School of Computer & Information Sciences

Integrated MTech (Computer Science Engineering)

Curriculum (w. e. f: 2024-25)

SCHOOL OF COMPUTER & INFORMATION SCIENCES

Vision Statement:

- To invent, create and bring computing technology solutions to the common man, to the privileged and underprivileged sections of India, to bridge the digital divide and eradication of computer ignorance and digital illiteracy and to build a prosperous and technologically advanced nation.

Mission Statements:

MS-1: To pursue academic and research excellence, nationally and internationally

MS-2: To provide training, advisory, and consultancy to all the stakeholders.

MS-3: To lead the efforts in creative and newer modes of instruction delivery & supervision

School of Computer and Information Sciences

Name of the Academic Program: Integrated MTech (Computer Science and Engineering)

Program Educational Objectives (PEOs)

PEO-1: To produce graduates with strong foundational concepts, techniques and tools to enable them to be pursue higher studies.

PEO-2: To prepare students to apply engineering knowledge to solve problems in computer science and other fields.

PEO-3: To produce graduates with strong human values and professional ethics

PEO-4: Produce Post graduates who can contribute to the Research & Development effectively

PEO-5: To provide students a deep insight into cutting edge technologies and tools.

PEO-6: To create globally competent technocrats with exposure to Scientific & Engineering aspects of development

PEO-7: To work collaboratively on multi-disciplinary projects and exhibit high levels of professional & ethical values

PEO-8: Create awareness of societal problems and its impact

Note: ***PEO-1 to PEO-3 applies to I. MTech I-VI and PEO-4 to PEO-8 applies to I. MTech VII-X***

Mapping Program Educational Objectives (PEOs) with Mission Statements (MS)

	MS – 1	MS – 2	MS – 3
PEO – 1	3	2	
PEO – 2	3		2
PEO – 3	2		3
PEO – 4	3	2	1
PEO – 5	2	3	1
PEO – 6	3	2	1
PEO – 7	2	1	3
PEO – 8	1	2	3

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Name of the Academic Program: Integrated M.Tech (Computer Science and Engineering) I-VI

Program Outcomes (POs)

PO-1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and engineering specialization to the solution of the complex engineering problems.

PO-2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO-3: Design/Development of Solutions: Design solutions for complex engineering problems and design system component or processes that meets the specified needs with appropriate consideration for the public health and safety, and the cultural societal and environmental considerations.

PO-4: Conduct Investigation of Complex Problems: Use research-based knowledge and research methods including designs of experiments, analysis, and interpretation of data and synthesis of the information to provide valid conclusions

PO-5: Model tool Usage: Create, select, and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of limitations.

PO-6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice

PO-7: Environmental and sustainability: Understand the impact of the professional engineering solutions in societal and environmental context, and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

PO-9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend ad write the effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and a leader in a team, to manage projects and in multidisciplinary environments

PO-12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Mapping of Program Outcomes (POs) and Program Specific Outcomes (PSOs) with Program Educational Objectives (PEOs)

	PEO-1	**PEO-2**	**PEO-3**
PO-1	2	3	1
PO-2	3	2	1
PO-3	1	3	2
PO-4	1	3	2
PO-5	2	3	1
PO-6	3	2	1
PO-7	1	2	3
PO-8	2	1	3
PO-9	1	2	3
PO-10	3	1	2
PO-11	1	2	3
PO-12	3	1	2

Mapping of Program Specific Outcomes (PSOs) where applicable.

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

School of Computer & Information Sciences

Integrated M.Tech (CSE)

Student Induction Program: Starts with three weeks SIP that includes Universal Human Values – 1 (0 credits) before regular Classes.

SEMESTER – I				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
English (HS101)	3	HSMC	HSC	CELS
Engineering Mathematics – I (BS101)	3	BSC	SSC	School of Mathematics & Statistics
Engineering Physics – I (BS103)	3	BSC	SSC	SEST
Problem Solving and Programming (CS101)	3	ESC	DSC	SCIS
Problem Solving and Programming Lab (CS103)	2	ESC	DSC	SCIS
Principles of Engineering and Sustainability (MT101)	4	ESC	SSC	SEST
Basic Engineering Laboratory – I (ES105)	2	ESC	SSC	SEST & SCIS
GEC – I (At least 2 Credits)	2		GEC	
Total Credits, Sem – I = 22				
SEMESTER – II				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Discrete Mathematics (IE203)	3	BSC	SSC	SEST
Engineering Physics – II (BS104)	3	BSC	SSC	SEST
Engineering Mathematics – II (BS108)	3	BSC	SSC	School of Mathematics & Statistics
Creativity and Innovation (ES102)	3	EEC	OE	SEST
Data and File Structures (IE161)	3	PCC	DSC	SCIS
Data and File Structures Lab (IE162)	2	PCC	DSC	SCIS
Engineering Drawing ES104)	2	ESC	SSC	SEST
GEC – II (At least 2 Credits)	2		GEC	
Total Credits, Sem – II = 21		**Cumulative Credits at the end of Semester – II = 43**		

SEMESTER – III				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Introduction to Public Health (MS221)	3	HSMC	HSC	
Information Theory (IE210)	3	PCC	DSC	SCIS
Electronic Devices & Circuits (IE205)	3	ESC	SSC	SCIS
Object Oriented Programming (IE207)	3	PCC	DSC	SCIS
Skill Development (IE209)	3	EEC	FSE	SCIS
Object Oriented Programming Lab (IE208)	1.5	PCC	DSC	SCIS
Electronic Devices & Circuits Lab (IE206)	1.5	ESC	SSC	SCIS
Computer Organization & Architecture (IE202)	4	PCC	DSC	SCIS
Total Credits, Sem – III = 22		**Cumulative Credits at the end of Semester – III = 65**		
SEMESTER – IV				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Computer Based Numerical & Optimization Techniques (IE259)	3	BSC	SSC	SCIS
Computer Based Numerical & Optimization Techniques Lab (IE260)	2	BSC	SSC	SCIS
Data Base Management Systems (IE258)	3	PCC	DSC	SCIS
Data Base Management Systems Lab (IE261)	2	PCC	DSC	SCIS
UHV-II (IE262)	3	MC		
Theory of Computation (IE254)	3	PCC	DSC	SCIS
Environmental Sciences (IE263)	1	MC		
Constitution of India (IE264)	1	MC		
Open Elective-I(Other Schools/Swayam)	3	OEC	OE	
Total Credits, Sem – IV = 21		**Cumulative Credits at the end of Semester – IV = 86**		

SEMESTER – V				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Accounting and Financial Management (IE311)	3	HSMC	HSC	
Operating Systems (IE301)	3	PCC	DSC	SCIS
Computer Networks (IE310)	4	PCC	DSC	SCIS
Algorithms (IE304)	4	PCC	DSC	SCIS
Elective-I	4	PEC	SSE	SCIS
Operating Systems Lab (IE308)	2	PCC	DSC	SCIS
Computer Networks Lab (IE312)	2	PCC	DSC	SCIS
Total Credits, Sem – V = 22		**Cumulative Credits at the end of Semester – V = 108**		
SEMESTER – VI				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Compiler Design (IE360)	3	PCC	DSC	SCIS
Software Engineering (IE351)	3	PCC	DSC	SCIS
Internet Technologies (IE361)	3	PCC	DSC	SCIS
Humanities-I (Applied Linguistics, etc.)	3	HSMC	HSC	
Essentials of AI (IE362)	4	PCC	DSC	SCIS
Software Engineering Lab (IE355)	2	PCC	DSC	SCIS
Internet Technologies Lab (IE363)	2	PCC	DSC	SCIS
Total Credits, Sem – VI = 20		**Cumulative Credits at the end of Semester – VI = 128**		
SUMMER INTERNESHIP (IE414)		**Credits = 3**		
SEMESTER – VII				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Machine Learning (IE410)	3	PCC	DSC	SCIS
Software Project Management (IE411)	3	HSMC	HSC	SCIS
Elective-II	4	PCE	SSE	SCIS
Elective-III	4	PCE	SSE	SCIS
Cyber Security (IE412)	3	PCC	DSC	SCIS
Machine Learning Lab (IE413)	2	PCC	DSC	SCIS
Total Credits, Sem – VII (inc. Internship) = 22		**Cumulative Credits at the end of Semester – VII = 150**		

SEMESTER – VIII				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Open Elective-II(Swayam/HCU)	3	OEC	OE	
Open Elective-III(Swayam/HCU)	3	OEC	OE	
Elective-IV	4	PEC	SSE	SCIS
Elective-V	4	PEC	SSE	SCIS
Pattern Recognition (IE456)	4	PCC	DSC	SCIS
Advanced Algorithms (IE457)	4	PCC	DSC	SCIS
Total Credits, Sem – VIII = 22		**Cumulative Credits at the end of Semester – VIII = 172**		

SEMESTER – VIII (B. TECH EXIT)				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Open Elective-II(Swayam/HCU)	3	OEC	OE	
Open Elective-III(Swayam/HCU)	3	OEC	OE	
Project (IE502)	12	EEC	RI	
Total Credits, Sem – VIII = 18		**Cumulative Credits at the end of Semester – VIII = 168**		

SEMESTER – IX				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Research Methods (Swayam/HCU) (IE503)	3	HSMC	HSC	
Open Elective-IV(Swayam/HCU)	3	OEC	OE	
Disertation-1/Industrial Project (IE502)	10	EEC	RI	
Total Credits, Sem – IX = 16		**Cumulative Credits at the end of Semester – IX = 188**		

SEMESTER – X				
Course Title/Code	**Credits**	**Type of Subject (AICTE)**	**Type of Subject (UoH NEP)**	**Offering Academic Unit**
Disertation-2/Industrial Project (IE551)	16	EEC	RI	
Total Credits, Sem – X = 16		**Cumulative Credits at the end of Semester – X = 204**		

Course title: English

Code: HS101

Type: HSC/GEC

Credits: 3

Semester: I

Course	English (HS101)	**Credits**	3
Course Type	HSC/GEC		

Course Description

This course is designed to equip engineering students with the essential English language skills needed for success in their academic and professional careers. It goes beyond technical writing and communication, focusing on developing well-rounded individuals who can effectively communicate, critically analyse information and navigate the complexities of the modern world.

Course Objectives

CO-1: Communicate effectively in both technical and non-technical contexts.
CO-2: Critically evaluate digital and technical information.
CO-3: Demonstrate sensitivity to cultural and gender issues in communication.
CO-4: Apply creative and ethical perspectives in engineering practices.
CO-5: Conduct research using credible sources and structured arguments.
CO-6: Develop emotional intelligence and handle ethical dilemmas thoughtfully.

Course Learning Outcomes

CLO-1: Communicate effectively in both technical and non-technical contexts.
CLO-2: Critically evaluate digital and technical information.
CLO-3: Demonstrate sensitivity to cultural and gender issues in communication.
CLO-4: Apply creative and ethical perspectives in engineering practices.
CLO-5: Conduct research using credible sources and structured arguments.
CLO-6: Develop emotional intelligence and handle ethical dilemmas thoughtfully.

Mapped to Programme Level Outcomes

	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	2	2	3	3	3	3	2	2	2	2	2	1
CLO2	2	2	3	3	2	2	2	2	2	2	2	1
CLO3	2	2	3	3	2	2	2	2	2	2	2	1
CLO4	2	2	3	1	1	2	2	2	2	2	2	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT I: Humour in Communication

Understand the power of humour in building rapport and creating a positive work environment. Analyse the use of humour in daily communication, readings, technical writing and presentations. Explore cultural nuances and sensitivities around humour.

UNIT 2: Gender Sensitization and Inclusive Communication

Develop awareness of gender equality and inclusivity in the engineering field. Identify and challenge gender stereotypes in engineering workplaces. Practice using inclusive language and communication strategies.

UNIT 3: Digital Readership and Critical Thinking

Develop critical thinking skills to evaluate online information effectively. Identify credible sources for technical research. Practice effective research techniques for engineering topics. Observe the post-pandemic digital readership and its challenges

UNIT 4: Art, Literature and the Engineer's Perspective

Broaden perspectives and foster creativity through the lens of art and literature. Analyse the portrayal of engineers in various literary works. Explore the intersection of technology and art in modern society. Discuss the ethical implications of technological advancements.

UNIT 5: Life and its Teachings: Emotional Intelligence and Well-being

Develop emotional intelligence skills for personal and professional well-being. Discuss ethical dilemmas faced by engineers in real-world scenarios. Explore the impact of technology on society and individual well-being. Practice mindfulness and stress management techniques.

Course title: Engineering Mathematics – I

Code: BS101

Type: BSC/SSC

Credits: 3

Semester: I

Course	Engineering Mathematics – I (BS101)	**Credits**	3
Course Type	BSC/SSC		

Course Description

Engineering Mathematics - 1 is a foundational course that aims to impart essential mathematical skills and techniques relevant to various engineering disciplines to solve engineering problems. The topics covered include calculus, ordinary differential equations and linear algebra.

Course Objectives

CO-1: To Develop foundational mathematical skills for solving engineering problems.
CO-2: To Equip students with advanced techniques in calculus and partial derivatives.
CO-3: To Foster understanding of ordinary differential equations and their applications.
CO-4: To Enhance knowledge of linear algebra, including matrices and vector spaces.
CO-5: To Introduce vector calculus concepts for analyzing vector fields and integrals.
CO-6: To Apply mathematical methods to model and solve real-world engineering scenarios.

Course Learning Outcomes

CLO-1: Apply partial derivatives and optimization techniques to engineering problems.
CLO-2: Solve ordinary differential equations using various analytical methods.
CLO-3: Analyze and solve linear systems using matrix algebra and determinants.
CLO-4: Determine eigenvalues, eigenvectors, and perform matrix diagonalization.
CLO-5: Utilize vector calculus to compute gradients, divergences, and curls.
CLO-6: Apply integral theorems in vector calculus to evaluate line, surface, and volume integrals.

