develop multithreaded applications to perform tasks concurrently. • Design and implement graphical user interfaces using Java AWT, Swing, and JavaFX to create interactive applications.

Content

Module 1: Introduction to Object-Oriented Programming and Java Basics

Understanding the fundamentals of OOP concepts - classes, objects, inheritance, polymorphism, encapsulation, and abstraction. Overview of the Java programming language, setting up the Java environment (JDK, JRE), compiling and running Java programs. Java syntax and basics including data types, variables, operators, and control structures (if-else, switch, loops). Creating classes and objects, constructors, use of `this` keyword, static members. Introduction to Java development tools, basics of IDEs, Java compiler, code editors, debugging tools, and basic command-line operations.

Module 2: Object-Oriented Concepts and Exception Handling

Types of inheritance, `super` keyword, method overriding, `final` keyword, and inheritance hierarchies. Understanding polymorphism, compile-time vs. runtime polymorphism, method overloading, and dynamic method dispatch. Abstract classes and interfaces, implementing interfaces, handling multiple inheritance in Java through interfaces. Concepts of encapsulation, access modifiers, and data hiding. Exception handling basics, types of exceptions, `try`, `catch`, `finally`, `throw`, and `throws` keywords, creating custom exceptions, and best practices for error handling.

Module 3: Java API and Packages, Collections Framework

Introduction to Java Standard Library and commonly used packages (`java.lang`, `java.util`, `java.io`, etc.). Working with `String` and `StringBuilder` classes, wrapper classes, auto-boxing, and unboxing. Introduction to the

Collections Framework, working with Lists, Sets, Maps, and their implementations (ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap). Iterators and enhanced for loop, understanding generics in Java, using and creating generic classes and methods, and an introduction to lambda expressions.
Module 4: File I/O, Multithreading, and Networking in Java
Understanding file handling in Java, working with File I/O (InputStream, OutputStream, Reader, Writer classes), reading and writing text and binary files. Basics of multithreading, creating threads by extending `Thread` class and implementing `Runnable` interface, thread lifecycle, synchronization, inter-thread communication, `wait`, `notify`, and `notifyAll` methods.
Module 5: GUI Development, Applets, and Event Handling
Basics of GUI programming using the AWT (Abstract Window Toolkit) and Swing libraries, creating windows, panels, buttons, and other GUI components. Event-driven programming in Java, handling events through event listeners and adapters. Introduction to applets, lifecycle of an applet, creating and embedding applets in web pages.

Reference Books	
	• Herbert Schildt, Java: The Complete Reference, 10th Edition, McGraw Hill. • Kathy Sierra and Bert Bates, Head First Java, 2nd Edition, O'Reilly Media. • Paul Deitel and Harvey Deitel, Java: How to Program, 10th Edition, Pearson Education. • Cay S. Horstmann, Core Java Volume I – Fundamentals, 11th Edition, Pearson. • Bruce Eckel, Thinking in Java, 4th Edition, Prentice Hall.
Experiment list	
1. Basic Java Program: Write a Java program to illustrate the use of basic operators, control statements, and simple I/O operations. 2. Classes and Objects: Implement a class to model a real-world object (e.g., Student, Bank Account) with properties and methods. 3. Inheritance and Polymorphism: Demonstrate inheritance with a superclass and subclass hierarchy, applying polymorphism with method overriding. 4. Abstract Classes and Interfaces: Create an abstract class and an interface to define a blueprint for subclasses to implement. 5. Exception Handling: Write a program to demonstrate the handling of checked and unchecked exceptions. 6. String Manipulation: Develop a program that uses String and StringBuilder classes for various string operations. 7. File Handling: Create a Java application to read from and write to a file, handling I/O exceptions appropriately. 8. Multithreading: Implement a Java program with multiple threads to perform tasks concurrently, demonstrating thread synchronization. 9. Collection Framework: Write a Java program to demonstrate the use of List, Set, and Map collections, implementing basic operations. 10. GUI Development: Design a simple GUI calculator using Java Swing or JavaFX, handling events such as button clicks for different operations.	

Course title:	**Digital Logic and Circuits**	**Sub code:**				
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**2**	**4**
Course Objective:	• Introduce number systems, binary codes, and basic logic gates. • Enable students to simplify Boolean expressions using algebraic methods and Karnaugh Maps. • Teach the design and analysis of combinational logic circuits. • Introduce various types of flip-flops, counters, and shift registers in sequential logic. • Familiarize students with memory elements, logic families, and programmable logic devices					

Course Outcome:	**CO1:** Perform conversions between number systems and design logic functions using basic and universal gates. **CO2:** Simplify Boolean expressions using Karnaugh Maps and Boolean algebra for circuit minimization. **CO3:** Design and analyze combinational logic circuits like adders, subtractors, multiplexers, and decoders. **CO4:** Understand and implement sequential circuits such as flip-flops, counters, and shift registers. **CO5:** Understand the architecture and applications of memory units and programmable logic devices.

Content	**No. of hours**	**ESE Marks(%)**
Module 1: Number System, Codes and Boolean Algebra	8	20

Number systems and their inter-conversion, Binary Arithmetic (Addition, Subtraction, Multiplication and Division), BCD codes, Excess-3 code, Gray code, Hamming code, error detection and correction. Boolean Algebra, Basic Theorems and properties of Boolean Algebra, Boolean Functions, Canonical and Standard forms-map method, Two, Three, Four and Five variable K-maps, Sum of products and Product of Sums Simplification. The tabulation method, Determination of Prime implicants, Selection of Essential Prime implicants.
Digital Logic Gates, Various Logic Families - DTL, TTL and CMOS.

Module 2: Introduction to Verilog HDL	8	20

Verilog as HDL, Levels of Design Description, Concurrency, Simulation and Functional Verification. Language Constructs and Conventions: Introduction, Keywords, Identifiers, White Space Characters, Comments, Numbers, Strings, Logic Values, Strengths, Data Types, Scalars and Vectors, Parameters, Operators. Test benches.
Modelling and simulation of logic gates and Boolean functions using Verilog HDL. Functional verification using test benches.

Module 3: Combinational Logic Design	8	20

NAND and NOR implementation of basic gates and Boolean functions. Binary adder and sub tractor, Multiplexers and De-Multiplexers, Decoders and Encoders. Carry Lookahead adder. BCD to 7 segment decoders. An 8 bit-ALU design.
Modelling and simulation of combination circuits using Verilog HDL. Writing test cases, Functional verification using test benches. Direct and random verification.

Module 4: Sequential Logic Design	8	20

Introduction, S-R Flip-flops, JK flip-flop, D flip-flop, T flip-flop, master slave flip-flop. Flip-flop excitation table, Interconversion of flip-flop, Register and Counter circuits. Finite state machines-Mealy and Moore FSMs. FSM encoding techniques-binary, Gray and One hot.

Verilog RTL Coding. Modelling and simulation of sequential circuits using Verilog HDL. Writing test cases, Functional verification using test benches. Direct and random verification.

Module 5: Programmable Logic Devices-Architecture and Programming	8	20

PAL, PLA, PLD, CPLD and FPGA. LUT and CLB architecture. FPGA Architecture-Fine grained and coarse-grained. Verilog RTL coding for synthesis. FPGA Design Flow.
Overview of FPGA development tools. Case study-Xilinc-Kintex-7 FPGA Family. Implementation of logic circuits on FPGA. In system Debugging using ILA.

Reference Books

1. **M. Morris Mano, Michael D. Ciletti** – *Digital Design*, Pearson Education
2. **R.P. Jain** – *Modern Digital Electronics*, McGraw-Hill Education
3. **Thomas L. Floyd** – *Digital Fundamentals*, Pearson Education
4. **John F. Wakerly** – *Digital Design: Principles and Practices*, Pearson
5. **Tokheim** – *Digital Electronics: Principles and Applications*, McGraw-Hill Education

list of Experiments

1. **Verification of Logic Gates:** Implement and verify the truth tables of AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.
2. **Universal Gates Implementation:** Realize basic logic functions using only NAND or NOR gates.
3. **Boolean Expression Simplification:** Simplify given Boolean expressions and implement the minimized circuits using logic gates.
4. **Design of Half Adder and Full Adder:** Construct and test circuits for half and full adders using logic gates.
5. **Design of Half Subtractor and Full Subtractor:** Implement and verify half and full subtractor circuits.
6. **Multiplexer and Demultiplexer:** Design and test 4:1 MUX and 1:4 DEMUX circuits.
7. **Flip-Flop Implementations:** Implement and observe SR, D, T, and JK flip-flops using basic gates or ICs.
8. **Counters:** Design and test asynchronous and synchronous up/down counters (e.g., mod-8, mod-10).
9. **Shift Registers:** Construct and analyze SIPO, SISO, PIPO, and PISO shift registers.
10. **BCD to 7-Segment Display:** Design a decoder to display BCD inputs on a 7-segment LED display.
11. Design and Simulation of 8-bit ALU in Verilog
12. CLA implementation using Verilog.
13. Finite State Machine (FSM) – Design of Sequence Detector using Verilog.
14. Design and Synthesis of Combinational Circuit on FPGA Board (e.g., Basys 3 / Kintex-7)
15. FPGA-based 4-bit Counter with LED Display and On-Chip Debugging using ILA
16. Verilog Implementation of a Small FSM Synthesized on Xilinx FPGA.

Course title:	**Constitution of India/ Essence of Indian Knowledge Tradition**	**Sub code:**	**HS301**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**0**	**3**

Course Objective:	• To provide students with a comprehensive understanding of the historical context and processes involved in the making of the Indian Constitution. • To examine and analyze the significance and interpretation of the Preamble and the Basic Structure of the Indian Constitution. • To familiarize students with the fundamental rights and duties enshrined in the Indian Constitution and their interpretation in contemporary contexts. • To explore the principles of state policy and their implications for governance and development in India.
Course Outcome:	• Demonstrate a thorough understanding of the historical events and key figures involved in the making of the Indian Constitution. • Analyze and evaluate the significance of the Preamble and the Basic Structure doctrine in shaping Indian constitutional law. • Apply knowledge of fundamental rights and duties to analyze contemporary socio-political issues and debates. • Evaluate the role and powers of various organs of the Indian government, including the President, Prime Minister, and Parliament, in the functioning of the Indian Union. • Assess the functioning of local administrations, including the roles of governors, chief ministers, and local governing bodies, in implementing governance at the grassroots level.

Content

Module 1: The Constitution - Introduction

The History of the Making of the Indian Constitution, Preamble and the Basic Structure, and its interpretation , Fundamental Rights and Duties and their interpretation , State Policy Principles.

Module 2: Union Government		
Structure of the Indian Union, President – Role and Power, Prime Minister and Council of Ministers Lok Sabha and Rajya Sabha		
Module 3: Local Administration		
Governor – Role and Power, Chief Minister and Council of Ministers, State Secretariat		
Module 4: Local Administration		
District Administration, Municipal Corporation, Zila Panchayat		
Module 5: Election Commission		
Role and Functioning, Chief Election Commissioner, State Election Commission		
Reference Books		
	1	Ethics and Politics of the Indian Constitution by Rajeev Bhargava Oxford University Press, New Delhi, 2008
	2	The Constitution of India by B.L. Fadia Sahitya Bhawan; New edition (2017)
	3	Introduction to the Constitution of India by DD Basu Lexis Nexis; Twenty-Third 2018 edition

Course Title:	**Computer Architecture and Organization**	**Sub code:**		**PC401**		
		Structure:	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Course Objective:	• This course aims to familiarize students with fundamental concepts related to computer organization, covering hardware and software aspects such as CPU architecture, memory organization, input/output systems, and operating system functionality. • Students will gain an overview of the design principles underlying digital computing systems, including topics such as instruction set architecture, processor microarchitecture, pipelining, memory hierarchy, and input/output interfaces. • Through theoretical study and practical exercises, students will learn how data is represented at the machine level, including binary representation of numbers, character encoding schemes, floating-point representation, and memory organization, enabling them to understand how information is processed within a computer system. • Students will gain insight into the execution of computations at the machine level, including arithmetic and logic operations, control flow mechanisms, instruction execution cycle, and addressing modes, enabling them to comprehend the underlying mechanisms of program execution and data manipulation within a computer system					
Course Outcome:	• Understand the fundamental principles and components of computer architecture including CPU, memory, and I/O devices. • Analyze and design the organization of computer systems at both hardware and software levels. • Gain proficiency in understanding the instruction set architecture (ISA) and its implications on system design. • Explore advanced topics such as pipelining, memory hierarchy, and parallel processing architectures. • Develop practical skills in designing and evaluating computer systems through simulation, prototyping, and performance analysis.					
Content						
Module 1: Introduction of Processor						

Introduction, Technologies for building Processors and Memory, Performance, The Power Wall, Operations of the Computer Hardware, Operands Signed and Unsigned numbers, Representing Instructions, Logical Operations, Instructions for Making Decisions

Module 2: Instructions Set

MIPS Addressing for 32-Bit Immediate and Addresses, Parallelism and Instructions: Synchronization,Translating and Starting a Program, Addition and Subtraction, Multiplication, Division, Floating Point, Parallelism and Computer Arithmetic: Sub word Parallelism, Streaming SIMD Extensions and Advanced Vector Extensions in x86.