Mapped to Programme Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	3	3	3	3	3	2	2	2	2	1	1
CLO2	2	2	3	3	3	2	2	3	2	2	1	1
CLO3	2	3	3	3	2	2	2	2	2	2	1	1
CLO4	2	2	3	1	1	2	2	2	2	2	2	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Partial Derivatives

Partial derivatives, Chain rule, Differentiation of Implicit functions, Exact differentials. Maxima, Minima and saddle points, Method of Lagrange multipliers. Differentiation under Integral sign, Jacobians and transformations of coordinates

UNIT 2: Ordinary Differential Equations (ODEs)

Basic Concepts. Geometric Meaning of y'= f(x, y). Direction Fields, Euler's Method, Separable ODEs. Exact ODEs. Integrating Factors, Linear ODEs. Bernoulli Equation. Population Dynamics, Orthogonal Trajectories. Homogeneous Linear ODEs with Constant Coefficients. Differential Operators. Modelling of Free Oscillations of a Mass–Spring System, Euler–Cauchy Equations. Wronskian, Nonhomogeneous ODEs, Solution by Variation of Parameters. Power Series Method for solution of ODEs: Legendre's Equation. Legendre Polynomials, Bessel's Equation, Bessels's functions Jn(x) and Yn(x). Gamma Function

UNIT 3: Linear Algebra

Matrices and Determinants, Gauss Elimination, Linear Independence. Rank of a Matrix. Vector Space. Solutions of Linear Systems and concept of Existence, Uniqueness, Determinants. Cramer's Rule, Gauss–Jordan Elimination. The Matrix Eigenvalue Problem. Determining Eigenvalues and Eigenvectors, Symmetric, Skew-Symmetric, and Orthogonal Matrices. Eigenbases. Diagonalization. Quadratic Forms. Cayley – Hamilton Theorem (without proof)

UNIT 4: Vector Calculus

Vector and Scalar Functions and Their Fields. Derivatives, Curves. Arc Length. Curvature. Torsion, Gradient of a Scalar Field. Directional Derivative, Divergence of a Vector Field, Curl of a Vector Field, Line Integrals, Path Independence of Line Integrals, Double Integrals, Green's Theorem in the Plane, Surfaces for Surface Integrals, Surface Integrals, Triple Integrals, Stokes Theorem. Divergence Theorem of Gauss.

Course title: Engineering Physics – I

Code: BS103

Type: BSC/SSC

Credits: 3

Semester: I

Course	Engineering Physics – I (BS103)	**Credits**	3
Course Type	BSC/SSC		

Course Description

This course is designed to provide a fundamental understanding of the physical principles that underpin various engineering disciplines. The course covers the engineering applications built on fundamental physical principles of mechanics, and heat transfer.

Course Objectives

CO-1: To Provide a fundamental understanding of classical mechanics and its engineering applications.
CO-2: To Introduce the concepts of stress, strain, and material deformation in engineering contexts.
CO-3: To Explain the principles of thermodynamics and their applications in energy systems.
CO-4: To Develop a foundational understanding of quantum mechanics and its engineering applications.
CO-5: To Enhance problem-solving skills using physical principles across various engineering disciplines.
CO-6: To Apply theoretical concepts to analyze and design engineering systems.

Course Learning Outcomes

CLO-1: Analyze kinematic and dynamic systems using classical mechanics principles.
CLO-2: Apply stress-strain concepts to material selection and structural analysis.
CLO-3: Utilize thermodynamic laws in designing and evaluating energy systems.
CLO-4: Understand and apply quantum mechanics principles to modern engineering challenges.
CLO-5: Solve engineering problems using conservation laws and mechanical principles.
CLO-6: Integrate physical principles into the design and optimization of engineering systems.

Mapped to Programme Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	3	1	1	1	1	1	1	1	1	1	1
CLO2	3	3	1	3	1	1	1	1	1	1	1	1
CLO3	3	3	1	3	1	1	1	1	1	1	1	1
CLO4	3	3	1	3	1	1	1	1	1	1	1	1
CLO5	3	3	1	3	1	1	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Classical Mechanics

Kinematics and dynamics of particles and rigid bodies. Conservation laws. Concept of work, energy, and power, linear momentum and impulse. Rotational motion. Select examples of application in engineering systems such as pendulum, LCR oscillator, lever, etc.

UNIT 2: Engineering Mechanics

Concept of stress, strain, elasticity, plasticity, viscosity, natural frequency, moment of inertia, free body diagram. Select examples of these concepts in engineering applications such as bridges.

UNIT 3: Thermodynamics

State property, first law, reversible vs irreversible process, Carnot cycle, entropy, Clausius inequality, second law; statistical interpretation of entropy; Entropy applied in materials engineering and in information theory.

UNIT 4: Quantum Mechanics

Wave-particle duality and de Broglie hypothesis. Heisenberg's uncertainty principle. Schrodinger equation. Particle in a box model. Simple harmonic oscillator, Applications in material and computer engineering such as DFT, quantum computing, etc.

Course title: Problem Solving and Programming and Lab

Code: CS101 and CS103

Type: ESC/DSC

Credits: 3 + 2

Semester: I

Course	Problem Solving and Programming and Lab (CS101 + CS103)	**Credits**	3 + 2
Course Type	**ESC/DSC**		

Course Description

This course introduces problem-solving techniques and structured programming using the C language. It covers problem analysis, algorithm design, and solution representation through pseudocode and flowcharts. Students learn fundamental programming constructs, modular design, recursion, and data handling using arrays and strings. The course emphasizes developing correct, efficient, and well-structured programs to solve computational problems.

Course Objectives

CO1: Analyze computational problems by identifying inputs, outputs, constraints, and appropriate solution strategies.
CO2: Design and represent algorithms using structured approaches such as pseudocode and flowcharts.
CO3: Develop correct and efficient C programs using fundamental programming constructs and good coding practices.
CO4: Apply programming techniques to solve problems involving iteration, numerical computations, and basic data processing.
CO5: Implement modular and scalable solutions using functions, recursion, arrays, and basic data structures.

Course Learning Outcomes

CLO-1: Understand basic terminology of computers and programming languages and their evolution (Understand)
CLO-2: Know basic building blocks of algorithms and their programming language counterparts (Understand)
CLO-3: Design the solution from specification of a problem and write pseudo code of the algorithm. (Create)
CLO-4: Translate an algorithm into a computer program (Create)
CLO-5: Analyze programs using debugging tools. (Analyze)

Mapped to Programme Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	3	3	3	2	2	3	3	2	2	3	2
CLO2	3	3	3	3	2	2	3	3	2	2	3	2
CLO3	2	2	2	3	1	1	1	2	2	2	2	2
CLO4	2	2	3	3	1	2	3	2	2	2	2	2
CLO5	2	2	3	3	1	2	3	2	2	2	2	2

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1(CLO-1, CLO-2):

Problems And Problem Instances, Generalization and Special Cases, Types of Computational Problems, Classification of Problems, Analysis of Problems, Solution Approaches, Algorithm Development, Analysis of Algorithm, Efficiency, Correctness, Data Representation, Role of Data Structures in Problem Solving, Problem-Solving Steps (Understand the Problem, Plan, Execute, and Review), Breaking the Problem into Subproblems, Input/Output Specification, Input Validation, Pre and Post Conditions.

UNIT 2 (CLO-2, CLO-3, CLO-4):

Structured Programming Concepts: Sequence (Input/Output/Assignment), Selection (If, If-Else) And Repetition (For, While, Do-While) Statements, Control Structure Stacking and Nesting. Different Kinds of Repetitions: Entry Controlled, Exit Controlled, Counter Controlled, Definite, Indefinite and Sentinel-Controlled Repetitions. Pseudocode and Flowcharts. Definition And Characteristics of Algorithms, Standard Algorithm Format.

C Language: Introduction To Programming Languages, Different Generations of Programming Languages. Typed Vs Typeless Programming Languages, History of C Language, An Empty C Program. C Language Counterparts For Input (scanf()), Output (printf()) Statements, Assignment, Arithmetic, Relational and Logical Operators. C statements for If, If-Else, For, While, Do-While Statements. Data Types. Translating Pseudocode/Algorithm to C Program. Incremental Compilation and Testing of The C Program. Simple Problems Involving Input, Output, Assignment Statement, Selection and Repetition Statements. Good Coding Practices.

UNIT 3 (CLO-2, CLO-3, CLO-4):

Problems Involving Iteration and Nesting: Displaying Different Patterns and Shapes Using Symbols and Numbers, Generating Arithmetic and Geometric Progression, Fibonacci and Other Sequences, Approximate Values For π, Sin(x), Cos(x), Etc. Using Taylor Series.

UNIT 4 (CLO-2, CLO-3, CLO-4):

Problems on Numbers: Extracting Digits of a Number (Left to Right and Right to Left), Palindrome, Prime Number, Prime Factors, Amicable Number, Perfect Number, Armstrong Number, Factorial, Converting Number from One Base to Another. Statistics (Maximum, Minimum, Sum and Average) on a Sequence of Numbers which are Read using Sentinel-Controlled Repetition using only a few Variables.

C Language: else-if Ladder, switch Case, Increment/Decrement Operators, break and continue Statements.

UNIT 5 (CLO-2, CLO-3, CLO-4, CLO-5):

Modular Programming, Top-Down and Bottom-Up Approaches to Problem Solving. Recursion. Problems on Arrays: Reading and Writing of Array Elements, Maximum, Minimum, Sum, Average, Median and Mode. Sequential And Binary Search. Any one Sorting Algorithm, Merging, Matrix Operations, Character Arrays (Strings): String Functions, User Defined Data Types and Their Use Cases.

C Language: Function Definition and Declaration (Prototype), Role of Return Statement, One Dimensional and Two-Dimensional Arrays. String Functions. Other Operators, Operator Precedence and Associativity. Debugging and Makefile.

Text Books

1. Harvey Deitel and Paul Deitel, C How to Program, 9th edition, Pearson India, 2015.
2. R G Dromey, How to Solve It by Computer, First Edition, Pearson India, 2007.

Reference Books

1. Brian W. Kernighan and Dennis Ritchie, The C Programming Language, 2nd edition, Pearson, 2015.
2. Jeri Hanly and Elliot Koffman, Problem Solving and Program Design in C, 8th edition, Pearson, 2015.
3. Stephen Kochan. “Programming in C”, Pearson Education India.

Suggested Lab Exercises:

1. Converting *degrees Celsius to Fahrenheit and vice versa?*
2. *Display three input numbers in sorted (non-decreasing) order?*
3. Given a positive integer value n (>= 0) display number, square and cube of numbers from 1 to n in a tabular format?
4. Given an input positive integer number, display the odd numbers in the range [1, n]?
5. Display the first 10 mathematical tables, each table up to 10 rows? Generalise this to display first n (> 0) mathematical tables up to m (m > 0) rows?
6. Display following patterns of n rows (n > 0), For the below examples n = 5? For each pattern write a separate algorithm/program?

$	$	12345	12345
$$	$$	1234	1234
$$$	$$$	123	123
$$$$	$$$$	12	12
$$$$$	$$$$$	1	1

7. Display the following patterns of n rows (n > 0), for the below examples n = 5?

Hollow square pattern:	Triangle Patterns with numbers:	Square with diagonals:	Diamond Pattern
#####		* * * * *	*
# #	1	* * * *	***
# #	121	* * *	*****
# #	12321		***
#####	1234321	* * * *	*
	123454321	* * * * *	

8. Given the first term (a), difference/multiplier (d) and number of terms (n > 0), display the first n terms of the arithmetic/geometric progression?
9. Display the first n (n > 0) terms of the Fibonacci sequence?
10. Display the first n (n > 0) terms of the Tribonacci sequence?

11. Given two positive integer numbers n1 and n2 check if the numbers are consecutive numbers of the Fibonacci sequence?
12. Compute approximate value of π considering first n (n > 0) terms of the Taylor series for π?
13. Compute approximate value of e^x considering first n (n > 0) terms of the Taylor series for e^x?
14. Compute approximate value of sin(x)/cos(x) considering first n (n > 0) terms of the Taylor series for sin(x)/cos(x)?
15. Extract digits of a positive integer number (left to right and right to left)?
16. Given a sequence of digits, form the number composed of the digits. Use sentinel-controlled repetition to read the digits followed by -1. For example, for the input digits 2 7 3 2 9 -1 the output number is 27329?
17. Check if a given positive integer number is a palindrome or not?
18. Compute character grade from the marks ($0 \leq$ marks ≤ 100) of a subject. Grading Scheme: 80-100: A, 60 - 79: B, 50 - 59: C, 40-49: D, 0-39: F? Solve this using both else-if ladder and switch case?
19. Compute the sum of a sequence of numbers entered using sentinel-controlled repetition?
20. Check if a given positive integer number is a prime number or not?
21. Compute prime factors of a positive integer number?
22. Check if two numbers n1 and n2 are co-primes or not?
23. Check if two positive integer numbers are amicable numbers or not?
24. Check if a given positive integer number is a perfect number or not?
25. Check if a given positive integer number is an Armstrong number or not?
26. Converting a positive integer number (n > 0) from one base (inputBase) to another base (outputBase) (2 <= inputBase, outputBase <= 10). Input number should be validated before converting to make sure the number uses only digits allowed in the input base?
27. Write a program to display a number in text form. For example, If the number is 543200 the output should be “FIVE FOUR THREE TWO ZERO ZERO”?
28. Using the grading scheme described in the question 18, compute how many students awarded each grade and display the frequency as a bar chart (horizontal) using single “*” for each student. Use sentinel-controlled repetition (-1 as sentinel value) in reading the students marks. Use else-if ladder/switch case to compute the grade and the corresponding frequency.
    Sample bar chart when the class has 7-A, 10-B, 3-C, 7-D and 1-F grades.

    A: *******

    B: **********

    C: ***

    D: *******

F: *

29. Compute maximum, minimum, sum and average of a sequence of numbers which are read using sentinel-controlled repetition using only few variables?
30. Compute body mass index, BMI = weightinKGs / (HeightinMeters *HeightinMeters), Both weight and height values are positive real numbers. Your program should display BMI value followed by whether the person is Underweight, Normal, Overweight or Obese using the below ranges:

    BMI Values

    Underweight: less than 18.5

    Normal: >=18.5 and <25

    Overweight: >=25 and < 30

    Obese: >= 30
31. Design a modularized algorithm/program to check if a given positive integer number is a circular prime or not?
32. Design a modularized algorithm/program to compute a maximum of 8 numbers?
33. Design a modular algorithm/program which reads an array of n integer elements and outputs mean (average), range (max-min) and mode (most frequent elements)?
34. Design a modular algorithm/program which reads an array of n integer elements and outputs median?
35. Implement your own string length and string reversal functions?
36. Design algorithm/program to perform matrix operations addition, subtraction and transpose?
37. Write a recursive program to count the number of digits of a positive integer number?
38. Recursive solutions for the following problems:
    a. Factorial of a number?
    b. Display digits of a number from left to right (and right to left)?
    c. Compute x^y using only multiplication?
    d. To print a sequence of numbers entered using sentinel-controlled repetition in reverse order?

Course title: Principles of Engineering and Sustainability

Code: MT101

Type: PCC/SSC

Credits: 4

Semester: I

Course	Principles of Engineering and Sustainability (MT101)	**Credits**	4
Course Type	PCC/SSC		

Course Description

This is an introductory course designed to provide an overview of the basic principles involved in engineering across disciplines, with focus on the process employed in design and problem solving. This course covers both the classical methods and modern practices along with an emphasis on sustainability.

Course Objectives

CO-1: To Introduce the foundational principles of engineering across various disciplines.
CO-2: To Explain the engineering design process and problem-solving methodologies.
CO-3: To Familiarize students with fundamental engineering concepts and practices.
CO-4: To Highlight the importance of sustainability in engineering design and practice.
CO-5: To Examine the historical evolution of engineering and its impact on society.
CO-6: To Explore modern engineering practices including data analytics and energy management.

Course Learning Outcomes

CLO-1: Understand the basic principles and definitions related to science, engineering, and technology.
CLO-2: Apply the engineering design process to solve problems with consideration of constraints.
CLO-3: Utilize fundamental engineering concepts such as dimensional analysis in practical applications.
CLO-4: Evaluate sustainability using the sustainability index and life cycle analysis.
CLO-5: Analyze the impact of the industrial revolutions on the environment and engineering practices.
CLO-6: Implement modern engineering practices in automation, optimization, and sustainability efforts

Mapped to Programme Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	1	1	2	1	1	1	1	1	1	1	1
CLO2	3	1	2	2	1	1	2	1	1	1	1	1
CLO3	3	2	1	1	1	1	1	1	1	1	1	1
CLO4	2	2	1	1	1	1	1	1	1	1	1	1
CLO5	2	2	1	1	1	1	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Introduction

What is science, engineering and technology. History of engineering and key contributions of engineering across different disciplines. What engineers do?

UNIT 2: Building Blocks of Engineering

Understanding engineering through simple machines and devices – Inclined plane, wedge, screw, levers, pulley, wheel and axle, gear, piston, transformer and transistor.

UNIT 3: Engineering Problem Solving

Basic steps in engineering problem solving. Need-Know-How-Solve approach. Select case studies of problem solving covering various engineering disciplines.

UNIT 4: Engineering Design and Practice

Dimensional analysis and units. Standards and codes. Fundamental principles of engineering design. Steps in engineering design. Select case studies of the design process covering various engineering disciplines.

UNIT 5: Sustainability

Industrial revolution 1.0-4.0 and its impact on environment. Significance of sustainable technology. Quantification of sustainability-by-sustainability index. Life cycle analysis. Determining total energy consumption.

Course title: Basic Engineering Laboratory – I

Code: ES105

Type: ESC/SSC

Credits: 2

Semester: I

Course	Basic Engineering Laboratory - 1 (ES105)	**Credits**	2
Course Type	ESC/SSC		

Course Description

This course provides hands on foundational training on basic aspects of computer, electrical and electronics engineering.

Course Objectives

CO-1: To Provide foundational training in computer hardware and software
CO-2: To Introduce basic concepts and practices in electrical wiring.
CO-3: To Teach fundamental electronics skills, including circuit design and measurement.
CO-4: To Develop hands-on experience with electrical and electronics components.
CO-5: To Familiarize students with practical applications of series and parallel circuits.
CO-6: To Equip students with skills to build and test basic electronic circuits.

Course Learning Outcomes

CLO-1: Understand the components and functions of computer hardware and software
CLO-2: Perform basic electrical wiring tasks, including one-way and two-way switch setups.
CLO-3: Use a multimeter for accurate electrical measurements.
CLO-4: Design and assemble circuits on a breadboard using various components.
CLO-5: Analyze and implement series and parallel resistor configurations.
CLO-6: Build and test circuits incorporating LEDs, resistors, capacitors, transistors, and ICs.

Mapped to Programme Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	3	3	3	2	2	3	3	2	2	3	2
CLO2	3	3	3	3	2	2	3	3	2	2	3	2
CLO3	2	2	2	3	1	1	1	2	2	2	2	2
CLO4	2	2	3	3	1	2	3	2	2	2	2	2
CLO5	2	2	3	3	1	2	3	2	2	2	2	2
CLO6	2	2	2	2	1	2	3	3	3	2	2	2
CLO7	2	2	2	2	1	2	3	3	3	2	2	2
CLO8	3	3	3	3	2	2	3	3	2	2	3	2

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

Electrical Lab

1. Fundamentals of electrical circuitry and safety: A practical approach
2. Single switch and light bulb connection and wiring
3. Two-Way switch and wiring
4. Building an extension box with multiple switches and sockets
5. Basic residential wiring with multiple loads

Electronics Lab

1. Basics of electronic components and wiring with a bread board
2. Voltage and current division in series and parallel circuits
3. Building logic gates using switches
4. Understanding the working principle of transistors in simple circuits
5. Building a water level indicator using IC555

IT Lab

1. Opening up of a computer and identification of computer peripherals and components: Appreciating the block diagram along with the configuration of each component and its functionality: Input/ Output devices, I/O ports and interfaces, main memory, cache memory and secondary storage technologies, network, graphics cards etc.
2. Understanding the directory structure of Linux and ideas of open source
3. Learning basic Linux commands
4. Installing the Linux operating system
5. Installing Application software (Eg. Jupyter notebook, Calibre etc.) and package management. Exploring Libre Office packages.

Course title: Discrete Mathematics

Code: IE203

Type: BSC/SSC

Credits: 3

Semester: II

Course	Discrete Mathematics (IE203)	**Credits**	3
Course Type	BSC/SSC		

Course Description

This course introduces fundamental concepts of Discrete Mathematics that is importsnt in Coputer science and Engineering. It covers topics such as Sets, Relations, Logic, Boolean Functions, Graphs and Trees etc which are relevant to various Computer Science Topics.

Course Objectives

CO-1: To Introduce fundamental concepts of Logic and its application.
CO-2: To Explain concepts of Graphs and Trees along with their role in CS.
CO-3: To Explore Sets and Relations and relevant concepts such as Lattices.
CO-4: To Teach about recursion and its usefulness in problem solving.
CO-5: To Explain concepts of Boolean Functions and their role in Circuits.
CO-6: To Introduce relevance above concepts in Computer Science applications.

Course Learning Outcomes

CLO-1: Apply predicate and propositional logic to represent and solve problems. (Apply)
CLO-2: Discuss various ways of simplification and apply the same on minimizing logical circuits. (Understand)
CLO-3: Using principle of recursion, be able to frame a real-world situation as a recurrence relation and solve. (Apply).
CLO-4: Describe counting principles (Understand)
CLO-5: Apply counting principles in real world scenarios. (Apply)
CLO-6: Describe graphs and trees techniques (Understand)
CLO-7: Apply the graphs and trees techniques to solve the real time problems (Apply)

	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1		3	2			1						
CLO2	1		3		2							
CLO3	2	3			1							
CLO4	3			2	1							
CLO5	1		2	3								
CLO6	3		1			2						
CLO7			1		2	3						

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Sets, Relations and Functions

Sets, relations and functions; Methods of proof; Equivalence relations; Cardinality; Countable and uncountable sets.

UNIT 2: Introductory Logic

Fundamentals of Logic; Logic operators such as AND, OR etc., Truth tables; Logical inferences; Methods of proofs of an implication; First order logic; Predicate calculus Predicates and Quantifiers; Rules of inference for quantified propositions.

UNIT 3: Recurrence Relations

Recursion, Forming and solving recurrence relations by substitution method and generating functions; Method of characteristic roots; solving inhomogeneous recurrence relations.

UNIT 4: Boolean Algebra

Partial order relations; Lattices; Boolean algebra; Combinatorial circuits; Minimization of Boolean functions using Karnaugh maps

UNIT 5: Theory of Graphs

Graphs, subgraphs, isomorphism, proofs; Types of graphs; paths and cycles; Adjacency matrices; Transitive closure; Connectivity; Directed acyclic graphs; Planar graphs and Euler's formula; Dual of a graph; Hamiltonian and Eulerian graphs; Applications like matching and colouring graphs; Graph traversals (BFS and DFS); Trees; Spanning trees.

Reference Books

1. Kenneth H Rosen (2012), "Discrete Mathematics and Its Applications",7th Edition, McGraw Hill, NY
2. Ralph P Girimaldi(2003), "Discrete and Combinatorial Mathematics –An Applied Introduction", 5th Edition, Pearson Addison Wesley, Indian Edition
3. J.R Mott, A Kandel, T.P Baker (2015), "Discrete Mathematics for Computer Scientists and Mathematicians", Pearson
4. Ronald L Graham, Donald E Knuth, Oren Patashnik(1994), "Concrete Mathematics- A Foundation of Computer Science", 2nd Edition, Addison Wesley .
5. Susanna S. Epp(2010), "Discrete Mathematics with Applications", 4th Edition, Brooks/Cole Cengage Learning.

Course title: Engineering Physics – II

Code: BS104

Type: BSC/SSC

Credits: 3

Semester: II

Course	Engineering Physics -II (BS104)	**Credits**	3
Course Type	BSC/SSC		

Course Description

This course is designed to provide a fundamental understanding of the physical principles that underpin various engineering disciplines. The course covers the engineering applications built on fundamental physical principles of optics, electromagnetism, semiconductors etc.

Course Objectives

CO-1: To Provide a foundational understanding of wave optics and its engineering applications.
CO-2: To Explain the principles of electricity and magnetism relevant to engineering.
CO-3: To Introduce the fundamentals of semiconductor physics and electronic devices.
CO-4: To Explore the basic concepts of nuclear physics and their applications.
CO-5: To Develop problem-solving skills using physical principles in engineering contexts.
CO-6: To Demonstrate the application of these physical principles in modern engineering technologies.

Course Learning Outcomes

CLO-1: Understand wave motion, optics, and apply them in areas like fibre optics and lasers.
CLO-2: Analyze electric and magnetic fields, and apply them in circuits, motors, and generators.
CLO-3: Apply semiconductor principles to the design of rectifiers, amplifiers, and oscillators.
CLO-4: Comprehend atomic and nuclear physics concepts and their engineering applications.
CLO-5: Solve engineering problems using principles of electromagnetism and semiconductor physics.
CLO-6: Design and analyze electronic circuits using transistors, FETs, and digital electronics.

	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	2	2	2	2	1	1	1	1	1	1	1	1
CLO2	3	3	3	2	1	1	1	1	1	1	1	1
CLO3	2	2	2	2	1	1	1	1	1	1	1	1
CLO4	2	2	2	2	1	1	1	1	1	1	1	1
CLO5	3	3	3	3	1	1	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Optics

Wave motion: types of waves and their properties. Wave equation. Concepts of reflection and refraction and applications in lenses, and mirrors. Basics of interference and diffraction of light. Select examples of application of these concepts in the area of acoustics, optics including fibre optics and lasers.

UNIT 2: Electricity and Magnetism

Review of basic concepts of electric charge, electric fields, and Gauss's law. Electric potential, capacitance, and dielectrics. Electric current, resistance, and Ohm's law. Magnetic fields, magnetic forces, and Ampere's law. Electromagnetic induction and Faraday's law. Select examples of application of these concepts in design of circuits, motors, generators, etc.

UNIT 3: Semiconductors

Introduction to semiconducting materials, p-n junction diode and its functionality, Diode equation, half-wave, full wave and bridge rectifiers with voltage regulation circuits. Clipping, clamping, voltage doubler and multipliers. Transistors NPN, PNP, CE, CB and CC configurations, transistor characteristics CE amplifier designing, RC Oscillator, FET amplifier design, MOSFET and Introduction to Digital electronics.

UNIT 4: Nuclear Physics

Basic concepts in atomic and nuclear physics- atomic structure, radioactivity, and nuclear reactions. Applications in nuclear engineering and medical field.

Course title: Engineering Mathematics – II

Code: BS108

Type: BSC/SSC

Credits: 3

Semester: II

Course	Engineering Mathematics – II (BS108)	**Credits**	3
Course Type	BSC/SSC		

Course Description

Engineering Mathematics - 2 is a foundational course that aims to impart essential mathematical skills and techniques relevant to various engineering disciplines to solve engineering problems. The topics covered include probability statistics and partial differential equations.

Course Objectives

CO-1: To Develop a strong foundation in complex analysis and its engineering applications.
CO-2: To Introduce fundamental concepts in probability and statistics for engineering problems.
CO-3: To Equip students with skills in integral transformations, including Laplace and Fourier transforms.
CO-4: To Teach the application of partial differential equations (PDEs) to model and solve physical problems.
CO-5: To Enhance problem-solving abilities using advanced mathematical techniques.
CO-6: To Apply mathematical methods to analyze and solve real-world engineering challenges.

Course Learning Outcomes

CLO-1: Understand and apply complex analysis, including Cauchy's theorems and series expansions.
CLO-2: Use probability and statistical methods in engineering applications like quality control.
CLO-3: Apply Laplace and Fourier transform to solve ordinary differential equations (ODEs).
CLO-4: Model and solve physical problems using partial differential equations (PDEs).
CLO-5: Analyze and interpret statistical data in the context of engineering reliability and risk analysis.
CLO-6: Utilize integral transformations in solving complex engineering problems.

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	1	1	2	1	1	1	1	1	1	1	1
CLO2	3	1	2	2	1	1	2	1	1	1	1	1
CLO3	3	2	1	1	1	1	1	1	1	1	1	1
CLO4	2	2	1	1	1	1	1	1	1	1	1	1
CLO5	3	2	1	2	1	2	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Complex Analysis

Complex Numbers and Their Geometric Representation, Polar Form of Complex Numbers. Powers and Roots, Derivative. Analytic Function, Cauchy–Riemann Equations. Laplace's Equation, Exponential Function, Trigonometric and Hyperbolic Functions. Euler's Formula, de 'Moivre's theorem, Logarithm. General Power. Principal Value. Singularities and Zeros. Infinity, Line Integral in the Complex Plane, Cauchy's Integral Theorem, Cauchy's Integral Formula, Derivatives of Analytic Functions, Taylor and Maclaurin Series.