Module 3: Architecture Building Block

Logic Design Conventions, building a Datapath, A Simple Implementation Scheme, overview of Pipelining, Pipelined Datapath, Data Hazards: Forwarding versus Stalling, Control Hazards, Exceptions, Parallelism via Instructions, The ARM Cortex – A8 and Intel Core i7 Pipelines, Instruction –Level Parallelism and Matrix Multiply Hardware Design language.

Module 4: Memory Mapping

Memory Technologies, Basics of Caches, Measuring and Improving Cache Performance, dependable memory hierarchy, Virtual Machines, Virtual Memory, Using FSM to Control a Simple Cache, Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks, Advanced Material: Implementing Cache Controllers.

Module 5: Memory Management

Disk Storage and Dependability, RAID levels, performance of storage systems, Introduction to multi-threading clusters, message passing multiprocessors.

Text/Reference Books:

1. David A. Patterson and John L. Hennessey, "Computer organization and design, The Hardware/Software interface", Morgan Kauffman / Elsevier, Fifth edition, 2014
2. V. Carl Hamacher, Zvonko G. Varanesic, and Safat G. Zaky, "Computer Organization", 6 th edition, McGraw-Hill Inc, 2012.
3. William Stallings, "Computer Organization and Architecture", 8th Edition, PearsonEducation, 2010

List of Experiments

List of 10 Practical Experiments

Module 1 & 2: Processor Basics and Instruction Set

1. **Simulation of Basic Arithmetic Operations in MIPS Assembly**
   - Objective: Write MIPS assembly code to perform addition, subtraction, multiplication, and division of two integers.
   - Tool: MIPS Simulator (e.g., QtSpim, MARS)
2. **Binary and Hexadecimal Conversion & Representation**
   - Objective: Implement a program (in C or Assembly) to convert between binary, hexadecimal, and decimal formats; demonstrate signed and unsigned number representations.
3. **Instruction Encoding and Decoding**
   - Objective: Manually encode and decode sample MIPS instructions and verify using a simulator.

Module 3: Datapath and Pipelining

4. **Design a Basic Arithmetic Logic Unit (ALU) Using VHDL/Verilog**
   - Objective: Implement an ALU capable of performing AND, OR, ADD, SUB, and SLT operations.
   - Tool: Xilinx Vivado / ModelSim / Logisim
5. **Simulate Single-Cycle and Multi-Cycle Datapaths**
   - Objective: Model a simplified processor datapath (single-cycle and pipelined) and compare performance.
   - Tool: Digital Simulator / Logisim / Ripes (RISC-V simulator)
6. **Pipeline Hazard Detection and Resolution**

	o Objective: Simulate and analyze data and control hazards in a pipelined processor; demonstrate stalling and forwarding. Module 4 & 5: Memory Mapping and Management 7. **Cache Memory Simulation** o Objective: Simulate direct-mapped, fully-associative, and set-associative caches; compute hit/miss ratio. o Tool: Custom Python/C program or existing cache simulator tools. 8. **Implementation of Virtual Memory using Paging** o Objective: Simulate address translation using page tables; demonstrate page faults and TLBs. o Tool: Python or Java-based simulation 9. **Disk Scheduling Algorithm Simulation** o Objective: Implement and compare performance of FCFS, SSTF, SCAN, and LOOK disk scheduling algorithms. 10. **RAID Level Simulation and Performance Comparison** • Objective: Simulate different RAID levels (RAID 0, 1, 5) and evaluate redundancy, speed, and fault tolerance.

Course title:	**Discrete Mathematics**	**Sub code:**	**PC402**			
		Structure:	**L**	**T**	**P**	**C**
			3	**1**	**0**	**4**

Course Objective:	• Develop a foundational understanding of discrete structures and their applications in computer science and mathematics. • Master techniques for solving combinatorial problems and analyzing discrete structures such as graphs and trees. • Gain proficiency in logical reasoning and proof techniques essential for tackling discrete mathematical problems. • Explore the connections between discrete mathematics and real-world applications in fields like cryptography, computer algorithms, and network analysis.
Course Outcome:	• Demonstrate the ability to apply discrete mathematical concepts to solve problems in various domains, including computer science, cryptography, and optimization. • Construct and analyze mathematical arguments and proofs involving discrete structures such as sets, graphs, and Boolean algebra. • Utilize discrete mathematics tools and techniques to design and analyze algorithms for solving computational problems efficiently. • Recognize the relevance of discrete mathematics in modeling real-world phenomena and making informed decisions in technological and scientific contexts.

Content
Module 1: Reasoning
Mathematical reasoning; propositions; negation disjunction and conjunction; implication and equivalence; truth tables; predicates; quantifiers; natural deduction; rules of Inference; methods of proofs; use in program proving; resolution principle
Module 2: Set Theory
Set theory; Paradoxes in set theory; inductive definition of sets and proof by induction; Peono postulates; Relations; representation of relations by graphs; properties of relations, equivalence relations and partitions; Partial orderings; Posets; Linear and well-ordered sets
Module 3: Graph Theory
Graph Theory; elements of graph theory, Euler graph, Hamiltonian path, trees, tree traversals, spanning trees.

Module 4: Functions		
Functions; mappings; injection and surjections; composition of functions; inverse functions; special functions; Peono postulates; pigeonhole principle; recursive function theory.		
Module 5: **Groups**		
Definition and elementary properties of groups, semigroups, monoids, rings, fields, vector spaces and lattices. Elementary combinatorics; counting techniques; recurrence relation; generating functions.		
Reference Books		
	1	Liu, C. L., & Mohapatra, D. P. (2008). Elements of Discrete Mathematics. Tata McGraw-Hill.
	2	Rosen, K. H. (2019). Discrete Mathematics and Its Applications. (8th Edition) ISBN10: 125967651X ISBN13: 9781259676512.
	3	Huth, M., & Ryan, M. (2004). Logic in Computer Science: Modelling and Reasoning about Systems (2nd ed.). Cambridge University Press.
	4	Cohen, D. I. A. (1978). Basic techniques of combinatorial theory. John Wiley
	5	Niven, I., Zuckerman, H. S., & Montgomery, H. L. (1991). An introduction to the theory of numbers. John Wiley & Sons.
	6	Shoup, V. (2009). A computational introduction to number theory and algebra. Cambridge University Press.
	7	Herstein, I. N. (2006). Topics in algebra. John Wiley & Sons
	8	C. J. Date (2019). Database Design and Relational Theory. Normal Forms and All That Jazz.

Course title:	**Design and Analysis of Algorithms**	**Sub code:**	**PC403**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**2**	**4**

Course Objective:	• Develop a comprehensive understanding of fundamental algorithm design techniques and their applications in solving complex computational problems. • Master the art of analyzing algorithmic efficiency through rigorous mathematical analysis and empirical testing. • Acquire proficiency in implementing algorithms using appropriate data structures and programming languages.
Course Outcome:	• Apply algorithm design paradigms such as divide and conquer, dynamic programming, and greedy algorithms to solve a variety of computational problems efficiently. • Analyze the time and space complexity of algorithms using mathematical methods and asymptotic notation. • Implement and evaluate algorithms using appropriate data structures and programming languages to address real-world challenges. • Demonstrate proficiency in designing algorithms for specific problem domains, including sorting, searching, graph traversal, and optimization. • Evaluate the correctness and efficiency of algorithms through rigorous testing, benchmarking, and performance analysis.

Content
Module 1: Applications of Graph Search
Review of BFS/DFS (ii) Checking if an undirected graph is 2-edge connected. (iii) Checking if a directed graph is strongly connected.
Module 2: Greedy algorithms

Introduction to the greedy paradigm (ii) Examples of activity selection, deadline scheduling, fractional knapsack, Kruskal's algorithm for minimum spanning trees, Huffman coding.

Module 3: Divide and Conquer

Explain why the divide and conquer paradigm is useful, Illustrate the paradigm through integer multiplication., Writing recurrence relations and solving them, Further examples from geometry – domination number of a set of points, identifying maximal points, closest pair of points, Linear time algorithm for finding the median, Randomized divide and conquer algorithms: randomized quicksort and selection.

Module 4: Dynamic Programming and shortest paths

Computing Fibonacci numbers and why divide-and conquer is not a good idea. Idea of storing function calls, tables, Notion of sub problems and optimal substructure, Illustration through subset sum, (integer) knapsack, longest increasing subsequence, longest common subsequence, matrix chain multiplication. Dijkstra's algorithm for single-source shortest paths, Bellman-Ford for SSSP with negative weights, Floyd Warshall for APSP.

Module 5: Network flows

The maximum s-t flow problem in capacitated networks. Ford Fulkerson algorithm or maximum flow. Max-flow min-cut theorem and integrality of maximum flow for integral capacities. Applications of max flow to maximum bipartite matching, max disjoint paths

Module 6: Intractability

Models of computation,Turing machines, RAM model. Brief discussion on other models of computation e.g. PRAM model, Memory Hierarchy etc.
Notion of polynomial time computation.
Polynomial time reductions.Yes and No instances of decision problems. Decision vs optimization.
NP as a class of problems with Yes certificates which can be efficiently checked. NP-hardness and Cook- Levin theorem (just the statement).
NP-completeness. Examples of Reductions.

Reference Books

	1	Algorithm Design, 1st Edition, Jon Kleinberg and Éva Tardos, Pearson
	2	Algorithms, Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani
	3	Introduction to Algorithms, 4th Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
	4	Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
	5	The Design and analysis of Algorithms Author A V Aho et al Publishe r Pearson Education
	6	Fundamental of Algorithms Author Brassard Bratley Publishe r PHI Edition 1st Edition 1996

list of Experiments

	1	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted. The elements can be read from a file or can be generated using the random number generator
	2	Implement a Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted .The elements can be read from a file or can be generated using the random number generator.
	3	A) Obtain the Topological ordering of vertices in a given digraph. B) Compute the transitive closure of a given directed graph using Warshall's algorithm
	4	Implement 0/1 Knapsack problem using Dynamic Programming
	5	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijikstra's algorithm
	6	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm
	7	A) Print all the nodes reachable from a given starting node in a digraph using BFS method. B) Check whether a given graph is connected or not using DFS method.

	8	Find a subset of a given set S = {s1, s2,….., sN} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.
	9	Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
	10	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.

Course title:	**Database Management Systems**	**Sub code:**	**PC404**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**2**	**4**

Course Objective:	• To understand the role and functions of a database management system and its impact on the overall performance of a computer system. • To understand the concepts and techniques involved in ER modeling. • To understand the SQL commands and relational algebraic expressions for query processing. • To gain hands-on experience with designing and implementing database management systems through programming projects and case studies.
Course Outcome:	• Learn the basic concepts of Database Systems • Model the real-world systems using Entity Relationship Diagrams and convert the ER model into a relational logical schema using various mapping algorithms • Make use of SQL commands and relational algebraic expressions for query processing • Simplify databases using normalization process based on identified keys and functional dependencies and solve the atomicity, consistency, isolation, durability, transaction, and concurrency related issues of databases

Content

Module 1: Introduction

- General introduction to database systems; Database - DBMS disctinction, approaches to building a database, data models, database management system, threeschema architecture of a database, challenges in building a DBMS, various components of a DBMS.