UNIT 2: Probability and Statistics

Probability concepts - sample space, events, probability rules. Discrete and continuous probability distributions - binomial, normal etc. Statistical measures - mean, variance, standard deviation. Statistical inference and hypothesis testing. Applications of probability and statistics in engineering - reliability, quality control and risk analysis.

UNIT 3: Integral Transformations

Laplace Transforms: Definitions and existence (without proof), properties, First Shifting Theorem (s-Shifting), Transforms of Derivatives and Integrals and ODEs, Convolution. Integral Equations, Differentiation and Integration of Transforms. Solution of ODEs with Variable Coefficients, Solution of Systems of ODEs. Inverse Laplace transform and its properties. Fourier Analysis: Fourier Series, Arbitrary Period. Even and Odd Functions. Half-Range Expansions, Fourier Integral, Fourier Cosine and Sine Transforms, Fourier Transform. Usage of F ODEs. Inverse Fourier transform and its properties.

UNIT 4: PDEs and Applications

Basic Concepts of PDEs. Modelling: Vibrating String, Wave Equation. Solution by Separating Variables. Use of Fourier Series. D'Alembert's Solution of the Wave Equation. Characteristics. Modelling: Heat Flow from a Body in Space. Heat Equation: Solution by Fourier Series. Steady Two-Dimensional Heat Problems.

Course title: Creativity and Innovation

Code: ES102

Type: ESC/OE

Credits: 3

Semester: II

Course	Creativity and Innovation (ES102)	**Credits**	3
Course Type	ESC/OE		

Course Description

This course is aimed to expose the students to the importance of creativity and innovation in science, engineering, and technology. To achieve high sustainability scores in technology it is important to be creative and innovate. This course shall motivate the students to understand and practice creativity and innovation.

Course Objectives

CO-1: To Highlight the importance of creativity and innovation in science, engineering, and technology.
CO-2: To Explain the different types of innovation, including process, product, and business model innovations.
CO-3: To Teach methods for thoroughly understanding and defining engineering problems.
CO-4: To Introduce techniques for generating new ideas and overcoming barriers to innovation.
CO-5: To Explore stages and examples of innovation in science, engineering, and technology.
CO-6: To Emphasize the importance of mind management and mental attitudes in fostering creativity.

Course Learning Outcomes

CLO-1: Understand and define creativity and innovation, and recognize their importance in technology.
CLO-2: Identify and categorize different types of innovation with relevant examples.
CLO-3: Apply methods to systematically understand and define problems.
CLO-4: Utilize various techniques for generating new ideas and overcoming innovation barriers.
CLO-5: Analyze and apply the stages of innovation in scientific and engineering contexts.
CLO-6: Develop strategies for mind management to enhance creativity and innovation skills.

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	1	1	1	3	2	1	1	1	1	1	1
CLO2	1	1	1	1	3	1	1	1	1	1	1	1
CLO3	1	1	1	1	2	1	1	1	1	1	1	1
CLO4	1	1	1	1	3	1	1	1	1	1	1	1
CLO5	2	1	1	1	2	1	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Introduction

Definitions of Creativity and Innovation; Bringing new ideas related to practices, procedures, products, etc. into existence; Types of innovation: process innovation, product innovation, service innovation and business model innovation.

UNIT 2: Systematic Understanding of a Problem

Importance of thoroughly understanding a problem; How to understand a problem? Methods for defining a problem (Kipling, problem statement and challenge methods) with examples.

UNIT 3: Techniques for Creating New Ideas

When do we get new ideas? How do we get new ideas? Why is it required to innovate? Important techniques: Attribute listening, Brainstorming, Brainwriting, Brainfaxing, and Visioning with examples; Conceptual blocks and barriers for innovation.

UNIT 4: Innovation in Science, Engineering, and Technology

Stages in innovation (preparation, verification/validation, incubation and realization) with examples; Innovation in science (question on nature, inquiry and discovery methods, evidence, proposed explanation, solution, usefulness to the society); Innovation in engineering and technology (questions on problems adapting to the environment, design and invention strategies, evidence, proposed explanation, verification, validation, solution directly useful to the society).

UNIT 5: Mind Management

Importance of mind management; Human mind; Left and right Brian; super conscious mind; shaping rightful mental attitude; any memory game.

Course title: Data and File Structures and Lab

Code: IE161 and IE162

Type: PCC/DSC

Credits: 3

Semester: II

Course	Data and File Structures and Lab (IE161 and IE162)	**Credits**	3 + 2
Course Type	PCC/DSC		

Course Description

This course introduces fundamental data structures and their implementation using C, including arrays, linked lists, stacks, queues, trees, and graphs. It emphasizes algorithm design, time complexity analysis, and efficient data organization. The course also covers searching, sorting techniques, hashing, and file structures such as B-trees and B+ trees for handling large datasets.

Course Objectives

CO-1: Discuss which data structures are used for static and dynamic allocations. (Understand)

CO-2: Solve the problem where in elements can be traversed in either direction and select the suitable data structure for this idea using C/Java Programming Language (Apply)

CO-3: Analyze the time taken to solve the given problem by using C/Java programming language (Analyze)

CO-4: Assess the solution in terms of efficiency, modularity and well-documented programs in C/Java under Linux environment (Evaluate)

Course Learning Outcomes

CLO-1: Solve a given problem by choosing appropriate data structures (Apply).

CLO-2: Select suitable data structure for given idea and propose an appropriate solution (Understand)

CLO-3: Analyze the time taken to solve a given problem (Analyze)

CLO-4: Assess the solution in terms of performance and standard programming principles under Linux environment (Evaluate)

THEORY

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1			2	3	1							
CLO2					3	2					1	
CLO3				3	2		1					
CLO4	1			2	3							

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1:

Introduction to data structures and data types: Primitive and Non-Primitive types, Arrays- Sparse matrix, Stacks, Queues, Circular queues, Priority queues, Dequeues, Conversions and Evaluations of expressions, Polynomial representation using arrays, Time complexity analysis of algorithms with respect to data structure operations.

UNIT 2:

Linked Lists: Linked stacks and queues, Circular and Doubly linked lists, Polynomial representation using linked lists.

UNIT 3:

Trees and Graphs: Binary Trees, Tree Traversal, Binary Search trees and basic operations, Heaps, AVL Trees, height balanced trees, Graphs – Representation of the graphs, Graph Traversals.

UNIT 4:

Sorting and Searching Mechanism: Selection sort, Bubble sort, Insertion sort, Merge sort, Quick sort, Heap sort. Linear Search, Binary Search, Hash Tables.

UNIT 5:

File structures: Concepts of Double Buffering and Block Buffering, Indexing, B-tree needs, properties, creations and Uses, B+ trees

Reference Books

1. Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.
2. Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.
3. Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.
4. Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill.
5. Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in C; Prentice - Hall of India, 1999.
6. The C Programming Language by Brian W.Kernighan , Dennis M. Ritchie
7. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.
8. Schaum"s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.
9. Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

10. Michael J. Folk and Bill Zoellick, “File Structures” (Second Edition).
11. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition Addison-Wesley, 1997.
12. Schaum‟s Outline Series, “Data Structure”, TMH, Special Indian Ed., Seventeenth Reprint, 2009.
13. Mary E. S. Loomes, “Data Management and File Structure”, PHI, 2nd Ed., 1989.

LAB

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1					3	2					1	
CLO2	1	2		3								
CLO3				3	2		1					
CLO4	1			2	3							

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1:

Implementation of Stacks and different types of Queues data structures using arrays, Conversions and Evaluations of expressions.

Suggested Assignments:

- Large integer arithmetic. Arithmetic operations are to be performed on very large integers of N digits (where $0 < N < 20$). Write a C program that performs the operations of multiplication and division on such large integers.
- Two stacks using a single array. Implement two stacks using a single array such that neither overflows unless the total number of elements in both the stacks is equal to the size of the array.
- Deque. Deque is a queue which allows insertions and deletions at both ends. Write a C program that implements deque using both arrays and linked lists. Each implementation should provide support for basic operations such as (i) Insertion of an element at both ends (ii) Deletion of an element from both ends (iii) Displaying all elements of deque.
- Evaluation of arithmetic expression. Write a C program that reads an input arithmetic expression in (INFIX, PREFIX, POSTFIX) notations and outputs its result.
- Arithmetic expression notation conversion. Write a C program that reads an input arithmetic expression in infix notation (fully parenthesized) and converts it into an output arithmetic expression in postfix notation.

UNIT 2:

Implementation of basic Linked List operations such as addition, updation, deletion, searching and traversal of all elements of the list.

Suggested Assignments:

- Implement Stack data structure (using linked list) operations. Write a C program to perform the basic operations on the stack.
- Implement Queue data structure (using linked list) operations. Write a C program to perform the basic operations on the queue.
- Doubly Linked List operations. Linked list (doubly) is an important data structure for dynamic allocation wherein elements can be traversed by either direction. Write a C program to perform the basic operations on the linked list.

- Union and Intersection. Given two linked lists of numbers, write a program that finds a resultant linked list which is union of the two input linked lists and another resultant linked list which is intersection of the two input linked lists.
- Detecting cyclic linked lists. Write a C program that detects whether a given linked list is cyclic or not, if yes, then return the node where the cycle begins.

UNIT 3:

Implementation of Trees and Graphs of basic operations. Implementation of Adjacency Matrix and List Representation. Breadth and Depth First Search.

Suggested Assignments:

- Tree traversals. Write C program to display a tree using all the methods of traversals: (i) Inorder traversal, (ii) Preorder traversal, (iii) Postorder traversal
- Binary Search Tree. Binary Search Tree is an important data structure for dynamic allocation and optimized searching. Write C program to perform the basic operations on binary search tree (BST): (i) Adding, (ii) Updating (iii) Deleting (iv) Search for an element (v) Displaying all elements (in-order).
- Building heap. Using a C program, build a max-heap, given N random integers. Display the heap thus formed in its in-order form.
- Heap sort. Using the heap data structure, sort the given N random integers.
- AVL Trees. Using a C program, perform the following operations on the AVL tree data structure: (i) Inserting, (ii) Deleting, (iii) Update, (iv) Searching, (v) Displaying an element
- Topological Sorting. In a university curriculum, often each course has a set of pre- requisites. Given a set of courses along with their respective set of pre-requisites, prepare a curriculum such that no course appears before its prerequisite.
- Using Linked List and BST: Construction of a city database using a linked list and binary search tree and the appropriateness of these structures under various demands for the data.

UNIT 4:

Sorting and Searching Mechanism: Linear Search, Binary Search, Implementation of Bubble, Insertion, Quick, Selection Sort.

Suggested Assignments:

- Write C programs to perform both linear and binary search on a given random set of integers. The following points should perform by the program
- Take as input an integer, N, which would decide number of integers to be processed and another input an integer, X ($0 < X < N+1$), which is the key to be searched
- Randomly generate N integers whose values are between 1 to N, multiple entries are allowed
- Output all the indexes (positions) of key in given set of random integers

- Count number of comparisons in the linear and binary searching process, please note comparisons involved in sorting process (in case of binary search) are not to be included
- Output the result in following table: -

Input Size (N)	Number of Comparisons	
	Linear Search	Binary Search
10		
30		
50		
70		
100		

Merge Sort. Logging activity files of two users are given as input, merge them into a single file. Assume that the format of the logging activity file is two columns with the first column representing the date-time record and second column the event description. Merging is to be done with respect to the date-time record field.

UNIT 5: File Structures: Implementation of B Trees and B+ Trees

File structures: Concepts of Double Buffering and Block Buffering, Indexing, B-tree needs, properties, creations and Uses, B+ trees

Suggested Assignments:

- Improved file copying. Modify the file copy program to avoid overwriting the existing target file, instead if the target file has some contents, then target file is appended by contents of source file.
- File handling API. Write a program to read and write a file using following combinations of functions: -
    - fgetc( ) and fputc( )
    - fprintf( ) and fscanf( )
    - fgets( ) and fputs( )
    - fread( ) and fwrite( )

Course title: Engineering Drawing

Code: ES104

Type: ESC/SSC

Credits: 2

Semester: II

Course	Engineering Drawing (ES104)	**Credits**	2
Course Type	ESC/SSC		

Course Description

This course designed to impart basic skills in visual communication and graphical representation of engineering concepts using a CAD software. Through a combination of lectures, demonstrations, and hands-on exercises, students will learn fundamental principles of engineering drawing, computer-aided design (CAD), and technical sketching.

Course Objectives

CO-1: Develop basic skills in visual communication and graphical representation using CAD software.
CO-2: Introduce fundamental principles of engineering drawing and technical sketching.
CO-3: Teach geometric constructions and their applications in engineering graphics.
CO-4: Explain orthographic and isometric projections for representing engineering designs.
CO-5: Demonstrate methods for drawing sections of solids and visualizing internal features.
CO-6: Apply engineering drawings to practical applications including machine drawings and 3D printing.

Course Learning Outcomes

CLO-1: Understand and apply engineering drawing principles and CAD software for graphical representation.
CLO-2: Perform basic geometric constructions and implement them using CAD software.
CLO-3: Create and interpret orthographic projections and understand their relationships.
CLO-4: Draw and interpret isometric projections of simple solids using CAD software.
CLO-5: Represent sections of solids and visualize internal features in engineering drawings.
CLO-6: Apply drawing techniques to machine design, fabrication, and 3D printing applications.

Mapped to Program Level Outcomes

	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	2	2	2	2	1	1	1	1	1	1	1	1
CLO2	2	2	2	2	1	1	1	1	1	1	1	1
CLO3	2	2	2	2	1	1	1	1	1	1	1	1
CLO4	2	2	2	2	1	1	1	1	1	1	1	1
CLO5	2	2	2	2	1	1	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Introduction to Engineering Drawing & Basic Geometrical Constructions

Introduction to the field of engineering graphics and its applications in various engineering disciplines. Importance of engineering drawings in the design and manufacturing process. Introduction CAD software. Engineering lettering and dimensioning practices according to standards. Lines, angles, and their classifications. Construction of triangles, quadrilaterals, and other polygons. Methods for construction of circles and tangents, ellipses and parabolas. Using CAD software to implement the same.

UNIT 2: Orthographic Projection

Principles of orthographic projection (first-angle and third-angle projections). Drawing projections of points, lines, and planes in various orientations. Relationship between views and true lengths and angles. Projections of simple solids (prisms, pyramids, cylinders, cones).

UNIT 3: Sections of Solids

Methods for representing sections of solids in orthographic drawings. True shapes of sections obtained by cutting planes. Applications in visualizing internal features and designing components.

UNIT 4: Isometric Projection

Introduction to isometric projection as a pictorial representation. Isometric axes and scaling principles. Drawing isometric views of simple solids from orthographic projections using CAD software. Conversion between isometric and orthographic projections.

UNIT 5: Application of Drawings

Machine drawings, conventional fabrication and 3D printing.

Course title: Introduction to Public Health

Code: MS221

Type: HSMC/HSC

Credits: 3

Semester: III

Course	Introduction to Public Health	**Credits**	3
Course Type	HSMC/HSC		

Course Description

This course will aid to examine the evolution of definitions and concepts in the understanding of public health. It will be a means for improving critical thinking to understand public health as an evolving, multifaceted practice. This course introduces the core disciplines and the history and philosophy of public health. It aims to illustrate the nature, role, and organization of the public health response using historical examples of important public health challenges. In the realm of public health, technology supports the ways in which professionals can gather and analyse information and provide improved care to communities. Technologists, specifically engineers, have vast opportunities to create and use dynamic public health technology solutions that can have a profound impact on population health and patient care.

This course is proposed to be taught through a combination of approaches: lectures and class discussions. The set of readings and written assignments will broaden the student's understanding of topics covered in class and reinforce the concept of multidisciplinary integration in public health and engineering technology.

Course Objectives

CO-1: Describe key features of the historical development of public health, including the most important achievements of public health.
CO-2: Identify and describe core functions of public health and the role and contributions of each of the core disciplines in public health.
CO-3: Describe important public health problems facing society including health disparities and how technology can address it
CO-4: Describe the various components of the national, state, and local public health systems and what engineers and technologists can do.
CO-5: Understand the determinants of health from a global perspective, including environmental, social, cultural, behavioural, and biological factors in developing and implementing health care technology.
CO-6: Outline the concepts of prevention, detection, and control of diseases from the technology perspective.
CO-7: Explore with examples how technology aids in protecting and improving the health of communities with greater efficiency.

Syllabus

UNIT 1: Introduction to Human body, Health and Disease

1. Human body and its various systems
2. Concept of health and disease,
3. Natural history of disease,
4. Levels of prevention

UNIT 2: Introduction to Public Health

1. Definition of Public Health and Associated Terms
2. Current Concerns in Public Health: Global and Local
3. Understanding of current public health challenges from their shared experiences or regions
4. Role of Humanities and Social Sciences in Public Health

UNIT 3: History and Evolution of Public Health

1. History of public health
2. Evolution of Public Health
3. Core functions of public health.
4. Scope of public health.

UNIT 4: Health and Determinants

1. Dimensions of Health
2. Social determinants of health
3. Health disparities
4. Health among vulnerable populations

Course title: Information Theory

Code: IE210

Type: PCC/DSC

Credits: 3

Semester: III

Course	Information Theory	**Credits**	3
Course Type	PCC/DSC		

Course Description

This course introduces the fundamentals of probability and information theory, including entropy, source coding, and data compression techniques. It covers channel capacity, coding theorems, and error-correcting codes for reliable communication. The course also provides an overview of modern cryptographic techniques, including symmetric and public-key encryption, for secure information transmission.

Course Objectives

CO-1: To develop the student's ability to understand the concept of information theory.
CO-2: To provide the students about various codes used for data compression.
CO-3: To develop the student's ability to analyse the error correcting codes used for reliable transfer of data.
CO-4: To familiarize the student with the various decoding techniques.
CO-5: To introduce the elementary cryptographic algorithms used in information theory.

Course Learning Outcomes

CLO-1: Able to acquire knowledge about concept of mutual information and entropy in information theory.
CLO-2: Able to acquire knowledge about various data compression codes
CLO-3: Able to understand and analyse the various error correcting codes used for reliable transfer of data.
CLO-4: Able to understand and analyse the decoding techniques.
CLO-5: Able to understand and analyse some of the cryptographic algorithms used in information theory.

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	2	0	0	0	0	0	0	0	0	0	0
CLO2	3	3	1	0	0	0	0	0	2	0	1	0
CLO3	3	3	3	1	2	0	2	0	2	0	0	0
CLO4	2	3	3	1	2	2	2	0	2	0	1	1
CLO5	2	3	3	1	2	2	2	0	2	0	3	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1:

Review of Probability Theory: Elementary Theorems on Probability, Random Variables, Probability distributions, Expectations, independence of random variables, Joint distributions of continuous RVs, Random Processes.

UNIT 2:

Introduction to Information Theory: Uncertainty, Information, Concept of mutual information, Entropy and their properties, Kraft's Inequality, Source Coding Theorem, Huffman Coding, Lempel-Ziv Algorithm, Introduction to Image Compression

UNIT 3:

Channel Capacity and Coding: Channel Models, Channel Capacity, Channel Coding, Shannon's Laws, The Shannon Limit.

UNIT 4:

Introduction to Error Correcting Codes: Linear Block Codes, Decoding, Hamming Codes, Maximum Distance Separable Codes, Cyclic Codes, BCH Codes

UNIT 5:

Introduction to Cryptography, Encryption Techniques, DES scheme, RSA scheme, Diffie & Hellman's Public Key Agreement Protocol, Secure Communication Using Chaos Function, Quantum Cryptography

Reference Books

1. Information Theory, Coding & Cryptography, Ranjan Bose 3rd Edition, McGraw Hill Education (India) Private Limited
2. Information Theory, Coding & Cryptography, Arijit Saha et al, Pearson
3. The Theory of Information and Coding, Robert McEliece, Cambridge

Course title: Electronic Devices and Circuits and Lab

Code: IE205/IE206

Type: ESC/SSC

Credits: 3 + 1.5

Semester: III

Course	Electronic Devices and Circuits and Lab (IE205 and IE206)	**Credits**	3 + 1.5
Course Type	**ESC/SSC**		

Course Description

This course introduces the fundamentals of semiconductor devices, including p–n junction diodes and their applications. It covers special diodes, bipolar junction transistors, biasing techniques, and amplifier operations. The course also explores field-effect transistors (JFET and MOSFET), their characteristics, and applications in electronic circuits.

Course Objectives

CO1: To understand the fundamental concepts of semiconductors and p–n junction diodes and their characteristics.
CO2: To study special diodes and their applications in rectifiers and filtering circuits.
CO3: To analyze the operation and characteristics of bipolar junction transistors and their use as amplifiers and switches.
CO4: To understand transistor biasing techniques and ensure thermal stability in electronic circuits.
CO5: To explore the operation, characteristics, and applications of FETs and MOSFETs.

Course Learning Outcomes

CLO-1: Explain the fundamental differences between insulators, semiconductors, and metals based on energy band theory. (Understand)
CLO-2: Design and analyze half-wave and full-wave rectifiers using diodes, including inductive and capacitive filters for AC to DC conversion (Analyze)
CLO-3: Analyze BJT configurations (CE, CB, CC) and their characteristics in terms of input-output behaviour. (Analyze)
CLO-4: Evaluate different biasing techniques and their stability against variations in key transistor parameters (VBE, Ic, β) and implement bias compensations (Evaluate)
CLO-5: Apply FET and MOSFET in practical circuits and applications, understanding their advantages and limitations over BJTs (Apply)

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	1	3	2	1	1	1	1	1	1	1	1
CLO2	2	1	3	2	1	1	1	1	1	1	1	1
CLO3	2	1	2	3	1	1	1	3	1	1	1	1
CLO4	2	1	2	3	1	1	1	1	2	1	1	1
CLO5	2	1	1	2	3	1	1	1	1	1	1	1

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Semiconductors and p-n Junction Diodes

Insulators, semiconductors and metals; Energy gap; Intrinsic and extrinsic semiconductors; free and bound electrons; majority and minority carriers; continuity equation and diffusion currents; p-n junctions and diodes; Quantitative analysis of diodes; Forward and reverse biases

UNIT 2: Special Diodes and Applications

Zener diode; Breakdown mechanisms; Photodiode, Varactor diode, Tunnel diode and Schottky diode; Applications of diodes: halfwave and full wave rectifiers; Inductive and capacitive filters

UNIT 3: Transistors and Amplifiers

Bipolar junction transistor, transistor current components; Transistor equation; CE, CB and CC configurations; transistor as a switch, transistor as an amplifier; Input and output characteristics; Transistor parameters: current amplification factor, relation(s) between α, β, γ.

UNIT 4: Transistor biasing and thermal stability

Need for biasing, operating point, load line analysis: DC and AC load lines; Basic stability, fixed, collector-to-base and self-biases; Stability against variations in VBE, Ic, β; Stability factors and bias compensations.

UNIT 5: FETs and MOSFETs

FET types and symbols; JFET, MOSFET; JFET: n- and p-channel construction, operation and characteristics; Drain and transfer parameters, drain resistance, amplification factor, transconductance, pinch-off voltage; MOSFET: n-and p-channel, enhancement MOSFET; comparison between JFET and MOSFET; applications.

Reference Books

1. J. Millman and C. Halkias. Electronic Devices and Circuits, Tata McGraw-Hill, Fourth Edition (2017).
2. Allan Mottershead. Electronic Devices and Circuits: An Introduction, Prentice-Hall India (2011).

Suggested Lab Excercises

5. UNIT 1 and 2:
    - To construct a half-wave rectifier circuit and observe the rectified output.
    - To construct a full-wave rectifier using two or four diodes (center-tapped) and observe the rectified output.
    - To construct a full-wave rectifier using two or four diodes (bridge configuration) and observe the rectified output.
    - To analyze the effect of capacitive filters on the output of a rectifier circuit.
    - To study the forward and reverse bias characteristics of a Zener diode and use it for voltage regulation.
6. UNIT 3, 4 and 5:
    - To study the input and output characteristics of a BJT in CE configuration.
    - To study the input and output characteristics of a BJT in CB configuration.
    - To study the input and output characteristics of a BJT in CC configuration
    - To demonstrate the use of a Transistor as an electronic switch
    - To demonstrate the amplification property of a BJT in CE configuration.

Course title: Object Oriented Programming and Lab

Code: IE207 and IE208

Type: PCC/DSC

Credits: 3 + 1.5

Semester: III

Course	Object Oriented Programming and Lab	**Credits**	3 + 1.5
Course Type	**PCC/DSC**		

Course Description

This course introduces object-oriented programming concepts using Java, including classes, objects, inheritance, polymorphism, and abstraction. It covers core Java features such as data types, control structures, arrays, strings, exception handling, and collections. The course also includes file handling and GUI programming using Swing, enabling students to design modular, reusable, and event-driven applications.

Course Objectives

CO-1: Understand object-oriented principles through Java programming. (Understand)
CO-2: Design classes for a given Computational problem (Create)
CO-3: Create Java programs for the object-oriented design of the given problem. (Create)
CO-4: Develop software solutions using encapsulation, inheritance, polymorphism, and abstraction. (Create)
CO-5: Apply Java libraries for exception handling, file I/O, and GUI development (Apply)

Course Learning Outcomes

CLO1: Explain object-oriented programming concepts and Java programming fundamentals.
CLO2: Develop Java programs using classes, objects, methods, arrays, and strings.
CLO3: Apply inheritance, polymorphism, abstraction, and design principles in software development.
CLO4: Utilize exception handling, generics, and collections for efficient program design.
CLO5: Build applications using file handling and GUI programming with event-driven techniques.

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	2	3	1	1	1	1	1	1	1	1	1
CLO2	1	1	1	2	3	1	1	1	1	1	1	1
CLO3	1	1	1	2	1	1	1	1	1	1	1	3
CLO4	1	1	3	2	1	1	1	1	1	1	1	1
CLO5	1	1	1	1	2	1	1	1	1	1	1	3

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1:

Evolution of Programming Paradigms, Procedural vs Object-Oriented, Advantages of OOP over Procedural Programming. Introduction to OOP, Core Principles: Abstraction, Encapsulation, Inheritance, Polymorphism. Object and Class Concepts. Introduction to Java: JVM, JDK, Basic Syntax, Data Types, Variables, Operators, Control Flow Statements: if, switch, loops, Methods and Parameter Passing.

UNIT 2:

Classes and Objects in Java. Encapsulation: Access Modifiers, Getters/Setters. Constructors: Default, Parameterized, Overloaded. Static Members and Initialization Blocks. Method Overloading and Recursion. Usage of this and final keywords. Array, ArrayList, String, StringBuilder and Regular Expression. Java Packages, Java Garbage Collection.

UNIT 3:

Inheritance in Java: Single, Multilevel, Hierarchical, super keyword and constructor chaining. Method Overriding and Dynamic Dispatch. Abstract Classes and Interfaces. Introduction to Polymorphism. Object-Oriented Design Principles: SOLID Principles (Single Responsibility, Open/Closed, etc.), Cohesion and Coupling.

UNIT 4:

Association, Aggregation, and Composition, Dependency relationships, Exception Handling in Java:Try-catch-finally, Throw/Throws, Custom Exceptions. Generic Classes and Methods. Java Collections Framework:List, Set, Map, Iterators, Generics.

UNIT 5:

Java File I/O, Sequential, Random Access, Object Serialization, Deserialization, Externalization.

GUI Programming with Swing:Components (JFrame, JButton, JTextField, etc.), Event Handling.

Reference Books

1. Java How to Program (Early Objects), Paul Deitel and Harvey Heitel, Pearson Education, Eleventh Edition.
2. Java A Beginner's Guide, Herbert Schildt, McGrawHill, Eighth Edition.
3. Object Oriented Programming with Java, Timothy Budd, Pearson Education.
4. Herbert Schildt, Java: The complete reference, McGraw hill.
5. Bruce Eckle, Thinking in Java, Prentice Hall.

Suggested Lab Excercises

1. Write a Java program that takes three numbers a, d, and n as input, where: a is the first term of the arithmetic progression (AP), d is the common difference and n is the number of terms and print the first n terms of the arithmetic progression.
2. Given a positive integer value n (>= 0) display number, square and cube of numbers from 1 to n in a tabular format?
3. Display the first n (n > 0) terms of the Tribonacci sequence?
4. Write a java Class to represent a pair of integers? The class should include a method to check if the pair of numbers are coprime or not?
5. Write a Java program to perform operations of two rational numbers. You need to write a class to represent rational numbers. Rational numbers are of the form p/q where p and q are integers and q != 0. Before representing the rational number, you need to make sure that the rational number is in simplified form. A rational number is in simplified form if its p and q are co-primes. For example, the rational number 8/12 is not in simplified form because 8 and 12 are not co-primes, they have a common factor. The simplified form of 8/12 is 2/3.