Module 2: Database design and ER Model

Overview, ER-Model, Constraints, ER-Diagrams, ERD Issues, weak entity sets, Codd's rules, Relational Schemas, Introduction to UML Relational database model: Logical view of data, keys, integrity rules. Relational Database design: features of good relational database design, atomic domain and Normalization (1NF, 2NF, 3NF, BCNF).

Module 3: Relational algebra

Introduction, Selection and projection, set operations, renaming, Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relational comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs algebra, computational capabilities.

Module 4: SQL

Introduction, data definition in SQL, table, and key and foreign key definitions, update behaviors. Querying in SQL - basic select-from-where block and its semantics, nested queries- correlated and uncorrelated, notion of aggregation, aggregation functions group by and having clauses, embedded SQL. Data Storage and Indexes - file organizations, primary, secondary index structures, various index structures - hashbased, dynamic hashing techniques, multi-level indexes, and B+ trees.

Module 5: Transaction management and Concurrency control

Transaction processing and Error recovery - concepts of transaction processing, ACID properties, and serializability concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, and database recovery management. Error recovery and logging, undo, redo, undo-redo logging and recovery methods.

Reference Books		
	1	Fundamentals of Database Systems Author R. Elmasri and S.B. Navathe Publisher Pearson Edition 2016
	2	Database Systems Concepts Author H.f.Korth and Silberschatz Publisher McGraw Hill
	3	Database System Author Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom Publisher Pearson Edition 2nd Edition
	4	Data Base Design Author C.J. Date Publisher Addison Wesley.
	5	DBM and Design Author Hansen and Hansen Publisher PHI
	6	"Database System Concepts" Authors: Abraham Silberschatz, Henry F. Korth, S. Sudarshan Publisher: McGraw-Hill Education Year: 2020 (7th edition).
list of Experiments		
	1	Library Management system (File Handling)
	2	Introduction to SQL ● Installation of SQL-Server ● SQL data definition ● Constraints in SQL ● Schema change Statement
	3	Basic SQL Queries
	4	Complex SQL Queries-1 ● Nested Queries ● Correlated Nested Queries ● EXISTS Function in SQL ● Aggregation Function
	5	Complex SQL Queries-2 ● Joined Tables ● Aggregate Functions
	6	Complex SQL Queries-3 ● Grouping ● EXISTS and UNIQUE functions ● Aggregate Functions
	7	Entity-Relationship Diagram from Case Study
	8	Normalization of the Case Study
	9	Webpage Connectivity with SQL Server Using XAMPP- 1
	10	Webpage Connectivity with SQL Server Using XAMPP- 2

Course Title:	**Operating Systems**	**Sub code:**	**PC501**			
		Structure:	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**

Course Objective:	• To understand the role and functions of an operating system and its impact on the overall performance of a computer system. • • To understand the concepts and techniques involved in process management, such as process creation, scheduling, interprocess communication, and synchronization. • To understand the concepts and techniques involved in memory management, such as virtual memory, swapping, paging, and segmentation. • To gain hands-on experience with the design and implementation of operating systems through programming projects and case studies.
Course Outcome:	• Understanding of the fundamental concepts, design principles, and implementation techniques of modern operating systems • Ability to design, implement, and evaluate process management, memory management, file system management, and input/output management algorithms • Ability to understand and implement distributed systems, such as client-server systems, distributed file systems, and distributed operating systems. • Hands-on experience with the design and implementation of operating systems through programming projects and case studies.

Content

Module 1: Introduction to Operating Systems

Operating System Functionalities, Types of Operating Systems, Computer Architecture support to Operating Systems.

Module 2: Process Management

Threads, Process Scheduling - Uniprocessor scheduling algorithms, Multiprocessor and Real-time scheduling algorithms, Process Synchronization - Peterson's Solution, Bakery. Algorithm, Hardware Support to Process Synchronization, Semaphores, Critical Regions, Monitors - Deadlock prevention, deadlock avoidance and Deadlock Detection and Recovery - Bankers Algorithm,.

Module 3: Memory Management

Segmentation and space allocation, Basics of linking and loading, Demand Paging, Page replacement algorithms, Analysis of page allocation policies, Thrashing- Working Set.

Module 4: File Systems

Contiguous, Sequential and Indexed Allocation, File system interface, File System implementation, Case study of Unix File system, Mounting and Unmounting files systems, Network File systems.

Module 5: I/O System

Disk Scheduling, Device drivers - block and character devices, streams, Character and Block device switch tables. Protection and Security - Accessibility and Capability Lists.

Text/Reference Books:

	1	Modern Operating Systems Author Andrew Tanenbaum Publisher Prentice Hall
	2	Operating Systems Author William Stallings Publisher Prentice Hall
	3	An introduction to operating systems Author Harvey M. Deitel Publisher Addison-Wesley
	4	Operating Systems: Design and Implementation Author Andrew Tanenbaum & Albert Woodhull Publisher Prentice-Hall
	5	Fundamentals of Operating Systems Author A.M. Lister Publisher Macmillan Edition 1979

List of Experiments

	1	Basic of Unix Commands
	2	Implementation of Process Related System Calls (Fork).
	3	Implementation of System Calls (Open, Read, Write and Close) for File Management
	4	Implementation of Process Synchronization
	5	Implementation of Memory Management Using Address Translation
	6	Implementation of FIFO Page Replacement Algorithms
	7	Implementation of LRU Page Replacement Algorithms
	8	Implementation of First Come First Serve and Shortest Job Fist Scheduling Algorithm
	9	Implementation of Priority and Round Robin CPU Scheduling Algorithm
	10	Implementation of Banker’s Algorithm.

	11	Implementation of Sleeping Barbar Problem in process synchronization
	12	Implementation of Algorithm for Deadlock Detection

Course Title:	**Machine Learning**	**Sub code:**	**PC502**			
		Structure:	**L**	**T**	**P**	**C**
			3	0	2	4

Course Objective:	• Develop a foundational understanding of machine learning algorithms and their applications across various domains • Gain proficiency in implementing and evaluating supervised, unsupervised, and reinforcement learning algorithms. • Explore advanced topics such as deep learning, natural language processing, and computer vision to solve complex problems • Acquire skills in data preprocessing, feature engineering, and model evaluation for effective machine learning pipeline development
Course Outcome:	• Ability to apply various machine learning algorithms to solve real-world problems and make data-driven decisions. • Competence in evaluating and selecting appropriate machine learning models based on dataset characteristics and problem requirements. • Proficiency in utilizing Python libraries such as scikit-learn and TensorFlow for implementing machine learning algorithms. • Understanding of ethical considerations and biases inherent in machine learning models, and ability to address them responsibly. • Capability to interpret and communicate the results of machine learning models effectively to stakeholders with varying levels of technical expertise □

Content

Module 1: Introduction to ML

Motivation and role of machine learning in computer science and problem olving, Representation (features), linear transformations, Appreciate linear transformations and matrix vector operations in the context of data and representation, Problem formulations (classification and regression),Appreciate the probability distributions in the context of data, Prior probabilities and Bayes Rule, Introduce paradigms of Learning (primarily supervised and unsupervised. Also a brief overview of others)

Module 2: Fundamentals of ML

PCA and Dimensionality Reduction, Nearest Neighbours and KNN, Linear Regression, Decision Tree Classifiers, Notion of Generalization and concern of Overfitting, Notion of Training, Validation and Testing; Connect to generalisation and overfitting.

Module 3: Selected Algorithms

Ensembling and RF (ii) Linear SVM, K Means, Logistic Regression , Naive Bayes

Module 4: Neural Network Learning

Role of Loss Functions and Optimization, Gradient Descent and Perceptron/Delta Learning, MLP, Backpropagation, MLP for Classification and Regression, Regularisation, Early Stopping, Introduction to Deep Learning ,CNNs

Module 5: Deep Learning Architectures

Popular CNN Architectures, RNNs, GANS and Generative Models.

Text/Reference Books :

	1	Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press (23 April 2020)

	2	Aurélien Géron Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, Inc. 2nd Edition.
	3	Tom M. Mitchell- Machine Learning - McGraw Hill Education, International Edition

List of Experiments	
	1. Implement linear transformations on datasets and visualize the effect on data representation. 2. Explore different probability distributions using real-world datasets and analyze their impact on classification and regression problems. 3. Apply Bayes' Rule to calculate posterior probabilities for classification tasks and assess its effectiveness. 4. Compare the performance of supervised and unsupervised learning algorithms on benchmark datasets to understand their strengths and limitations. 5. Perform Principal Component Analysis (PCA) on high-dimensional datasets to visualize dimensionality reduction. 6. Implement K-Nearest Neighbors (KNN) algorithm and evaluate its performance on various datasets. 7. Build and train linear regression models on different datasets to predict continuous variables. 8. Construct decision tree classifiers and analyze their decision boundaries and feature importance. 9. Investigate the concept of generalization and overfitting by experimenting with different model complexities and training/validation/testing datasets. 10. Develop and train neural network models, including Multi-Layer Perceptrons (MLP) and Convolutional Neural Networks (CNNs), for classification and regression tasks, and analyze their performance compared to traditional machine learning algorithms.

Course Title:	**Software Engineering**	**Sub code:**		**PC503**		
		Structure:	**L**	**T**	**P**	**C**
			3	**1**	**0**	**4**
Course Objective:	• Understand the principles and practices of software engineering to develop robust and scalable software solutions. • Gain proficiency in software development methodologies such as Agile, Scrum, and DevOps to improve team collaboration and project management. • Learn software design patterns and architectural principles to create maintainable and reusable codebases. • Acquire skills in requirements engineering, software testing, and quality assurance to ensure the reliability and usability of software products. • Explore emerging technologies and industry best practices to stay updated with advancements in the field of software engineering.					
Course Outcome:	• Ability to apply software engineering principles to design, develop, and maintain complex software systems. • Proficiency in using software development tools and techniques to effectively manage projects and teams. • Capability to analyze and solve software-related problems by applying appropriate methodologies and techniques. • Understanding of ethical and professional responsibilities in software engineering practice. • Aptitude for continuous learning and adaptation to evolving technologies and practices in the software engineering field.					

Content
Module 1: Introduction
Definition of software and Software engineering, Need of Software engineering, Difference between Program and Product, Software development life cycle, Different life cycle models (waterfall, Iterative waterfall, Prototype, Evolutionary, Incremental and Spiral model), Agile software development and their characteristics, V-Model. Critical Comparisons of SDLC models.
Module 2: Software Requirements
Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. System models: Context models, behavioral models, data models, object models, structured methods.
Module 3: Software Design
Goals of good software design, Design strategies and methodologies, Data oriented software design, Structured Design: Structure chart, Coupling, Cohesion, Modular structure, Packaging, Object oriented design, Topdown and bottom-up approach, Design patterns, Structured Analysis: DFD, Data Dictionary, Software Measurement and Metrics: Various Size Oriented Measures: Halstead's software science, Function Point (FP) based measures, Cyclomatic Complexity Measures: Control flow graphs. Development: Selecting a language, Coding guidelines, Writing code, Code documentation.
Module 4: Software Testing
Testing process, Design of test cases, Functional Testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path testing, Data flow and mutation testing, Unit testing, Integration and system testing, Debugging, Alpha & beta testing, testing tools & standards
Module 5: Software Maintenance
Management of maintenance, Maintenance process, Maintenance models, Regression testing, Reverse engineering, Software reengineering, Configuration management, documentation. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards. Current trends in Software Engineering: Software Engineering for projects and products. Introduction to Web Engineering and Agile process

Text/Reference Books:	
	Software Engineering, A practitioner's Approach Author Roger S. Pressman Publisher Mc Graw Hill International Edition. Edition 6th edition Software Engineering Author Sommerville Publisher Pearson Education Edition 7th edition Software Engineering principles and practice Author Waman S Jawadekar, Publisher The Mc Graw-Hill Companies. Edition 2000

Course title:	**Probability and Statistics**	**Sub code:**	**PC504**			
		Structure:	**L**	**T**	**P**	**C**
			3	**1**	**0**	**4**

Course Objective:	
	• To provide students with a strong foundation in probability theory and statistical methods applicable to computer science and engineering problems. • To familiarize students with essential concepts such as probability distributions, statistical inference, and Bayesian statistics relevant to data analysis and decision-making. • To equip students with practical skills in data analysis, hypothesis testing, and regression analysis for solving real-world problems in computer science and engineering. • To enable students to apply statistical techniques and tools effectively in designing algorithms, modeling systems, and making data-driven decisions in various domains.

Course Outcome:	• Apply fundamental principles of probability and statistics to analyze and interpret data sets commonly encountered in computer science and engineering. • Utilize statistical techniques to perform hypothesis testing, parameter estimation, and model validation for making informed decisions. • Implement and evaluate various machine learning algorithms using statistical methods for classification, regression, and clustering tasks. • Demonstrate proficiency in using statistical software tools and libraries to analyze large-scale datasets and extract meaningful insights. • Communicate effectively the results of statistical analyses and interpretations to technical and non-technical audiences through written reports and presentations.

Content

Module 1: Introduction to Probability

Basic concepts of probability: sample space, events, and probability axioms. Probability rules: addition rule, multiplication rule, and conditional probability. Discrete probability distributions: probability mass function, cumulative distribution function, and expected value. Continuous probability distributions: probability density function, cumulative distribution function, and expected value. Applications of probability in computer science and engineering.

Module 2: Statistical Methods

Descriptive statistics: measures of central tendency and dispersion. Probability distributions: binomial, Poisson, normal distributions, and their properties. Sampling distributions and central limit theorem. Point estimation: methods of estimation, maximum likelihood estimation, and properties of estimators. Interval estimation: confidence intervals for means and proportions. Introduction to hypothesis testing: null and alternative hypotheses, type I and type II errors, and significance level.

Module 3: Statistical Inference

Parametric tests: t-tests, chi-square tests, and F-tests. Non-parametric tests: Mann-Whitney U test, Wilcoxon signed-rank test, and Kruskal-Wallis test. Analysis of variance (ANOVA): one-way and two-way ANOVA. Linear regression analysis: simple linear regression and multiple linear regression. Model diagnostics and interpretation of regression results.

Module 4: Bayesian Statistics

Bayes' theorem and its applications. Bayesian inference: prior, likelihood, posterior, and Bayesian credible intervals. Bayesian estimation and hypothesis testing. Bayesian networks and their applications in machine learning and artificial intelligence.

Module 5: Applied Statistics in Computer Science

Data visualization techniques: histograms, box plots, scatter plots, and heatmaps. Exploratory data analysis (EDA): identifying patterns, outliers, and trends in data. Introduction to machine learning algorithms: linear regression, logistic regression, k-nearest neighbors, decision trees, and ensemble methods. Evaluation metrics for machine learning models: accuracy, precision, recall, F1-score, ROC curves, and AUC-ROC. Practical applications of statistics in computer science: data mining, pattern recognition, image processing, natural language processing, and bioinformatics.

Reference Books:

1. "Probability and Statistics for Computer Scientists" by Michael Baron
2. "Probability and Statistics with Applications: A Problem-Solving Text" by Leonard A. Asimow and Mark M. Maxwell
3. "Introduction to Probability and Statistics for Engineers and Scientists" by Sheldon M. Ross
4. "Statistical Inference: A Short Course" by David J. Olive
5. "Bayesian Data Analysis" by Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin

Course title:	**Theory of Computation**	**Sub code:**	**PC505**			
		Structure:	**L**	**T**	**P**	**C**
			3	**1**	**0**	**4**

Course Objective:	• Understand models and abstractions: automata as a basic model of computation • Link between languages, automata, and decision problems. • How to build new models from old ones: product, union, closure properties. • Argue about limitations of computational models. • Understand algebraic formalisms of languages such as regular expressions, context-free

	grammar. • Understand algorithms and computability through the lens of Turing machines. • Existence of unsolvable problems and what that means. • Relations between the various computational models.
Course Outcome:	• Proficiency in applying theoretical concepts to solve practical computing problems and design algorithms. • Mastery of formal language theory and automata theory concepts to analyze computational problems. • Ability to understand and prove properties of formal languages and automata. • Proficiency in designing and analyzing algorithms using formal computational models such as Turing machines and pushdown automata. • Understanding the limitations of computation, including topics like the halting problem and undecidability.

Content

Module 1: Introduction to Automata Theory

Overview of formal languages, grammars, and automata, Finite Automata (FA): Explanation of definitions, types, and properties. Regular Languages: Discussion on closure properties, regular expressions, and the conversion between Finite Automata and regular expressions.

Module 2: Context-Free Languages and Pushdown Automata

Context-Free Grammars (CFG): Explanation of definitions, Chomsky hierarchy, and normal forms. Pushdown Automata (PDA): Overview of definitions, types, and equivalence with CFG. Context-Free Languages: Discussion on closure properties, pumping lemma, and decision problems.

Module 3: Turing machines and Computability

Turing Machines (TM): Definition, types, and variations, Computability Theory: Halting problem, decidability, and undecidability. Church-Turing Thesis and Turing-recognizable languages.

Module 4: Complexity Theory

Time Complexity: Introduction to Big O notation, time complexity classes such as P, NP, NP-Complete, and discussions on reductions. Space Complexity: Explanation of space complexity classes, including Savitch's theorem and the space hierarchy theorem.NP-Completeness: Overview of the definition, Cook's theorem, and illustrative examples of NP-Complete problems.

Module 5: Advanced Topics in Theory of Computation

Formal Language Theory: Discussion on closure properties and the pumping lemma for context-free languages. Introduction to Computational Models: Overview of register machines, cellular automata, Recent Advancements and Applications in Theoretical Computer Science.

Reference Books:

1. Introduction to the Theory of Computation, 3rd edition. Michael Sipser, Cengage Publications (Lowcost Indian edition available).
2. Introduction to Automata, Theory, Languages and Computation. Third Edition. John Hopcroft, Rajeev Motwani, Jeffrey D. Ullmann, Pearson Publications (Low-cost Indian edition available).
3. Elements of the Theory of Computation, 2nd edition. Harry Lewis, Christos Papadimitriou, Prentice Hall.

Course title:	**Computer Network and Cyber Security**	**Sub code:**	**PC601**			
		Structure:	**L**	**T**	**P**	**C**
			2	**0**	**4**	**4**

Course Objective:	• This course aims to teach the basic concept of networks like network classification, network topologies network devices. • This course deal with the important concepts and techniques related to data communication and enable students to have an insight in to technology involved to make the network communication possible.

Course Outcome:	• Understand the fundamentals of computer networks, including protocols, architectures, and networking technologies. • Analyze and design network architectures for efficient data transmission, routing, and switching. • Gain proficiency in network security concepts, including cryptography, authentication, and access control. • Explore advanced topics such as Cyber security, network management, and cloud computing. • Develop practical skills in configuring and securing network devices, conducting vulnerability assessments, and implementing security measures to protect against cyber threats.

Content

Module 1: Introduction

Data Communications, Networks, The Internet, Protocols and Standards, Network Models, Layered Tasks, The OSI Model, Layers in the OSI Model, TCP/IP Protocol Suite, Addressing, Physical Layer and Media, Data and Signals, Analog and Digital, Periodic Analog Signals, Digital Signals, Transmission impairment, Data Rate Limits, Performance, Network topologies.

Module 2: Physical and Data Link Layer

Bandwidth utilization: Multiplexing and Spreading, Multiplexing, Spread Spectrum, Transmission Media, Guided Media, Unguided Media: Wireless, Switching, Circuit-Switched Networks, Datagram Networks, Virtual-Circuit Networks, Structure of a Switch, Introduction, Block Coding, Liner Block Codes, Cyclic Codes, Checksum, Data Link Control, Framing, Flow and Error Control, Protocols, Noiseless Channels, HDLC, Point-to-Point Protocol, Multiple Access, Random Access, Aloha, Controlled Access, Channelization, IEEE Standards, Standard Ethernet, Changes in the Standard, Fast Ethernet, Gigabit Ethernet, IEEE 802.11, Bluetooth, Connecting LANs, Backbone Networks, and Virtual LANs, Connecting Devices, Backbone Networks, Virtual LANs, Sonet Networks, Virtual Tributaries, Virtual- Circuit Networks: Frame Relay and ATM, Frame Relay, ATM, ATM LANs.

Module 3: Network Layer and Transport layer

Logical Addressing, IPv4 Addresses, IPv6 Addresses, Network Layer: Internet Protocol, Internetworking, IPv4, IPv6, Transition from IPv4 to IPv6, Network Layer: Address Mapping, Error Reporting and Multicasting, Address Mapping, ICMP, IGMP, ICMPv6, Network Layer: Delivery, Forwarding and Routing, Delivery, Forwarding, Unicast Routing Protocols, Multicast Routing Protocols. Process-Process Delivery: UDP, TCP and SCTP, Process-to-Process Delivery, User Datagram Protocol (UDP), TCP, SCTP, Congestion Control and Quality of Service, Data Traffic, Congestion, Congestion Control, Two Examples, Quality Service, Techniques to improve QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

Module 4: Application Layer

Domain Name System, Name Space, Domain Name Space, Distribution of Name Space, DNS in the Internet, Resolution, DNS Messages, Types of Records, Registrars, Dynamic Domain Name System (DDNS), Encapsulation, Remote Logging, Electronic Mail and File Transfer, Remote Logging, Telnet, Electronic Mail, File Transfer, WWW and HTTP: Architecture, Web Documents, HTTP, Network Management: SNMP, Network Management System, Simple Network Management Protocol (SNMP), Multimedia, Digitizing Audio and Video, Audio and Video Compression, Streaming Stored Audio/Video, Streaming Live Audio/Video, Real-Time Interactive Audio/Video, RTP, RTCP, Voice over IP.

Module 5: Cyber Security

Introduction to cyber threats, vulnerabilities, and principles.Overview of common cyber threats and threat actors. Introduction to network security technologies and tools. Basics of cryptography and secure communication protocols. Access control models and authentication methods. Incident response process and preparation. Importance of cybersecurity in computer networks and information systems. Recent cybersecurity incidents and their impact. Endpoint security solutions and endpoint detection. Introduction to security information and event management (SIEM) systems.

Reference Books:

	1. Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH. 2. Computer Networks, A.S. Tanenbaum,4th edition, Pearson education. 3. Introduction to Data communications and Networking, W. Tomasi, Pearson education. 4. Data and Computer Communications, G.S. Hura and M. Singhal, CRC Press, Taylor and Francis Group. 5. An Engineering Approach to Computer Networks-S. Keshav,2nd Edition, Pearson Education 6. Understanding communications and Networks,3rd Edition, W. A. Shay, Cengage Learning. 7. "Modern Cryptography, Theory & Practice", Pearson Education. Wenbo Mao 8. "Computer Security", Pearson Education. Matt Bishop
List of Experiments	
	1. **Experiment 1: Basic Client-Server Communication** Implement a simple TCP-based client-server program where the client sends a message to the server, and the server echoes it back. This introduces basic socket setup and message passing. 2. **Experiment 2: UDP Client-Server Communication** Develop a UDP-based client-server program for sending and receiving messages. Compare the differences in behavior between TCP and UDP in terms of connection, reliability, and data delivery. 3. **Experiment 3: File Transfer using Sockets** Write a client-server application where the client can upload a file to the server, and the server saves it. This experiment introduces file handling over sockets and data streaming. 4. **Experiment 4: Multi-Client Chat Application** Create a multi-client chat application using TCP sockets, where multiple clients can send messages to each other via a central server. The server should broadcast each message to all connected clients. 5. **Experiment 5: HTTP Client Implementation** Implement a simple HTTP client that connects to a web server, sends an HTTP GET request, and displays the response. This introduces concepts of HTTP protocols and headers. 6. **Experiment 6: Time Server and Client** Develop a client-server application where the client requests the current time from the server, and the server responds with the current system time. This experiment demonstrates request-response mechanisms. 7. **Experiment 7: Multi-threaded Server for Concurrent Clients** Design a multi-threaded server that can handle multiple clients concurrently. Each client should be able to connect to the server independently without blocking others. This introduces multi-threading with sockets. 8. **Experiment 8: Remote Command Execution** Implement a server that can execute basic commands sent by the client (like "ls" or "dir") and return the result. This experiment demonstrates how sockets can be used for remote command execution. 9. **Experiment 9: Simple File Transfer Protocol (FTP) Client** Create a simplified FTP client that connects to an FTP server, lists available files, and allows the client to download a file. This experiment introduces FTP commands and file transfer. 10. **Experiment 10: Encrypted Communication using Sockets** Implement a secure communication protocol using sockets where messages exchanged between client and server are encrypted (using simple symmetric encryption like XOR or AES). This introduces basic network security concepts.