   Your program should read four integer numbers a, b, c and d for rational numbers a/b and c/d. Display the simplified form of a/b and c/d and addition, subtraction, multiplication and division of the two rational numbers. All the resulting rational numbers should be in simplified form.
6. Write Java programs to perform operations Complex numbers. The Complex class has two integer instance variables: rp and ip to represent the real and imaginary parts of the complex number. Provide methods to add, subtract, multiply and divide two complex numbers. Write a test class to demonstrate the functionality of the Complex class.
7. Create a class to store and operate on dates. The Date class will have three attributes, day (1-31), month (1-12) and year (positive integer > 0). Provide constructors to initialize the date with given day, month and year values. Proper input validation on the date should be done before initialising the date. If date data is invalid set all the attributes to zero, and display a message that the date is invalid. Provide the following methods to operate on Date.
   - Provide set and get methods for attributes.
   - Compare () – compares two dates return -1/0/+1 for date smaller, equal and greater respectively
   - Provide methods to display date in different formats (dd/mm/yyyy or mm/dd/yyyy)
   - Method to get the day of the year, for example 1st January 2020 is day 1 and 1st March 2020 is day 61

   Write a test class to demonstrate the capabilities of the Date class. Read value for both the dates and display INVALID DATE if any of the dates is invalid. Display date1 (dd/mm/yyyy) format followed by date2 (mm/dd/yyyy) format and display whether date1 (</=/>) date and display day number of date 1 and date2.
8. Write a Java program to operate on sets of integer numbers. Use java one dimensional array to represent a set. The following operations should also be supported. You need to define a class Set to represent a Set.
   - getCardinality

- Union
- Set difference
- Symmetric Difference

Write a test class to demonstrate the functionalities of Set class. Your program should read two sets (A and B) sizes and the set elements and display the symmetric difference of A and B. (A Δ B = ((A-B) U (B-A))). Note that a set can be an empty set.

9. Implement a BankAccount class to simulate basic bank operations such as deposit, withdrawal, and account display. Write a test class UseBankAccount class to test the functionality of the BankAccount class. The BankAccount class will have the following Instance Variables:
    - accountName: The name of the account holder (String).
    - accountNumber: A unique identifier for each account (String). This should be generated automatically (e.g., using a static counter).
    - pinNumber: A 4-digit PIN chosen by the user for security (int).
    - depositAmount: The current balance in the account (double).
    - interestRate: The interest rate for the account (double).
    - staticCounter: A static counter to generate unique account numbers, initialise it to 1000.

    The BankAccount Class will have the following methods: Constructor, deposit(), withdraw(), toString() method, checkBalance().
10. A popular game of chance is a dice game known as Craps which is played in casinos. The rules of the game are as follows: A player rolls two dice. After the dice have come to rest, the sum of the values on the two upward faces is calculated. If the sum is 7 or 11 on the first throw, the player wins. If the sum is 2, 3 or 12 on the first throw, the player loses (i.e., the "house" wins). If the sum is 4, 5, 6, 8, 9 or 10 on the first throw, that sum becomes the player's points. To win, the player must continue rolling the dice until he reaches his winning points. Player loses by rolling a 7 before making his winning points. Write a java program which simulates the Craps game. Design Dice as an object with rollDice() function to simulate the dice?
11. Design a Java application which simulates card simulation and dealing. Playing card deck consists of 52 cards, with 13 face values (Ace, Deuce, Three, …Nine, Jack, Queen, King) and 4 suits (Hearts, Diamonds, Clubs and Spades). Create a class called Card, which models a card. A deck of cards can be generated using an array of 52 Card objects. Simulate the Shuffle and Dealing process as methods.
12. Airline Reservations System) A small airline has just purchased a computer for its new automated reservations system. You've been asked to develop the new system. You're to write an application to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your application should display the following alternatives: Please type 1 for First Class and Please type 2 for Economy. If the user types 1, your application should assign a seat in the first class section (seats 1–5). If the user types 2, your application should assign a seat in the economy section (seats 6–10). Your application should then display a boarding pass indicating the person's seat number and whether it's in the first-class or economy section of the plane. Use a one-dimensional array of primitive type boolean to represent the seating chart of the plane. Initialize all the elements of the array to false to indicate that all the seats are empty. As each seat is assigned, set the corresponding element of the array to true to indicate

that the seat is no longer available. Your application should never assign a seat that has already been assigned. When the economy section is full, your application should ask the person if it's acceptable to be placed in the first-class section (and vice versa). If yes, make the appropriate seat assignment. If no, display the message "Next flight leaves in 3 hours."

13. You need to represent and perform operations on matrices. Each matrix can be visualized as an object. Each object has data (matrix elements and its size) and some operations can be defined on the objects. Representing Matrix object data: A two dimensional array can be used to represent the elements of the matrix. If the size of the matrix is M X N, then, we need a two dimensional array of size M X N. Initializing the object: The following information is needed to create a matrix. Number of Rows (M), Number of Columns (N), Actual matrix elements. Provide the following ways to initialize the matrix (Constructors): Square matrix: Provided the value N (Matrix size is N X N), a null square matrix of size N X N can be created. Rectangular Matrix: Provided the values M and N (Matrix size is M X N), a null matrix of size M X N can be created Set and Get methods: Both the constructors initialize all the matrix elements to zero. You have to provide a set method to modify the matrix elements. You should also provide a get method to access the matrix elements. Operations on the matrix (methods):
    - To compute transpose of a matrix (Transpose is also another matrix)
    - To perform addition of two matrices (Addition of two matrices is also another matrix)
    - Provide a toString() method to display the matrix in a tabular format.

    Your program should read the dimension of the matrices A and B and the elements of the matrix A followed by the elements of matrix B and display transpose of (A+B).

    Additional Functionality: Multiplication of two matrices, Compute the determinant of the matrix (It is easy to solve this problem using recursion) (Your function should work for matrices of any size).
14. When we perform arithmetic operations on integers, there may be overflow as the size of the integer is limited. The maximum and minimum values we can represent in an integer are also limited. One way to overcome this limitation is to store integer number as a String. Write a java program to represent Huge Integers and perform operations on Huge Integers. The operations include:
    - Addition of two huge integers
    - Comparison of two huge integers (Two huge integers can be):
        i. Both equal
        ii. One is less than the other
        iii. One is greater than the other
    - Display the huge integer

    Design a functional class called HugeInteger which encapsulates the data and behavior of a HugeInteger. Provide a test class to demonstrate the operations of the HugeInteger.
15. Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point (with x and y coordinates) class to represent the points in each shape. Include methods to compute perimeter and area of the objects. Utilize the advantage of inheritance in providing these methods in the classes. Inheritance facilitates availability of super

class member variables and methods in the sub class, write a method in the sub-class only if it is necessary. If the super class already provides a method for a particular task then the same functionality is available to the subclass as well.

16. Write java programs for the following tasks. Write a java program FileReverse.java which reads a text file passed as command line argument and generates another text file (output file name should also be taken from command line) which contains the same contents as the input file but the words in reverse order.

    Example:

    Input file contents:

    School of Computer and Information Sciences

    University of Hyderabad

    Hyderabad

    Output file contents:

    Sciences Information and Computer of School

    Hyderabad of University

    Hyderabad

17. Object Serialization of Java provides a mechanism to write an entire object into a file and later the same object can be read from the file. This way the objects can be saved onto a file and later can be cloned from the file. Write a Java class to create a Serializable file which consists of true/false questions along with the answers. Each question in the question bank can be treated as an object with fields:

    - Question (String)
    - Answer (Boolean)

    Your Question bank creation program should read questions along with the answers from the user and write the same to a file as serializable objects.

    Write a TakeTest class which reads the questions from the file and conducts the test for the user. Once the test is complete your program should display the score.

    Additional Functionality: Make the Test class to display all the questions with answers and specify whether the user has answered correctly or not.

18. Write a Java program to read an array of n (n>0) array elements and write the elements on to a java random access file using java byte stream. The file will consist of the number of elements of the array followed by the array's elements. Now write an update function to update an element of the array at a specific index. You can use Java random access file processing to update the number without disturbing the other elements of the array. Simulate a polymorphic product catalog with discounts and taxes.

19. An abstract class Product with:

    - String name
    - double price
    - abstract methods: calculateDiscount(), calculateTax()

    Subclasses:

    - Electronics – 15% tax, variable discount
    - Clothing – 5% tax, fixed discount
    - Grocery – 2% tax, no discount

    Interface Returnable with method returnPolicy(). Only Electronics and Clothing implement it.

    In Main, calculate final price = price - discount + tax using a list of Product references.

20. Write a Java program for Smart Vehicle Management System with the following specification. Create an abstract class Vehicle with the following:
    - Protected variables: brand, speed,
    - A constructor to initialize brand and speed,
    - An abstract method accelerate (int increment)
    - A concrete method displayInfo() that prints brand and current speed.

    Define two interfaces:
    - Electric with method chargeBattery()
    - FuelBased with method refuel().

    Inheritance and Implementation: Create two concrete classes:
    - ElectricCar that extends Vehicle and implements Electric. It Overrides accelerate(int increment) to increase speed and implements chargeBattery() (print "Charging battery of ElectricCar").
    - PetrolBike that extends Vehicle and implements FuelBased. It Overrides accelerate(int increment) to increase speed. Implements refuel() (print "Refueling PetrolBike").

    Demonstrate the run-time polymorphism by creating objects of ElectricCar and PetrolBike and calling their functions.
21. Write a java program which reads a text file and displays a table of the number of occurrences of each different word (token) in the text file. The name of the text file to be passed as a command line argument? For Example: For the text file containing the below two lines:

    Java is fun and powerful.

    Java is also popular and versatile.

    Output would be:

WORD	FREQUENCY
Java	2
Is	2
Fun	1
And	2
Powerful	1
Also	1
Popular	1
Versatile	1

22. Implement a generic single linked list in Java with the following methods:
    - insertFirst()
    - insertLast()
    - insertAtIndex()
    - removeFirst()
    - removeLast()
    - removeAtIndex()

- Number of elements
- Maximum element

Now write a menu driven Java test class to demonstrate the working of doubly linked list.

The following operations to be applied iteratively to the doubly linked list of integers.

- Insert element e at the head of the list.
- Insert element e at the tail of the list.
- Insert element e at the index i. Index starts from 0. If a linked list currently has n elements then 0 to n are the valid indices for i.
- Delete the element from the head of the list and display the deleted element.
- Delete an element from the tail of the list and display the deleted element.
- Delete element at index i. If the linked list currently has n elements then 0 to n-1 are the valid indices for i.
- Display the list from head to tail.
- Display number of elements
- Display the maximum element of the list
- Exit
- At any iteration if input is invalid display "INVALID INPUT" and stop the program.
- At any time remove function is called on an empty list display "LE"

23. Implement A generic sorting method which sorts any comparable objects?
24. Implemented Radix sorting technique to sort positive integer numbers. Use Java Generic List Collection to solve the problem?
25. A file contains email ids. Write a java program using Java Hashmap to remove duplicate email ids from the file.
26. Create a graphical calculator that can perform basic arithmetic operations: addition, subtraction, multiplication, and division.

Course title: Skill Development

Code: IE209

Type: EEC/FSC

Credits: 3

Semester: III

Course	Skill Development (IE209)	**Credits**	3
Course Type	EEC/FSC		
Course Description			
Course Objectives			
The students to learn the basic tools related to Web Development and Mobile App Development and Data Visualization. 1. Website Development low code or no code approaches – Tools such Viz, Canva, Figma 2. Mobile App Development – FlutterFlow Bubble, WaveMaker, MIT app Inventor 3. Data Visualization – Flourish,Excel plus Gen AI Data Visualization			
Course Learning Outcomes			

Course title: Computer Organization & Architecture

Code: IE202

Type: PCC/DSC

Credits: 4

Semester: III

Course	Computer Organization & Architecture	**Credits**	4
Course Type	PCC/DSC		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Demonstrate arithmetic operations and assess their performance (Apply). CLO-2: Describe basic Instruction Set Architecture (ISA) (Understand) CLO-3: Explain the basic pipelining of instructions (Understand) CLO-4: Examine how the memory hierarchy has impact on performance of software. (Analyze) CLO-5: Describe Interrupt handling and DMA access for performing I/O.(Understand)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	2		3			1						
CLO2	2	3			2							
CLO3	3	2			1							
CLO4		2		1		3						
CLO5	3	2	1									

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Computer Evolution & Arithmetic

A Brief History of computers, Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, Signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm, Hardware Implementation, Division, Restoring and Non Restoring algorithms, Floating point representations, IEEE standards, Floating Ppoint Arithmetic.

UNIT 2: The Central Processing Unit

Machine Instruction characteristics, types of operands, types of operations, Addressing modes, Instruction formats, Instruction types, Processor organization, Processor as running example, Programmers model of , max/min mode, Register Organization, Instruction cycles, Read Write cycles, assembly instruction examples to explain addressing modes

UNIT 3: The Control Unit

Single Bus Organization, Control Unit Operations: Instruction sequencing, Micro operations and Register Transfer. Hardwired Control: Design methods – State table and classical method, Design Examples - Multiplier CU. Micro-programmed Control: Basic concepts, Microinstructions and micro- program sequencing

UNIT 4: Memory Organization

Characteristics of memory systems, Internal and External Memory, Types of memories: ROM: PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM, High-Speed Memories: Cache Memory, Organization and Mapping Techniques, Replacement Algorithms, Cache Coherence, Virtual Memory: Main Memory allocation, Segmentation, Paging, Address Translation Virtual to Physical. Secondary Storage: Magnetic Disk, Tape, DAT, RAID, Optical memory, CDROM, DVD

UNIT 5: I/O Organization

Input/ Output Systems (features and principles), Programmed I/O, Interrupt Driven I/O, Interrupt structure, Direct Memory Access (DMA), features Buses and standard Interfaces: Synchronous, Asynchronous, Parallel I/O features,

Serial I/O features, PCI, SCSI, USB Ports Working mechanisms of Peripherals: Keyboard, Mouse, Scanners, Video Displays, Touch Screen panel, Dot Matrix, Desk-jet and Laser Printers.