Course title:	**Compiler Design**	**Sub code:**	**PC602**			
		Structure	**L**	**T**	**P**	**C**

	3	**0**	**2**	**4**

Course Objective:	• To understand the role, functionality and structure of program translation and interpretation in software development. • To understand the difference between abstraction levels of a high-level language and a machine language. • To understand the role of a sequence of intermediate representations in lowering the level of abstractions in the process of language translation. • To get a first-hand experience of a practical application of elegant data structures, algorithms, and other core CS concepts such as automata theory. • To make effective use of tools such as LEX and YACC. • To become a much better programmer by appreciating all that happens behind the scenes in making an HLL program run.
Course Outcome:	• Capacity to devise and execute lexical analyzers, parsers, and code generators to translate high-level programming languages into machine code. • Skillfulness in enhancing compiler performance by employing techniques like code optimization and register allocation. • Comprehensive comprehension of formal language theory and automata theory in the context of compiler design. • Competence in scrutinizing and rectifying compiler errors while implementing improvements to accommodate new language features or architectures. • Proficiency in collaborating on compiler development projects and actively participating in open-source compiler communities.

Content

Module 1: Introduction to Compiler Design

Overview of compilers and their role in software development Phases of a compiler: Lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization, and code generation. Compiler architecture and design principles

Module 2: Lexical Analysis

The role of lexical analysis; Token, lexemes, and token codes; Regular Expressions (RE) to represent tokens, Deterministic finite automata (DFA), Traversing a DFA for recognising tokens; Generating a lexical analyzer using LEX/Flex.

Module 3: Syntax Analysis

Context Free Grammars (CFG), Concept of parsing, sentences and sentential forms, leftmost and rightmost derivations, parse trees, ambiguous grammars; Overview of top-down and bottom-up parsing;

Module 4: Semantic Analysis and intermediate Code Generation

Semantic analysis: Type checking, symbol tables, and scope resolution. Intermediate representations: Three-address code, abstract syntax trees (AST). Translation of expressions, control structures, and declarations into intermediate code. Semantic errors detection and reporting.

Module 5: Code Optimization and Code Generation

Introduction to code optimization. Data-flow analysis and optimization techniques. Loop optimization, common subexpression elimination, and register allocation. Code generation: Target machine architecture, instruction selection, and scheduling. Optimization and code generation tools (e.g., LLVM). Performance evaluation and benchmarking of compilers

Reference Books:

1. "Compilers: Principles, Techniques, and Tools" by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman
2. "Engineering a Compiler" by Keith D. Cooper and Linda Torczon
3. "Modern Compiler Implementation in C" by Andrew W. Appel
4. "Introduction to the Theory of Computation" by Michael Sipser
5. "Advanced Compiler Design and Implementation" by Steven Muchnick

List of Experiments:

	1. Implement a lexical analyzer using regular expressions to recognize tokens in a programming language. 2. Develop a recursive descent parser for a simple grammar to understand syntax analysis. 3. Design and implement an LL(1) parser generator to handle predictive parsing. 4. Build a bottom-up parser (e.g., LR parser) for a given context-free grammar. 5. Create a symbol table module for semantic analysis, including type checking and scope resolution. 6. Generate intermediate code (e.g., three-address code) for a subset of a programming language. 7. Implement basic optimization techniques such as constant folding and dead code elimination. 8. Develop a code generator to translate intermediate code into assembly language for a hypothetical target machine. 9. Experiment with loop optimization strategies (e.g., loop unrolling) and measure their impact on code performance. 10. Explore the use of compiler optimization flags and analyze the resulting executable code size and execution time.

Course Title:	**Data Mining**	**Sub code:**	**PC603**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**2**	**4**

Course Objective:	• Understand the fundamental concepts, processes, and applications of data mining. • Develop proficiency in data preprocessing techniques for cleaning, integrating, transforming, and reducing data. • Gain practical skills in exploratory data analysis (EDA) methods to interpret dataset structures and characteristics. • Explore and implement various data mining algorithms and techniques for classification, clustering, association analysis, and anomaly detection.
Course Outcome:	• Demonstrate knowledge of data mining concepts, processes, and applications, and apply them to real-world datasets. • Implement data preprocessing techniques effectively to prepare data for analysis and modeling. • Utilize exploratory data analysis methods to gain insights into data distributions, patterns, and relationships. • Apply classification and prediction algorithms to solve classification and prediction tasks, and evaluate model performance using appropriate metrics. • Implement clustering and association analysis techniques to uncover hidden patterns and relationships in data, and evaluate their effectiveness using relevant evaluation measures.

Content

Module 1: Introduction to Data Mining

Overview of data mining concepts, process, and applications, Data preprocessing techniques including data cleaning, integration, transformation, and reduction, Exploratory data analysis (EDA) methods to understand the structure and characteristics of datasets.

Module 2: Classification and Prediction Set

Classification methods encompassing decision trees, Bayesian classifiers, k-nearest neighbors, and support vector machines. Assessment measures for evaluating classification model efficacy, including accuracy, precision, recall, and F1-score. Forecasting approaches such as regression analysis, time series prediction, and ensemble learning methodologies.

Module 3: Clustering and Association Analysis

Clustering algorithms including k-means, hierarchical clustering, and density-based clustering. Association rule mining techniques such as Apriori algorithm for discovering frequent item sets and association rules. Evaluation measures for clustering quality and association rule interestingness.

Module 4: Anomaly Detection and Outlier Analysis

Anomaly detection methods for identifying rare events, outliers, and anomalies in data. Techniques including statistical approaches, clustering-based methods, and machine learning algorithms for anomaly detection. Applications of anomaly detection in various domains such as fraud detection, network security, and healthcare.

Module 5: Advance Data mining

Advanced data mining techniques such as text mining, web mining, and social network analysis. Feature selection and dimensionality reduction methods to improve model performance and interpretability. Ethical and privacy considerations in data mining, including data anonymization and fairness-aware mining.

Text/Reference Books:	
	1. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei 2. "Introduction to Data Mining" by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar 3. "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy 4. "Principles of Data Mining" by David Hand, Heikki Mannila, and Padhraic Smyth 5. "Pattern Recognition and Machine Learning" by Christopher M. Bishop
List of Experiments	
	1. Data preprocessing: Cleaning noisy data, handling missing values, and transforming categorical variables. 2. Exploratory data analysis: Visualizing data distributions, correlations, and outliers using plots and summary statistics. 3. Classification: Implementing decision trees, naive Bayes, and k-nearest neighbors algorithms for classifying datasets. 4. Model evaluation: Assessing classification model performance using metrics such as accuracy, precision, recall, and F1-score. 5. Clustering: Applying k-means and hierarchical clustering algorithms to group similar data points together. 6. Association rule mining: Using the Apriori algorithm to discover frequent itemsets and association rules from transaction data. 7. Anomaly detection: Detecting outliers and anomalies in data using statistical methods and machine learning algorithms. 8. Text mining: Extracting features from textual data, performing sentiment analysis, and building text classification models. 9. Web mining: Crawling web data, extracting information from web pages, and analyzing web usage patterns. 10. Feature selection: Identifying relevant features and reducing dimensionality to improve model performance and interpretability.

Course title:	**Dissertation Phase-I**	**Sub code:**	**EEC701**			
		Structure:	**L**	**T**	**P**	**C**
			0	**0**	**20**	**10**
Course Objective:	Final Year Projects represent the culmination of study towards the Bachelor of Engineering degree. Projects offer the opportunity to apply and extend material learned throughout the program. Assessment is by means of a seminarpresentation, submission of a thesis, and a public demonstration of work undertaken. The projects undertaken span a diverse range of topics, including theoretical, simulation and experimental studies. The emphasis is necessarily on facilitating student learning in technical, project management and presentation spheres.					
Course outcome:	• Apply knowledge and skills acquired throughout the program to solve real-world engineering problems. • Gain hands-on experience in designing, implementing, and testing innovative engineering solutions. • Develop project management skills including planning, scheduling, and resource allocation. • Enhance communication and presentation skills through the documentation and presentation of project outcomes. • Prepare for transition into the workforce or further academic pursuits by demonstrating competency in a specialized area of electronics engineering.					
Content						
Module 1: Term Work						

Dissertation Phase-II, is in continuation of Project Part-I undertaken by the candidates in first term. The term workshall consist of a typed report of about 70 pages or more, on the work carried out by the batch of students inrespect of the project assigned, during first term and second term. It should be in the proper format.
Module 2: Practical Examination:
It shall consist of demonstration of designed, fabricated project and oral based on it. The said examination will be conducted by a panel of two examiners; one of them will be a guide and another will be an externalexaminer. The external examiner will be either from the allied industry or a senior faculty member from another institute.

Course title:	**Dissertation Phase-II**	**Sub code:**	**EEC801**			
		Structure:	**L**	**T**	**P**	**C**
			0	**0**	**40**	**20**

Course Objective:	Final Year Projects represent the culmination of study towards the Bachelor of Engineering degree. Projects offer the opportunity to apply and extend material learned throughout the program. Assessment is by means of a seminarpresentation, submission of a thesis, and a public demonstration of work undertaken. The projects undertaken span a diverse range of topics, including theoretical, simulation and experimental studies. The emphasis is necessarily on facilitating student learning in technical, project management and presentation spheres.
Course outcome:	• Apply knowledge and skills acquired throughout the program to solve real-world engineering problems. • Gain hands-on experience in designing, implementing, and testing innovative engineering solutions. • Develop project management skills including planning, scheduling, and resource allocation. • Enhance communication and presentation skills through the documentation and presentation of project outcomes. • Prepare for transition into the workforce or further academic pursuits by demonstrating competency in a specialized area of electronics engineering.

Content
Module 1: Term Work
Dissertation Phase-II, is in continuation of Project Part-I undertaken by the candidates in first term. The term workshall consist of a typed report of about 70 pages or more, on the work carried out by the batch of students inrespect of the project assigned, during first term and second term. It should be in the proper format.
Module 2: Practical Examination:
It shall consist of demonstration of designed, fabricated project and oral based on it. The said examination will be conducted by a panel of two examiners; one of them will be a guide and another will be an externalexaminer. The external examiner will be either from the allied industry or a senior faculty member from another institute.

Annexure-I
List of Professional Electives

Sl No.	Elective	Course Name	Credit
1	Professional Elective I (PE201A/PE201B)	1. Energy and Environment Engineering 2. Smart Grids and Energy Management	3
2	Professional Elective II (PE401A/PE401B)	1. Artificial Intelligence 2. Natural Language Processing	3
3	Professional Elective III (PE402A/PE402B)	1. Management Principles and Practices 2. Employability skill	3
4	Professional Elective IV (PE601A/PE601B)	1. Theory of App Development 2. Theory of Web Development	4
5	Professional Elective V (PE602A/PE602B)	1. Cloud & Edge Computing 2. Soft Computing	3

Course title:	**ENERGY AND ENVIRONMENTAL ENGINEERING**	**Sub code**	PE201A

	Structure	L	T	P	C
		2	1	0	3

Course Objective:	• To provide students with a fundamental understanding of energy systems and their environmental impact, fostering awareness of sustainable practices and solutions. • To analyze environmental challenges arising from energy utilization and explore engineering solutions to mitigate environmental impacts, promoting a holistic approach to energy and environmental management.
Course Outcome:	• Principal renewable energy systems • Explore the environmental impact of various energy sources and also theeffects of different types of pollutants. • An understanding the problems of energy distribution, design, plan and execute • To make a thought in terms of scientific and technological advancement in the spirit of a sustainable energy greenhouse • Understand the relationships between natural resources, consumption, population, economics of consumerism, etc in an environmental context

Content

Module 1: Introduction:

Present Energy resources in India and its sustainability - Different type of conventional Power Plant-
-Energy Demand Scenario in India-Advantage and Disadvantage of conventional Power Plants –Conventional Vs Non-conventional power generation.

Module 2: Basics of Solar Energy:

Basics of Solar Energy- Solar Thermal Energy- Solar Photovoltaic- Advantages and Disadvantages-Environmental impacts and safety.

Module 3: Wind, Biomass, Geothermal conversions and resources:

Power and energy from wind turbines- India's wind energy potential- Types of wind turbines- Offshore Wind energy- Environmental benefits and impacts.
Biomass resources-Biomass conversion Technologies- Feedstock pre-processing and treatmentmethods- Bioenergy program in India-Environmental benefits and impacts.
Geothermal Energy resources –Ocean Thermal Energy Conversion – Tidal.

Module 4: Air Pollution and Greenhouse gases :

Air pollution- Sources, effects, control, air quality standards, air pollution act, air pollution measurement. Water pollution-Sources and impacts, Soil Pollution-Sources and impacts, disposal of solid waste.
Greenhouse gas effect, acid rain. Noise pollution. Pollution aspects of various power plants. Fossilfuels and impacts, Industrial and transport emissions- impacts.

Module 5: Social Issues related to Environment:

From Unsustainable to Sustainable development, Urban problems related to energy, Water conservation, rain water harvesting, watershed management, Resettlement and rehabilitation of people; its problems and concerns. Climate change, global warming, acid rain, ozone layer depletion and Eutrophication, Wasteland reclamation. Consumerism and waste products. Environment Protection Act. Air (Prevention and Control of Pollution) Act. Water (Prevention and control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act.

Reference Books:

1) Boyle, G. 2004. Renewable energy: Power for a sustainable future, Oxford University press.
2) B H Khan, Non-Conventional Energy Resources-The McGraw –Hill Second edition.
3) G. D. Rai, Non-conventional energy sources, Khanna Publishers, New Delhi, 2006.
4) Gilbert M. Masters, Introduction to Environmental Engineering and Science, 2nd Edition, Prentice Hall, 2003.

Course title:	Smart Grids and Energy Management	Sub code:	PE201B			
		Structure	L	T	P	C

		2	**1**	**0**	**3**
Course Objective:	• To introduce the fundamental concepts of smart grids and the technologies enabling modern energy management systems. • To develop an understanding of demand-side management and its role in energy conservation and load balancing. • To explore the integration of Internet of Things (IoT) in energy management for monitoring, control, and automation. • To provide students with knowledge of energy auditing practices and standards in assessing and optimizing energy use. • To analyze data analytics and machine learning techniques in optimizing energy management and predictive maintenance within smart grids.				
Course Outcome:	• Explain the architecture, components, and functioning of smart grids and their significance in modern power systems. • Apply demand-side management strategies to optimize energy consumption and improve grid reliability. • Implement IoT-based solutions for energy monitoring, control, and automation in smart grid applications. • Perform energy audits and assessments, understanding various energy performance metrics and their relevance. • Utilize data analytics and machine learning tools to enhance energy efficiency, predict failures, and support decision-making in energy management.				

Content

Module 1: Introduction to Smart Grids and Energy Management

- Evolution of traditional grids to smart grids, features, benefits, and architecture of smart grids.
- Key components of smart grids: Advanced Metering Infrastructure (AMI), Phasor Measurement Units (PMUs), and communication networks.
- Role of smart grids in integrating renewable energy sources and managing distributed generation.
- Cybersecurity challenges in smart grids and potential solutions.

Module 2: Demand-Side Management and Energy Efficiency

- Concept and importance of demand-side management (DSM), demand response (DR), and load forecasting.
- Energy conservation techniques and energy efficiency measures.
- Techniques for load shifting, peak shaving, and balancing energy supply and demand.
- Economic incentives for DSM and DR programs, and their impact on consumers and utilities.

Module 3: IoT in Smart Grids and Energy Management

- Overview of IoT architecture and protocols for smart grids.
- IoT-enabled devices and sensors for real-time monitoring, data collection, and remote control of energy systems.
- Smart meters, smart appliances, and home energy management systems (HEMS).
- Communication technologies in IoT-enabled smart grids: Zigbee, LoRa, NB-IoT, and 5G.

Module 4: Energy Auditing and Standards

- Basics of energy auditing: types of audits, objectives, and methodologies.
- Key energy performance indicators (KPIs), energy consumption analysis, and benchmarking.
- Standards and protocols in energy auditing (ISO 50001, ASHRAE standards).
- Role of energy audits in industry, buildings, and residential applications, case studies of successful energy audits.

Module 5: Data Analytics and Machine Learning in Energy Management

- Introduction to data analytics and machine learning techniques in energy management.
- Applications of machine learning for load forecasting, anomaly detection, and predictive maintenance.
- Case studies on using analytics for energy optimization and fault detection in smart grids.
- Overview of tools and platforms (MATLAB, Python, R) for energy data analysis and visualization.

Reference Books:

1. **James Momoh**, *Smart Grid: Fundamentals of Design and Analysis*, IEEE Press, Wiley.
2. **Gil Masters**, *Renewable and Efficient Electric Power Systems*, 2nd Edition, Wiley.
3. **Janaka Ekanayake, Nick Jenkins, Kithsiri Liyanage, Jianzhong Wu, Akihiko Yokoyama**, *Smart Grid: Technology and Applications*, Wiley.
4. **Fereidoon P. Sioshansi**, *Smart Grid: Integrating Renewable, Distributed & Efficient Energy*, Academic Press.
5. **Ali Keyhani**, *Design of Smart Power Grid Renewable Energy Systems*, Wiley-IEEE Press.

Subject Code	Course Name	L	T	P	Total Credits
PE401A	**Artificial Intelligence**	2	0	2	3

COURSE OUTCOME:

1. Understand the fundamentals of supervised, unsupervised, and reinforcement learning techniques and apply suitable algorithms to solve classification and regression problems.
2. Analyze and evaluate machine learning models using appropriate performance metrics and address challenges like overfitting, underfitting, and bias-variance tradeoffs.
3. Design and implement decision trees, ensemble methods (bagging, boosting), and feature engineering strategies to improve model accuracy and generalizability.
4. Develop and train neural networks using backpropagation, activation functions, and gradient-based optimization techniques; apply regularization and optimization to enhance learning.
5. Apply clustering algorithms, association rules, and dimensionality reduction techniques for unsupervised learning and exploratory data analysis.
6. Explore and implement advanced AI/ML techniques including population-based algorithms, reinforcement learning, deep learning for sequential data, NLP using transformers, and understand MLOps practices for model deployment.

COURSE CONTENT:

Module 1: Foundations of Machine Learning

Designing a learning system, Types of machine learning: Problem based learning, Supervised learning, Unsupervised learning, Reinforcement learning, Linear Regression: Weights and Features, Applications, Cost Functions, Finding best fit line, Gradient Descent Algorithm: Learning Algorithm, First order derivatives, Linear regression using gradient descent, Learning rate, Logistic Regression, Sigmoid Function, Cost Function for Logistic Regression, Multi-class classification, Probability Distribution, SoftMax Function, Polynomial Regression, Performance Metrics: Classification (Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC-AUC), Regression (MSE, MAE, RMSE, R2 Score).

Module 2: Decision Trees, Ensembles, and Feature Engineering

Decision Tree, Selecting Best Splitting Attribute, CART (Gini Index). ID3 (Entropy, Information Gain), Hyperparameters in Decision tree, Issues in Decision tree learning. Overfitting and Underfitting, Bias and Variance, Cross Validation. Ensemble Learning (Concord's Theorem), Bagging, Bootstrap and Aggregation, Random Forest. Boosting, AdaBoost, Gradient Boost. Feature Engineering, Feature Selection, Feature Extraction.

Module 3: Neural Networks and Bayesian Learning

Artificial Neural Network, Neural network representation, Perceptron model, Stepwise v/s Sigmoid function, Multilayer perceptron model, Matrix Calculus (Jacobian, Hessian Matrix), Computation Graph, Backpropagation Algorithm, Activation Functions, Stochastic Gradient Descent, Batch Gradient Descent, Mini-Batch Gradient Descent, Vanishing and Exploding Gradients, Overfitting Problem, Regularization (Ridge, Lasso, Elastic), Dropout and Early Stopping, Bayesian Learning: Bayes theorem and concept learning, Naïve Bayes classifier, Gibbs Algorithm, Bayesian belief networks, The EM algorithm, Support Vector Machines, Hyperplane, Support Vectors, Kernels, Non-Parametric Regression, Locally weighted regression, K-nearest neighbour.

Module 4: Unsupervised Learning and Dimensionality Reduction

Unsupervised learning (clustering, Association rule learning, Dimensionality reduction), Common distance

Measures, k-means clustering, Elbow method, Hierarchical Clustering – agglomerative and divisive, Dendogram, Similarity measures for hierarchical clustering, DBSCAN, Cluster Quality (R index, Silhouette Coefficient), Dimensionality Reduction, Principal Component Analysis, Singular Vector Decomposition, T-distributed Stochastic Neighbour Embedding

Module 5: Advanced AI Techniques and MLOps

Population Based Algorithms: Genetic Algorithm, Fitness Function, Selection, Crossover, Mutation, Swarm Optimization, Particle Swarm Optimization, Ant-Colony Optimization, Reinforcement Learning, Actors, State, Reward Policy, Actions, Computer Vision, Convolutional Neural Networks, Deep Learning for Sequential Data, Recurrent Neural Network, LSTM, GRU, Natural Language Processing, Word Embeddings, Transformers (BERT and GPT), Building and Deploying ML models (MLFlow), MLOps, Need of MLOps, ML Production Infrastructure.

LABORATORY/PRACTICALS:

1. Implement linear and logistic regression models from scratch and evaluate using performance metrics (MSE, MAE, Accuracy, F1-score, ROC-AUC).
2. Build a decision tree classifier using Gini Index and Entropy and visualize the tree structure.
3. Demonstrate overfitting and underfitting using polynomial regression; apply regularization methods (Ridge, Lasso, Elastic Net).
4. Perform ensemble learning using Random Forest, Bagging, and Boosting (AdaBoost, Gradient Boost).
5. Train and evaluate a multi-layer perceptron neural network with different activation functions and optimizers (SGD, Mini-batch GD).
6. Apply PCA and t-SNE for dimensionality reduction and visualize high-dimensional data.
7. Implement k-means clustering and DBSCAN; evaluate cluster quality using silhouette score and R-index.
8. Develop and evaluate a Naïve Bayes classifier and compare with k-Nearest Neighbors.
9. Implement Genetic Algorithm and Particle Swarm Optimization for function optimization problems.
10. Build and train a deep learning model for sequential data using RNN/LSTM/GRU for time-series or NLP tasks.
11. Develop a text classification or sentiment analysis system using BERT or GPT embeddings.
12. Deploy a trained machine learning model using MLFlow and explore MLOps workflows.

RESOURCES/BOOKS FOR REFERENCES:

1. Mitchell, Tom M. Machine Learning. 1st ed. McGraw Hill, 2017. ISBN 978-1259096952.
2. Alpaydin, Ethem. Introduction to machine learning. 4th ed. Phi, 2020. ISBN 978-8120350786.
3. Campesato, Oswald. Artificial intelligence, machine learning, and deep learning. 1st ed. Mercury Learning and Information, 2020. ISBN 9781683924665.

Subject Code	Course Name	L	T	P	Total Credits
PE401B	**Natural Language Processing**	2	0	2	3

COURSE OUTCOME:

1. Understand fundamental concepts and challenges of Natural Language Processing, including text preprocessing, linguistic ambiguity, and language-specific issues.
2. Apply standard text normalization techniques and represent textual data using models such as TF-IDF, Word2Vec, and GloVe for effective information retrieval and semantic analysis.
3. Implement neural network-based approaches for sequence modeling using RNNs, LSTMs, and GRUs, and evaluate their performance in real-world NLP tasks.
4. Utilize advanced NLP models and transformer architectures such as BERT, GPT, and RoBERTa for tasks like classification, generation, and question answering.

5. Analyze and apply methods for machine translation and text summarization using statistical, neural, and transformer-based approaches.
6. Design and develop end-to-end NLP applications using pre-trained models and embeddings, integrating evaluation methods for model accuracy and relevance.

COURSE CONTENT:

Module 1: Fundamentals of Natural Language Processing

Introduction to NLP and its real-world applications; industry demand and relevance. Challenges in NLP: ambiguity, coreference, context-dependent language, domain-specific language, and low-resource languages. Preprocessing techniques: segmentation, stemming, lemmatization, spelling correction, tokenization, stop words. Introduction to linguistic resources: synsets, hypernyms, WordNet, and WordNet similarity. Introduction to language corpora, n-gram models, and Hidden Markov Models (HMM).