UNIT 6: Case Studies

Concepts RISC: Instruction execution characteristics, RISC architecture and pipelining. RISC vs CISC. ARM and Embedded Systems PowerPC, Intel X86 Evolution from 32bit to 64bit architectures. AMD Opteron.

Reference Books

1. Patterson D.A. & Hennesy J.L., Computer Organisation & Design: The Hardware/Software Interface.
2. Computer Organization and Architecture, 10/E William Stallings ISBN-10: 0134101618 • ISBN-13: 9780134101613

Course title: Computer Based Numerical & Optimization Techniques and Lab

Code: IE259 and IE260

Type: BSC/SSC

Credits: 3 + 2

Semester: 4

Course	Computer Based Numerical & Optimization Techniques and Lab	**Credits**	3 + 2
Course Type	**BSC/SSC**		
Course Description			
Course Objectives			
CO-1: To study fundamental numerical and algorithmic techniques for solving mathematical problems using computers. It includes topics like solving linear and nonlinear equations, interpolation, differentiation, integration, and the numerical solution of ordinary differential equations. CO-2: To study various optimization techniques for linear programming problem, Transportation, Assignment Problems, Scheduling Problem, PERT and CPM. CO-3: To learn about numerical concepts like solving equations, interpolation, integrations, Linear Programming and different optimization techniques			
Course Learning Outcomes			
CLO-1: Understanding the fundamentals of numerical techniques CLO-2: Understanding different methods for solving numerical problems like simultaneous equations, Interpolation, integration, Ordinary differential equations CLO-3: Understanding various mathematical applications in industries. CLO-4: Solving linear programming problems CLO-5: Decision making for real-life environment			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	2	1	1	2								
CLO2	3	1	1	2								
CLO3				3	2							
CLO4			2					2				
CLO5			2	3				2				

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1:

- Introduction to significant digits and errors, Solution of system of linear Equations (direct methods, Iterative methods, Ill-conditioned systems)
- Roots of Nonlinear Equations (Bisection method, Regula-Falsi method, Newton-Raphson method, Fixed point iteration method, convergence criteria
- Eigenvalues and Eigenvectors, Jacobi method, Power methods

UNIT 2:

- Interpolation (Finite difference operators, difference tables, Newton's Forward/Backward difference, Lagrange interpolation and Newton's divided difference interpolation
- Numerical Differentiation (Using Forward/ Backward/central difference formula)
- Integration (Trapezoidal and Simpson's rules for integration)
- Solution of first order and second order ordinary differential equations (Euler method, Euler modified method, Runge-Kutta methods)

UNIT 3:

- Linear Programming - Mathematical Model assumption of linear Programming – Graphical method - Principles of Simplex method, Big-M Method, Duality, Dual simplex method.

UNIT 4: Memory Organization

- Transportation and Assignment problem, Scheduling problem
- PERT & CPM - Network representation - backward pass - Forward pass -computation – PERT Network

Reference Books

1. Jain, M. K., Iyengar, S. R. K. and Jain, R. K., "Numerical Methods for Scientific and Engineering Computation", New Age Pvt. Pub, New Delhi.

2. Krishnamurthy, E. V. & Sen, S. K., "Applied Numerical Analysis", East West Publication.
3. V. Rajaraman, “Computer Oriented Numerical Methods”, 4th Edition, PHI Publications
4. Operations Research, Manmohan, P.K. Gupta, Kanthi Swarup, S. Chand & Sons - 1997.
5. Operations Research, Hamdy A Taha, Pearson Education, 7th edition, 2002
6. Problems in Operations Research, P.K. Gupta, D.S. Hira, S. Chand Publishers

Course title: Data Base Management System and Lab

Code: IE258 and IE261

Type: PCC/DSC

Credits: 3 + 2

Semester: 4

Course	Data Base Management System	**Credits**	3
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Understand the basic concepts of database management systems (Understand) CLO-2: Design the database systems using ER and EER Models (Create) CLO-3: Use SQL to query the database systems (Apply) CLO-4: Evaluate the database design aspects by considering normalization principles (Evaluate) CLO-5: Explain indexing and hashing mechanisms used in database systems (Understand) CLO-6: Understand the transaction management in database systems (Understand)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	2	3	2	1	1	1	2	1	2	1	2
CLO2	3	2	1	2	3	1	2	2	2	1	1	3
CLO3	1	3	1	1	3	2	2	2	2	2	1	2
CLO4	1	3	3	3	2	2	1	2	3	1	2	2
CLO5	3	1	2	1	1	2	2	1	2	1	1	2
CLO6	3	1	2	2	1	1	1	2	1	2	1	2

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

UNIT 1: Introduction

Data and Information, Overview of File Management System, Database Systems Versus File Systems, Levels of Data Abstraction, Data Models, Database Languages- Data Definition Language, Data Manipulation Language, Data Control Language, Transaction Control Language, Functions of Database Administrator, Taxonomy of Database Users. Database System structure

UNIT 2: Data Models

Entity Relationship model- Basic Concepts, Constraints, Keys, Design Issues, E-R Diagram, Weak Entity Sets, Extended E-R Features, Design of E-R Database Schema, Reduction of E-R Schema to tables. Relational Model- Structure of Relational Database, Normalization, Functional Dependencies, Decomposition, Desirable properties of Decomposition, Normal Forms, Denormalization for Performance

UNIT 3: Structured Query Language (SQL)

Basic structure, Set operations, Aggregate functions, Null Values, Nested Subqueries, Views, Complex queries, joins Modification of database schema, Integrity Constraints. Relational Algebra & Relational Calculus.

UNIT 4: Application Design & Development

Application Design & Development: Discussion on complex data types like semi structured data, object based data, temporal data and spatial data. Big data analytics and how those applications place distinct demand on data management compared with traditional database applications (NoSQL, MapReduce, Apache Spark, streaming data and graph databases.

UNIT 5: Storage and File Structures

File organization, Organization of Records in Files, Data Dictionary Storage. Indexing and Hashing: Basic Concepts, Ordered Indices, Static Hashing, Dynamic Hashing, Comparison of Ordering Indexing and Hashing, Index Definition in SQL, Multiple-Key Access

UNIT 6: Transaction Management

ACID properties, Transaction State, Concurrent Executions, Serializability, Recoverability, Transaction Definition in SQL, Concurrency Control – Lock-Based Protocols, Timestamp-Based Protocols.

Reference Books

1. A. Silberschatz, H. F. Korth and S. Sudarshan, Database System Concepts, 7th Edition McGrawHill Publications
2. R. Elmasri, S. B. Navathe: Fundamentals of Database Systems, 7th Edition, Pearson Publication.
3. Hector Garcia-Molina, Jeffrey D Ullman, Jennifer Widom: Database Systems The Complete Book, Second Edition, Pearson, Prentice Hall.

Course	Data Base Management System Lab	**Credits**	2
Course Type			
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Design and implement database schema using RDBMS concepts (Create) CLO-2: Demonstrate physical and logical data independence (Apply) CLO-3: Use SQL to query the database systems (Apply) CLO-4: Demonstrate procedural data manipulation language constructs (Apply) CLO-5: Develop an application using SQLite and Duckdb features (Create) CLO-6: Develop a database application (Create)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	2	3	3	3	2	1	2	3	2	3	2
CLO2	1	2	3	3	3	2	1	2	3	2	3	2
CLO3	1	2	3	3	3	2	1	2	3	2	3	2
CLO4	1	2	3	3	3	2	1	2	3	2	3	2
CLO5	1	2	3	3	3	2	1	2	3	2	3	2
CLO6	1	2	3	3	3	2	1	2	3	2	3	2

'3' – 'High-level' Mapping; '2' – 'Medium-level' Mapping; '1' – 'Low-level' Mapping

Syllabus

1. Introduction to SQL. Features of SQL, DDL Statements and DML commands. Writing simple SQL queries using DDL statements and DML commands
2. Introduction to inner, outer and natural joins. Writing nested queries and correlated nested queries to retrieve and update the data. Writing SQL queries using EXISTS, NOT EXISTS, explicit join operation, aggregate functions, group by and having clauses.
3. Creating virtual tables (views). Using views in SQL queries.
4. PL/SQL programming.
5. Getting started with sqlite3 : https://sqlite.org/cli.html#getting_started /
6. Includes changing output formats, querying the database schema, opening database files, redirecting I/O, Accessing zip archives as database files, converting an entire database to a text file, Index recommendations (SQLite Expert), working with multiple database connections etc.
7. Working with DuckDb : https://duckdb.org/
8. Loading dataset into DuckDb using the SQLite Scanner.
9. Mini project.

Course title: UHV – II

Code: IE262

Type: MC

Credits: 3

Semester: 4

Course	UHV – II	**Credits**	3
Course Type	MC		

Course Description

Course Objectives

CO-1: To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.

CO-2: To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.

CO-3: To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

CO-4: This course is intended to provide a much-needed orientational input in value education to the young enquiring minds.

Course Learning Outcomes

Module 1:

CLO-1: Define core concepts such as right understanding, relationship, and physical facility within the context of holistic human development. (Remember)

CLO-2: Illustrate the purpose and significance of value education through the process of self-exploration. (Understand)

CLO-3: Apply the principles of continuous happiness and prosperity to analyse personal and societal aspirations. (Apply)

CLO-4: Assess the method of natural acceptance in relation to fulfilment of human values. (Evaluate)

Module 2:

CLO-5: Differentiate between the needs of the self and the body with relevant examples. (Understand)

CLO-6: Demonstrate an integrated understanding of the self-body relationship through reflective practices. (Apply)

CLO-7: Analyse how imagination, desires, and beliefs influence inner harmony. (Analyse)

CLO-8: Design a self-regulation program that enhances physical health and mental well-being. (Create)

Module 3:

CLO-9: Explain the importance of trust and respect in sustaining healthy interpersonal relationships. (Understand)

CLO-10: Apply value-based decision-making in resolving everyday social and familial situations. (Apply)

CLO-11: Evaluate societal structures and practices through the lens of justice and human values. (Evaluate)

CLO-12: Develop a blueprint for the universal human order grounded in ethical principles. (Create)

Module 4:

CLO-13: Describe the interrelatedness of the four orders of nature and their role in ecological balance. (Understand)

CLO-14: Exemplify sustainable living practices inspired by natural harmony and co-existence. (Apply)

CLO-15: Critique the human-nature relationship using the holistic framework of harmony in existence. (Evaluate)

Module 5:

CLO-16: Discuss the natural acceptance of human values as a foundation for ethical behaviour. (Understand)

CLO-17: Apply holistic principles to resolve dilemmas in professional and social contexts. (Apply)

CLO-18: Evaluate current professional practices against humanistic models of conduct and sustainability. (Evaluate)

CLO-19: Formulate a strategy for integrating value-based ethics in education, technology, and management. (Create)

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	1	2	2	1	3	2	3	2	2	1	3
CLO2	1	1	2	2	1	3	2	3	2	2	1	3
CLO3	1	1	2	2	1	3	2	3	2	2	1	3
CLO4	1	1	2	2	1	3	2	3	2	2	1	3
CLO5	1	1	2	2	1	3	2	3	2	2	1	3
CLO6	1	1	2	2	1	3	2	3	2	2	1	3
CLO7	1	1	2	2	1	3	2	3	2	2	1	3
CLO8	1	1	2	2	1	3	2	3	2	2	1	3
CLO9	1	1	2	2	1	3	2	3	2	2	1	3
CLO10	1	1	2	2	1	3	2	3	2	2	1	3
CLO11	1	1	2	2	1	3	2	3	2	2	1	3
CLO12	1	1	2	2	1	3	2	3	2	2	1	3
CLO13	1	1	2	2	1	3	2	3	2	2	1	3
CLO14	1	1	2	2	1	3	2	3	2	2	1	3
CLO15	1	1	2	2	1	3	2	3	2	2	1	3
CLO16	1	1	2	2	1	3	2	3	2	2	1	3
CLO17	1	1	2	2	1	3	2	3	2	2	1	3
CLO18	1	1	2	2	1	3	2	3	2	2	1	3
CLO19	1	1	2	2	1	3	2	3	2	2	1	3

Course Methodology

1. The methodology of this course is explorational and thus universally adaptable. It involves a systematic and rational study of the human being vis-à-vis the rest of existence.
2. The course is in the form of 28 lectures (discussions) and 14 practice sessions.
3. It is free from any dogma or value prescriptions.
4. It is a process of self-investigation and self-exploration, and not of giving sermons. Whatever is found as truth or reality is stated as a proposal and the students are facilitated to verify it in their own right, based on their Natural Acceptance and subsequent Experiential Validation –the whole existence is the lab and every activity is a source of reflection.
5. This process of self-exploration takes the form of a dialogue between the teacher and the students to begin with, and then to continue within the student in every activity, leading to continuous self-evolution.

Course Topics

The course has 28 lectures and 14 tutorials in 5 modules. The lectures and tutorials are of 1 hour duration. Tutorial sessions are to be used to explore and practice what has been proposed during the lecture sessions.