Module 2: Text Processing and Representation Techniques

Text normalization techniques including script normalization and case folding. Handling multilingual and code-mixed text. Parallel corpus and cross-lingual information retrieval. Text representation techniques: vector space model, term frequency (TF), TF-IDF, distributional semantics. Word embeddings: Word2Vec (CBOW and Skip-gram), GloVe, document-to-vector representations, embeddings for regional languages.

Module 3: Deep Learning for NLP

Neural networks for text processing. Sequence modeling using Recurrent Neural Networks (RNNs), vanishing and exploding gradient problems. Advanced architectures: LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), bidirectional models, and sequence-to-sequence modeling. Contextual representation models: transfer learning in NLP, multilingual embeddings, and MUSE. Applications including POS tagging, Named Entity Recognition (NER), sentiment analysis, and text clustering

Module 4: Advanced NLP Models and Architectures

Transformer architecture and attention mechanisms in NLP. Deep contextual models such as BERT, ALBERT, RoBERTa, GPT, and ULMFiT. Sub-word modeling and byte-pair encoding. Applications in natural language generation, self-attention in generative tasks, question answering systems, and 1D-CNN architectures for text classification.

Module 5: Applications in Text Summarization and Translation

Topic modeling and latent semantic analysis (LSA). Machine translation: statistical and neural approaches. Use of transformers in machine translation and summarization tasks. Techniques in text summarization: extractive vs. abstractive methods. Evaluation of summarization models and their deployment in real-world applications.

LABORATORY/PRACTICALS:

1. Implementation of text document reading, parsing, and applying text pre-processing techniques such as tokenization, stemming, and lemmatization.
2. Implementation of WordNet-based document representation and word search functionality.
3. Development of word-level and sentence-level text representation models for document retrieval and text searching.
4. Construction of one-hot vector representations for a given dataset in an Indian language.
5. Design and training of a basic neural network for simple text classification tasks.
6. Implementation of topic modeling using clustering techniques and Latent Semantic Analysis (LSA).
7. Implementation of a text classification model for sentiment analysis using classical or deep learning approaches.
8. Implementation of Part of Speech (POS) tagging and Named Entity Recognition (NER) using NLP libraries or custom models.
9. Implementation of a machine translation model using statistical or neural machine translation techniques.
10. Design and implementation of a chatbot using sequence-to-sequence (Seq2Seq) modeling architecture.
11. Development of text generation models for domain-specific content such as social media,

news, or customer service applications.

RESOURCES/BOOKS FOR REFERENCES:

1. Rao Delip and Brian McMahan, natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning (1 ed.), O'Reilly Media, 2019. ISBN 9781491978238.
2. Eisenstein Jacob, Introduction to natural Language Processing (1 ed.), The MIT Press; Illustrated edition, 2019. ISBN 9780262042843

Course title:	**Management Principles and Practices**	**Sub code:**	**PE402A**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**0**	**3**

Course Objective:	• Understand the fundamental principles and functions of management in the context of technology-driven organizations. • Develop skills in planning, decision-making, organizing, leading, controlling, and monitoring within the tech industry. • Analyze and apply management theories and practices to address challenges specific to the field of computer science engineering. • Explore the role of human resource management in nurturing talent and fostering innovation in tech teams.
Course Outcome:	• Demonstrate proficiency in applying management principles to solve real-world problems in technology-driven environments. • Develop effective planning, decision-making, and organizational skills relevant to computer science engineering projects. • Exhibit leadership qualities and the ability to motivate and engage tech teams towards achieving organizational goals. • Implement quality management and project management methodologies to ensure the success of tech projects. • Apply human resource management strategies to recruit, train, and retain talent, fostering a productive and collaborative work culture in technology organizations.

Content

Module 1: Introduction to Management

Overview of management principles and functions. Evolution of management theories. Managerial roles and responsibilities. Importance of management in organizations. Case studies on successful management practices

Module 2: Planning and Decision Making

Understanding the planning process: setting goals, objectives, and strategies. Types of plans: strategic, tactical, operational. Decision-making models and techniques. Risk management and contingency planning. Case studies on effective planning and decision-making in tech companies

Module 3: Organizing and Leading

Principles of organizational structure and design. Delegation of authority and responsibility. Leadership styles and theories. Motivation and employee engagement strategies. Case studies on leadership and organizational effectiveness in tech teams.

Module 4: Controlling and Monitoring

Techniques for performance measurement and control. Quality management principles and practices. Project management methodologies (e.g., Agile, Waterfall). Monitoring and evaluating project progress. Case studies on successful project management and quality control in tech projects

Module 5: Human Resource Management

Recruitment and selection processes for tech roles. Training and development programs for tech professionals. Performance appraisal systems and feedback mechanisms. Employee relations and conflict resolution strategies. Case studies on HRM practices in tech companies.

Reference Books	
	1. "Management: Principles and Practices for Engineering and Construction" by Alan C. Twort and J. Gordon Rees 2. "Introduction to Management Science" by Bernard W. Taylor III 3. "Management Information Systems: Managing the Digital Firm" by Kenneth C. Laudon and Jane P. Laudon 4. "Human Resource Management" by Gary Dessler 5. "Project Management for Engineering, Business, and Technology" by John M. Nicholas and Herman Steyn

Course title:	**Employability skill**	**Sub code:**	**PE402B**			
		Structure:	**L**	**T**	**P**	**C**
			2	**1**	**0**	**3**

Course Objective:	• To enhance students' soft skills, communication, and interpersonal skills, which are essential for career development. • To develop critical thinking, problem-solving, and decision-making skills required in the workplace. • To improve students' writing skills, including technical reports, resumes, and business communication. • To build teamwork, leadership, and time-management skills for effective collaboration in a professional setting. • To prepare students for interviews, presentations, and group discussions, helping them become industry-ready professionals.
Course Outcome:	• Demonstrate effective verbal and written communication skills for professional contexts. • Exhibit enhanced problem-solving abilities and the capability to think critically under different scenarios. • Write clear and concise technical reports, resumes, and professional emails, adhering to standard formats. • Collaborate effectively in teams, demonstrating leadership and managing time efficiently in project work. • Show readiness for job interviews and perform well in group discussions, technical rounds, and HR interviews.

Content

Module 1: Communication Skills and Professional Writing

- **Verbal Communication**: Importance of clear and concise communication, effective listening, and presentation skills.
- **Written Communication**: Email etiquette, technical report writing, formal letters, and memos.
- **Resume Building**: Structure of a professional resume, creating an impactful resume that highlights skills and achievements.
- **Cover Letters**: Writing personalized cover letters to accompany job applications.
- **Business Communication**: Effective use of communication in meetings, conferences, and business correspondence.

Module 2: Critical Thinking, Problem-Solving, and Decision Making

- **Critical Thinking**: Analyzing information objectively, logical reasoning, and evaluating arguments.
- **Problem-Solving**: Techniques for solving complex problems, identifying root causes, and finding solutions.
- **Decision-Making Models**: Understanding various decision-making models like SWOT analysis and cost-benefit analysis.
- **Creativity in Problem Solving**: Techniques for fostering creativity and innovation in finding solutions.
- **Case Studies**: Analyzing real-life industry problems and providing solutions using critical thinking.

Module 3: Teamwork, Leadership, and Collaboration

- **Teamwork Skills**: Importance of teamwork, roles within a team, and conflict resolution techniques.

- **Leadership Styles**: Different leadership styles (e.g., transformational, transactional, and situational leadership) and their applications.
- **Effective Collaboration**: Techniques for improving collaboration in a diverse work environment.
- **Project Management**: Basics of project management, resource allocation, scheduling, and tracking progress.
- **Group Dynamics**: Understanding group roles, managing group conflict, and enhancing group performance.

Module 4: Time Management and Organizational Skills

- **Time Management**: Techniques like the Pomodoro technique, Eisenhower Matrix, and setting SMART goals.
- **Prioritization**: How to prioritize tasks based on urgency and importance.
- **Stress Management**: Techniques for managing stress and maintaining productivity during peak workloads.
- **Work-Life Balance**: Strategies for achieving a balance between professional responsibilities and personal life.
- **Organizational Skills**: Managing resources, organizing workspaces, and using tools for scheduling and tracking progress.

Module 5: Interview Preparation, Group Discussions, and Public Speaking

- **Interview Techniques**: Preparation for technical, HR, and behavioral interviews. Mock interviews and interview etiquette.
- **Group Discussions**: Approaches to participating effectively in group discussions, including listening skills, argument presentation, and summarizing.
- **Public Speaking**: Overcoming stage fear, techniques for effective public speaking, and making impactful presentations.
- **Communication in Interviews**: Handling difficult questions, body language, and personal branding.
- **Personality Development**: Confidence-building, self-awareness, and maintaining a positive attitude in professional settings.

Reference Books

1. **Meenakshi Raman and Sangeeta Sharma**, *Technical Communication: Principles and Practice*, 3rd Edition, Oxford University Press.
2. **P. K. Sinha**, *Employability Skills for Engineers*, Cengage Learning.
3. **Rajendra Pal and J.S. Korlahalli**, *Essentials of Business Communication*, 8th Edition, Sultan Chand & Sons.
4. **K. K. Sinha**, *Soft Skills: Know Yourself and Know the World*, 2nd Edition, Cengage Learning.
5. **Stephen P. Robbins**, *Organizational Behavior*, 12th Edition, Pearson Education.

Course title:	**Theory of App Development**	**Sub code:**	**PE601A**			
		Structure:	**L**	**T**	**P**	**C**
			2	**0**	**4**	**4**

Course Objectives	1. Gain foundational knowledge of mobile app development platforms, frameworks, and lifecycle. 2. Develop familiarity with prevalent programming languages and UI/UX design principles for app development. 3. Understand app architecture, design patterns, and best practices for creating scalable and maintainable apps. 4. Learn data management techniques, including storage options, synchronization, CRUD operations, and security measures.
Course Outcome	1. Demonstrate proficiency in building modular, reusable, and testable app components following industry-standard practices. 2. Analyze and implement various architectural patterns to design scalable and maintainable mobile applications. 3. Apply data management and persistence strategies to enhance app efficiency and security. 4. Develop networking skills for integrating web services, handling asynchronous requests, and ensuring secure API access.

	5. Implement effective testing, deployment, and maintenance strategies to ensure app quality, reliability, and performance.

Content

Module 1: Introduction to App Development

Introduction to mobile app development platforms and frameworks, comprehension of the app development lifecycle, familiarity with programming languages prevalent in app development such as Java, Swift, and Kotlin, understanding of user interface (UI) and user experience (UX) design principles for apps, and initiation to version control systems like Git for facilitating collaborative app development.

Module 2: App Architecture and Design Pattern

Grasping the structure of mobile applications, including the Model-View-Controller architecture, delving into prevalent design patterns like Singleton, Observer, and Factory, initiating architectural patterns for creating scalable and maintainable apps such as MVP and MVVM, analyzing case studies and real-world examples of app architectures, and adhering to best practices in designing modular, reusable, and testable app components.

Module 3: Data Management and Persistence

Summary of data storage choices for mobile apps, including SQLite, Core Data, and Realm, comprehension of data synchronization and offline functionalities in mobile apps, initiation to CRUD operations (Create, Read, Update, Delete) in mobile app development, examination of data caching techniques for enhancing app efficiency, integration of data encryption and security protocols in mobile apps.

Module 4: Networking and Web Services

Initiation to networking principles essential for app development, such as HTTP and RESTful APIs, insight into asynchronous programming methods for managing network requests, execution of network request and response management utilizing libraries like Retrofit and Alamofire, overview of authentication and authorization mechanisms ensuring secure API access, incorporation of best practices for error handling and retry tactics in network communication.

Module 5: Testing, Deployment, and Maintenance

Summary of various testing methodologies in app development, including unit testing, integration testing, and UI testing, initiation to test-driven development (TDD) and behavior-driven development (BDD) approaches, comprehension of app deployment procedures across platforms such as App Store and Google Play, formulation of strategies for app versioning, updating, and maintenance, exploration of app analytics and monitoring utilities to optimize performance and track bugs.

Reference Books:

1. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart
2. "iOS Programming: The Big Nerd Ranch Guide" by Christian Keur and Aaron Hillegass
3. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
4. "Effective Java" by Joshua Bloch
5. "iOS Programming: Pushing the Limits" by Rob Napier and Mugunth Kumar

List of Experiments

1. Create a simple Android or iOS app to understand the app development lifecycle and UI/UX design principles.
2. Implement different architectural patterns (e.g., MVC, MVP, MVVM) in separate projects to comprehend their usage and benefits.
3. Develop a mobile app that utilizes SQLite or Core Data for data storage and synchronization to practice data management techniques.
4. Integrate RESTful APIs into a mobile app to handle network requests asynchronously and understand networking principles.
5. Build a test suite for a mobile app, including unit tests, integration tests, and UI tests, using appropriate testing frameworks.
6. Explore version control systems like Git by collaborating on a mobile app project with peers, managing branches, and merging changes.
7. Deploy a mobile app to both the App Store and Google Play Store, understanding the app deployment process for different platforms.
8. Implement app versioning and update strategies to maintain and release new versions of a mobile app effectively.
9. Utilize analytics and monitoring tools to track user engagement, performance metrics, and identify potential issues in a deployed app.
10. Work on a real-world app project, integrating various concepts learned throughout the course, from architecture design to deployment and maintenance.

Course title:	Theory of Web Development	Sub code:	PE601B			
		Structure:	L	T	P	C
			2	0	4	4

Course Objectives	1. To introduce the fundamental concepts of web development and its architecture, including client-server interaction. 2. To provide a detailed understanding of web technologies, including HTML, CSS, JavaScript, and web frameworks. 3. To enable students to design, develop, and deploy static and dynamic web pages. 4. To teach the principles of web accessibility, responsiveness, and usability to create user-friendly web applications. To familiarize students with web development tools, best practices, and performance optimization techniques.
Course Outcome	1. Demonstrate an understanding of the basic structure and design of web pages using HTML, CSS, and JavaScript. 2. Develop static and dynamic web pages, integrating multimedia elements and creating interactive web applications. 3. Apply responsive design principles to create websites that work across multiple devices and screen sizes. 4. Implement server-side programming concepts to build dynamic, data-driven websites. 5. Analyze and apply web development tools for performance optimization, debugging, and version control.

Content

Module 1: Introduction to Web Technologies

- Overview of the Web: Client-Server Architecture, HTTP/HTTPS protocols.
- Understanding URLs, DNS, and Web Servers.
- Web Development Stacks: LAMP, MERN, MEAN.
- Introduction to Front-end vs Back-end development.
- Role of web browsers and rendering engines.

Module 2: Web Design and Front-End Technologies

- **HTML5**: Structure of a web page, semantic HTML, forms, tables, links, and multimedia elements (audio, video, images).
- **CSS3**: Styling web pages, box model, layout techniques (flexbox, grid system), styling for text, fonts, and colors.
- **Responsive Web Design**: Media queries, fluid grids, and designing for mobile-first.
- **JavaScript Basics**: Variables, data types, operators, control structures, functions, arrays, and objects.
- DOM manipulation and event handling.

Module 3: Advanced Front-End Development and Frameworks

- **JavaScript Frameworks**: Introduction to React.js, Angular, or Vue.js.
- **AJAX and Asynchronous Programming**: Fetch API, promises, and callbacks for dynamic content loading.
- **Version Control with Git**: Git basics, repositories, branching, and version control workflows (GitHub/GitLab).
- **Web Accessibility (WCAG)**: Accessibility principles, ARIA roles, creating accessible forms and navigation.

Module 4: Back-End Development and Web Databases

- **Server-side Scripting**: Introduction to server-side languages (Node.js, PHP, Python, etc.).
- **Web Frameworks**: Express.js (for Node.js), Flask/Django (for Python), and Laravel (for PHP).
- **Databases**: Introduction to SQL and NoSQL databases (MySQL, MongoDB).
- **CRUD Operations**: Create, Read, Update, Delete functionality with database integration.
- **Authentication and Authorization**: Implementing user login systems using sessions, cookies, and JWT (JSON Web Tokens).

Module 5: Testing, Deployment, and Maintenance

- **Web Hosting and Deployment**: Introduction to web hosting services, domain registration, and deploying web applications (e.g., using AWS, Heroku, or Firebase).
- **Web Application Security**: Understanding common web vulnerabilities (XSS, CSRF, SQL Injection) and their prevention.
- **Performance Optimization**: Caching, minification, image optimization, lazy loading, and web performance best practices (Google Lighthouse).
- **Web Standards and Best Practices**: SEO optimization, cross-browser compatibility, and progressive web apps (PWA).

Reference Books:

1. **Jon Duckett**, *HTML and CSS: Design and Build Websites*, Wiley.
2. **Ethan Marcotte**, *Responsive Web Design*, A Book Apart.
3. **David Flanagan**, *JavaScript: The Definitive Guide*, 7th Edition, O'Reilly Media.
4. **Luke Wroblewski**, *Mobile First*, A Book Apart.
5. **Ben Frain**, *Responsive Web Design with HTML5 and CSS3*, 2nd Edition, Packt Publishing.
6. **Brad Dayley**, *Node.js, MongoDB, and AngularJS Web Development*, 3rd Edition, Addison-Wesley.
7. **Robert W. Sebesta**, *Programming the World Wide Web*, 7th Edition, Pearson.

List of Experiments

1. **Basic HTML Page Creation**: Develop a simple static webpage using HTML.
2. **CSS Styling**: Apply different CSS properties for text formatting, layout, and design.
3. **Responsive Web Design**: Create a mobile-first website using media queries and responsive grid layouts.
4. **JavaScript Functions and DOM Manipulation**: Implement a basic JavaScript-based calculator or to-do list application with DOM manipulation.
5. **AJAX in Web Development**: Develop an interactive webpage that fetches and displays data asynchronously using AJAX.
6. **Form Validation**: Implement client-side form validation with JavaScript for user input.
7. **Version Control with Git**: Create a Git repository, commit changes, and practice using branches and pull requests.
8. **Node.js Server Development**: Build a basic Node.js server and serve static files using Express.js.
9. **Database Integration**: Implement a simple CRUD application with a MySQL or MongoDB database.
10. **Web Application Deployment**: Host a web application on a cloud service like AWS, Heroku, or Firebase and deploy a live website.

Subject Code	Course Name	L	T	P	Total Credits
PE602A	Cloud & Edge Computing	2	1	0	3

COURSE OUTCOME:

1. Demonstrate the cloud computing models according to computing requirements
2. Illustrate Cloud Computing services for engineering practice and responsibilities
3. Select the most appropriate cloud architecture
4. Identify the security aspects in cloud computing
5. Describe edge computing concepts and technologies, including infrastructure components, models, services, and their applications.

COURSE CONTENT:

Module 1: Evolution of Model Computing

Evolution of sharing on the Internet, Cluster Computing, Grid Computing, Parallel Computing and Distributed Computing, Introduction of Cloud Computing: Definition of cloud, Cloud Deployment Models, Cloud Service Models, Key Characteristics, Benefits and Risks in Cloud Computing, Service oriented architecture (SOA) and Cloud Computing Reference Architecture by IBM

Module 2: Services Delivered from the Cloud

Model architecture, Benefits and Drawbacks: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Business-Process-as-aservice (BPaaS), Identity-as-aservice (IDaaS), Communication-as-a-service (CaaS), Monitoring-as-a-service (MaaS), Storage as a service: Traditional storage versus storage cloud, Cloud Service providers: Infrastructure as service: Amazon EC2, Platform as Service: Google App Engine, Force.com.

Module 3: Cloud Technologies

Web services: SOAP and REST, SOAP VS REST, Virtualization, Desktop and Application Virtualization, Storage and Network Virtualization., QoS [Quality of Service] of Cloud

Module 4: NoSQL data stores

Table-based: Google BigTable, Amazon Dynamo, Cassandra, Document-based, sCouchDB, MongoDB Batch cloud computing: MapReduce and Hadoop. Applications in NoSQL data stores.

Module 5: Edge Computing

Fundamentals of Edge computing**:** Distributed systems and edge computing core concepts , Edge computing architectures , IoT devices and edge computing in IoT gateways , Edge computing interfaces , Edge computing vs cloud computing vs fog computing , Edge Computing Networks

RESOURCES/BOOKS FOR REFERENCES:

1. Cloud computing Bible by Barrie Sosinsky, Wiley India Pvt Ltd (2011)
2. Enterprise Cloud Computing: Technology, Architecture, Applications by Gautam Shroff, Cambridge University Press.
3. Cloud Computing Implementation, Management, and Security By John W. Rittinghouse,James F. Ransome , CRC Press.
4. Mastering Cloud Computing Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi
5. Dr. Kumar Saurabh,"Cloud Computing", Wiley Publication
6. . Borko Furht, "Handbook of Cloud Computing", Springer
7. Venkata Josyula,"Cloud computing – Automated virtualized data center", CISCO Press
8. Greg Schulr,"Cloud and virtual data storage networking",CRC Press

9. Mark Carlson,"Cloud data management and storage", Mc Graw hill
10. Lizhe Wang, " Cloud Computing:Methodology, System and Applications", CRC Press
11. Cloud computing: Data Intensive Computing and Scheduling by Chapman Hall/CRC
12. Christopher M. Moyer, Building Applications in the Cloud: Concepts, Patterns, and Projects
13. IBM smart storage cloud Red paper by Larry Coyne Mark Bagley Gaurav Chhaunker
14. Google Apps by Scott Granneman,Pearson
15. Cloud Security & Privacy by Tim Malhar, S.Kumaraswammy, S.Latif (SPD,O'REILLY)
16. Cloud Computing : A Practical Approach, Antohy T Velte, et.al McGraw Hill,
17. Stefano Ferretti et.al.,||QoS–aware Clouds", 2010 IEEE 3rd International Conference on Cloud Computing
18. Edge Computing: Simply In Depth [Print Replica] Kindle Edition Publication date 22 March 2019

Subject Code	Course Name	L	T	P	Total Credits
PE602B	Soft Computing	2	1	0	3

COURSE OUTCOME:

1. Describe artificial neural networks and different learning algorithms to formulate computing requirements
2. Identify soft computing techniques and their role in professional engineering practice
3. Apply fuzzy logic and reasoning to handle uncertainty in engineering problems
4. Make use of genetic algorithms to solve optimization problems
5. Demonstrate applications of soft computing

COURSE CONTENT:

Module 1: Soft Computing

Introduction to soft computing, structure and working of a biological neural network, artificial neural network, terminology, models of neurons, Basic learning laws, functional units for ANN for Pattern Recognition Task

Module 2: ANN

Different ANN Architecture, Biological Neurons & its working, Supervised Learning, Perceptron Learning Algorithms, Multilayered Network Architectures, Back propagation Learning Algorithm, Applications of feed forward neural networks, Feedback Neural Networks & Self Organizing Feature Map

Module 3: Fuzzy Logic

Basic concepts of Fuzzy Logic, linguistic variables, possibility distributions, fuzzy rules, Fuzzy sets, Operations on fuzzy sets, properties, geometric representation of fuzzy sets, possibility theory

Module 4: Fuzzy relational data models

Fuzzy Logic in database and Information systems, fuzzy relational data models, operations in fuzzy relational data Models, fuzzy object oriented databases, Fuzzy 22 information retrieval and web search

Module 5: Genetic Algorithm

Basics of Genetic algorithm, Basic GA framework and different GA architectures, Design issues, GA operators: Encoding, Crossover, Selection, Mutation, Solving single-objective optimization problems using GAs. Case studies and industrial applications

RESOURCES/BOOKS FOR REFERENCES:

1. B. Yegnanarayana , “Artificial Neural Networks”, PHI publications
2. Satish Kumar, “Neural Networks- A classroom Approach”, TMH Publication
3. John Yen, Reza Langari, “Fuzzy Logic”, Pearson Education
4. S. Rajasekaran, Vijaylakshmi Pari, “Neural networks, Fuzzy Logic and Genetic AlgorithmsSynthesis and Applications”, PHI publication.
5. Lotfi A. Zadeh, “Soft computing and Fuzzy Logic”,World Scientific Publishing Co., Inc. River Edge, NJ, USA.
6. J.S.R.Jang, C.T.Sun and E.Mizutani, .Neuro-Fuzzy and Soft Computing., PHI, 2004, Pearson Education 2004.