Module 1 – Introduction to Value Education (6 lectures and 3 tutorials for practice session)

- Lecture 1: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education)
- Lecture 2: Understanding Value Education
- Tutorial 1: Practice Session PS1 Sharing about Oneself
- Lecture 3: Self-exploration as the Process for Value Education
- Lecture 4: Continuous Happiness and Prosperity – the Basic Human Aspirations
- Tutorial 2: Practice Session PS2 Exploring Human Consciousness
- Lecture 5: Happiness and Prosperity – Current Scenario
- Lecture 6: Method to Fulfill the Basic Human Aspirations
- Tutorial 3: Practice Session PS3 Exploring Natural Acceptance

Module 2 – Harmony in the Human Being (6 lectures and 3 tutorials for practice session)

- Lecture 7: Understanding Human being as the Co-existence of the Self and the Body
- Lecture 8: Distinguishing between the Needs of the Self and the Body
- Tutorial 4: Practice Session PS4 Exploring the difference of Needs of Self and Body
- Lecture 9: The Body as an Instrument of the Self
- Lecture 10: Understanding Harmony in the Self
- Tutorial 5: Practice Session PS5 Exploring Sources of Imagination in the Self
- Lecture 11: Harmony of the Self with the Body
- Lecture 12: Programme to ensure self-regulation and Health
- Tutorial 6: Practice Session PS6 Exploring Harmony of Self with the Body

Module 3 – Harmony in the Family and Society (6 lectures and 3 tutorials for practice session)

- Lecture 13: Harmony in the Family – the Basic Unit of Human Interaction
- Lecture 14: 'Trust' – the Foundational Value in Relationship
- Tutorial 7: Practice Session PS7 Exploring the Feeling of Trust
- Lecture 15: 'Respect' – as the Right Evaluation
- Tutorial 8: Practice Session PS8 Exploring the Feeling of Respect
- Lecture 16: Other Feelings, Justice in Human-to-Human Relationship
- Lecture 17: Understanding Harmony in the Society

- Lecture 18: Vision for the Universal Human Order
- Tutorial 9: Practice Session PS9 Exploring Systems to fulfil Human Goal

Module 4 – Harmony in the Nature/Existence (4 lectures and 2 tutorials for practice session)

- Lecture 19: Understanding Harmony in the Nature
- Lecture 20: Interconnectedness, self-regulation and Mutual Fulfilment among the Four
- Orders of Nature
- Tutorial 10: Practice Session PS10 Exploring the Four Orders of Nature
- Lecture 21: Realizing Existence as Co-existence at All Levels
- Lecture 22: The Holistic Perception of Harmony in Existence
- Tutorial 11: Practice Session PS11 Exploring Co-existence in Existence

Module 5 – Implications of the Holistic Understanding – a Look at Professional Ethics (6 lectures and 3 tutorials for practice session)

- Lecture 23: Natural Acceptance of Human Values
- Lecture 24: Definitiveness of (Ethical) Human Conduct
- Tutorial 12: Practice Session PS12 Exploring Ethical Human Conduct
- Lecture 25: A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order
- Lecture 26: Competence in Professional Ethics
- Tutorial 13: Practice Session PS13 Exploring Humanistic Models in Education
- Lecture 27: Holistic Technologies, Production Systems and Management Models-Typical Case Studies
- Lecture 28: Strategies for Transition towards Value-based Life and Profession
- Tutorial 14: Practice Session PS14 Exploring Steps of Transition towards Universal Human Order

Course Topics

1. A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034- 47-1

Course title: Theory of Computation

Code: IE254

Type: PCC/DSC

Credits: 3

Semester: 4

Course	Theory of Computation	**Credits**	3
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Review various models of computation, their capabilities and limitations (Understand) CLO-2: Outline the suitability of different models of computation in various application scenarios. (Analyze) CLO-3: Categorize decidable and undecidable problems (Create) CLO-4: Develop programs simulating different models of computation (Create)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	1		2								
CLO2	1	3		2								
CLO3	1		1	3								
CLO4						1				3	2	

Syllabus

UNIT 1: Finite State Automata and Regular Languages (RL)

Preliminaries on Alphabets and Languages, Finite State Automata and Regular Languages (RL): Definition and examples; Regular expressions (RE), Non-deterministic finite automata (NFA), λ-NFA and Deterministic Finite automata (DFA); Equivalence of RE, NFA and DFA; Conversions from RE to (λ -) NFA to DFA to RE; Minimal DFA; Moore machine, Melay machine; Closure properties of RL; Pumping lemma for RL.

UNIT 2: Push Down Automata and Context-Free Languages (CFL)

CFL: Definition and examples; Grammar formalism for regular languages, Context free grammar (CFG), Derivation trees, Ambiguity, Normal forms; Push down automata (PDA) (deterministic and non-deterministic); Equivalence of CFG and PDA; CYK Algorithm; Closure properties of CFL; Pumping lemma for CFL.

UNIT 3: Turing Machine (TM)

Definition of TM; Examples; Variants of TM: Multi-tape and other variants of TM; Post Machine, Two-Stack PDA; Nondeterministic TM; Equivalence; Church-Turing Thesis; Universal Turing Machine

UNIT 4: Decidability and Undecidability

Definition of decidability; decidable problems concerning RL, CFL; Recursive and recursively enumerable languages; Undecidability; The Halting problem; Cantor's diagonalization argument; Examples of undecidable problems: Post's correspondence problem; Chomsky Hierarchy

Reference Books

1. J. Hopcroft, R. Motwani and J. Ullman, Introduction to Automata Theory, Languages and Computation. 3rd Edition, Pearson, 2014.
2. D.I.A. Cohen, Introduction to Computer Theory. 2nd Edition, Wiley India, 1991.
3. J.C. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw Hill, 2003
4. H. Lewis and C. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Prentice Hall, 1998.
5. M Sipser, Introduction to Theory of Computation, Thomson Learning, 2014.
6. Peter Linz, An introduction to formal languages and automata, Jones and Barlett Publishers, 2016.

Course title: Environmental Sciences

Code: IE263

Type: MC

Credits: 1

Semester: 4

Course	Environmental Sciences	**Credits**	1
Course Type	MC		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Analyse real life problems associated with the ecosystem. CLO-2: Explain various types of pollution sources and its effect. CLO-3: Apply various types of pollution control mechanism to curb environmental pollution. CLO-4: CO-4: Explain various sources of renewable energy and process of harnessing. CLO-5: CO-5: Explain solid waste management, ISO 14000 & Environmental management.			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	1	1	1	1	3	3	1	1	1	1	2
CLO2	1	1	1	1	1	3	3	1	1	1	1	2
CLO3	1	1	1	1	1	3	3	1	1	1	1	2
CLO4	1	1	1	1	1	3	3	1	1	1	1	2
CLO5	1	1	1	1	1	3	3	1	1	1	1	2
CLO6	1	1	1	1	1	3	3	1	1	1	1	2

Syllabus

UNIT 1: Ecosystem

Introduction, Aquatic and terrestrial ecosystem, Structure of ecosystem, Food chain and food web, Carbon cycle, Nitrogen Cycle, Sulphur Cycle, Phosphorous Cycle, Global warming, Innovative Activities

UNIT 2: Air and Noise Pollution

Introduction, Definition of pollution and pollutant, Gaseous Pollution Control, Noise Pollution, Innovative Activities

UNIT 3: Renewable sources of Energy

Introduction, Solar Energy, Biomass, Wind energy, New Energy Sources, Innovative Activities.

UNIT 4: Solid Waste Management, ISO 14000 & Environmental Management

Introduction Solid Waste Generation, Metallistic Wastes and Non-Metallic Wastes, Collection and Disposal of MSW (Municipal Solid Waste), Air Quality Act 2004, Structure and role of Central and State Pollution Control Board. Carbon Foot print and Carbon Credit, Environmental Management in Fabrication Industry, ISO14000, Innovative Activities

Reference Books

5. Subrat Roy, Environmental Science, ISBN 978-93-01505-65-3, Khanna Book Publishing Co.(P) Ltd, 2021.
6. Suresh K. Dhameja, Environmental Studies, S.K. Kataria & Sons, 2012.
7. Surinder Deswal, Energy, Environment Ecology and Society, Dhanpat Rai & Sons, 2014.
8. P.K. Pandey, Environment and Ecology, Sun India Publication, 2009.
9. P.S. Ramakrishnan, Energy and Sustainable Development, National Book Trust, 2014.
10. M.K. Goyal, Our Environment (Hindi text book), Agrawal Publication, Agra, 2013.
11. C.N.R. Rao, Understanding Chemistry, University press (India) Pvt. Ltd., 2011.
12. G.Chopra, Science Biology, Pradeep Publications, New Delhi – 2016.
13. http://www.indiaenvironmentportal.org.in/

Course title: Constitution of India

Code: IE264

Type: MC

Credits: 1

Semester: 4

Course	Constitution of India	**Credits**	1
Course Type	MC		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Describe historical background of the constitution making and its importance for building a democratic India. CLO-2: Explain the functioning of three wings of the government i.e., executive, legislative and judiciary. CLO-3: Explain the value of the fundamental rights and duties for becoming good citizen of India. CLO-4: Analyse the decentralisation of power between central, state and local self-government. CLO-5: Apply the knowledge in strengthening of the constitutional institutions like CAG, Election Commission and UPSC for sustaining democracy.			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1	1	1	1	1	3	3	2	1	1	1	2
CLO2	1	1	1	1	1	3	3	2	1	1	1	2
CLO3	1	1	1	1	1	3	3	2	1	1	1	2
CLO4	1	1	1	1	1	3	3	2	1	1	1	2
CLO5	1	1	1	1	1	3	3	2	1	1	1	2
CLO6	1	1	1	1	1	3	3	2	1	1	1	2

Syllabus

UNIT 1: Introduction to Indian Constitution

Constitution' meaning of the term, Indian Constitution - Sources and constitutional history, Features - Citizenship, Preamble, Fundamental Rights and Duties, Directive Principles of State Policy

UNIT 2: Union Government and its Administration Structure of the Indian Union

Federalism, Centre- State relationship, President: Role, power and position, PM and Council of ministers, Cabinet and Central Secretariat, Lok Sabha, Rajya Sabha, The Supreme Court and High Court: Powers and Functions;

UNIT 3:

Introduction, Solar Energy, Biomass, Wind energy, New Energy Sources, Innovative Activities. State Government and its Administration Governor - Role and Position - CM and Council of ministers, State Secretariat: Organisation, Structure and Functions

UNIT 4:

Local Administration - District's Administration Head - Role and Importance, Municipalities - Mayor and role of Elected Representative - CEO of Municipal Corporation Pachayati Raj: Functions PRI: Zilla Panchayat, Elected officials and their roles, CEO Zila Panchayat: Block level Organizational Hierarchy - (Different departments), Village level - Role of Elected and Appointed officials - Importance of grass-root democracy

UNIT 5:

Election Commission: Role of Chief Election Commissioner and Election Commission; State Election Commission: Functions of Commissions for the welfare of SC/ST/OBC and women

Reference Books

1. Durga Das Basu, Introduction to the Constitution of India, Prentice – Hall of India Pvt.Ltd. New Delhi.
2. SubashKashyap, Indian Constitution, National Book Trust J.A. Siwach, Dynamics of Indian Government & Politics D.C. Gupta, Indian Government and Politics.

3. H.M.Sreevai, Constitutional Law of India, 4th edition in 3 volumes (Universal Law Publication)
4. J.C. Johari, Indian Government and Politics Hans J. Raj Indian Government and Politics.
5. M.V. Pylee, Indian Constitution Durga Das Basu, Human Rights in Constitutional Law, Prentice – Hall of India Pvt.Ltd.. New Delhi
6. Noorani, A.G., (South Asia Human Rights Documentation Centre), Challenges to Civil Right), Challenges to Civil Rights Guarantees in India, Oxford University Press 2012

Course title: Accounting and Financial Management

Code: IE311

Type: HSMC/HSC

Credits: 3

Semester: 5

Course	Accounting and Financial Management	**Credits**	3
Course Type	**HSMC/HSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			

Course title: Operating Systems and Lab

Code: IE301 and IE308

Type: PCC/DSC

Credits: 3 + 2

Semester: 5

Course	Operating Systems and Lab	**Credits**	3 + 2
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Discuss the ways system calls work. (Understand) CLO-2: Develop basic process management tasks such as scheduling, deadlock avoidance algorithms. (Create) CLO-3: Develop paging algorithm. (Create) CLO-4: Construct simple device drivers. (Create) CLO-5: Describe different file systems in existence and learn the pros and cons of the various systems. (Understand) CLO-6: Examine real world OS scheduling algorithms such as those used in Linux and Windows. (Analyze)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1		3		2							
CLO2	3	2	1									
CLO3		1	2			3						
CLO4				2	1	3						
CLO5	3	2	1									
CLO6	1	2	3									

Syllabus

UNIT 1: Introduction and Operating System Structures

Operating Systems Functionality, Computer Organization and Architecture, OS Operations, Kernel Data Structures, OS Services, User interfaces to OS, Programmer interfaces to OS, OS Structure, System Boot.

UNIT 2: Process and Thread Management

Process Concept, Process operations, Process Scheduling, Extended Process State Diagram, Process Context Switch in detail; Inter process Communication: Pipes, Named Pipes, Shared Memory; Process Synchronization: Signals, Mutexes, Semaphores, Monitors; Thread Management: thread creation, thread scheduling, thread synchronization; Deadlocks: Resource Allocation Graphs, deadlock detection, prevention and avoidance, recovery from deadlock.

UNIT 3: Memory Management

Memory allocation techniques: paging and segmentation, Swapping, structure of the page table; Virtual memory: demand paging, copy-on-write, Page replacement, allocation of frames, kernel memory allocation, thrashing, memory-mapped files, Translation-Lookaside Buffer (TLB).

UNIT 4: File System Management

Disk management: formatting, boot block, swap-space management, RAID structure; Disk scheduling algorithms: elevator, C-SCAN; File concept, Access methods, Directory structure, file sharing, protection, file system structure; file system implementation: file system metadata storage structures such as inode, allocation methods, free space management, efficiency and performance including disk cache and recovery from failures.

UNIT 5: I/O Management

I/O devices: polling, interrupt-driven, DMA; Application I/O interface: character and block devices, network devices; clocks and timers, nonblocking and asynchronous I/O, vectored I/O; Kernel I/O interface: I/O scheduling, Buffering, Caching.

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne. Operating System Concepts, 9th edition, Wiley.
2. Charles Crowley. Operating Systems: A Design-Oriented Approach, Prentice-Hall India.
3. W. Richard Stevens, . Advanced Programming in Unix Environment, Pearson Education.
4. W. Richard Stevens. Unix Network Programming, vol. 2, Pearson Education.
5. William Stallings. Operating Systems: Internals and Design Principles, Pearson Education.
6. Maurice J. Bach. The Design of the Unix Operating System, Prentice-Hall India.
7. Robert Love. Linux Kernel Development, Pearson Education.
8. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice, 2nd edition, Recursive Books.

Course title: Computer Networks and Lab

Code: IE310 and IE312

Type: PCC/DSC

Credits: 4 + 2

Semester: 5

Course	Computer Networks and Lab	**Credits**	4 + 2
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Distinguish between multiplexing techniques (Understand) CLO-2: Evaluate the different types of switched networks (Analyze) CLO-3: Explain the functionalities media access for data-link and network protocols. (Understand) CLO-4: Apply IP addressing and routing algorithms to design networks by subnetting/supernetting (Apply) CLO-5: Describe the essential principles such as reliable data transfer, flow control, congestion control of a transport layer protocol (Understand) CLO-6: Predict the topology given the routing protocol messages (Apply) CLO-7: Analyze and capture network traffic using simulation tools. (Analyze)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	2		1								
CLO2		2		3						1		
CLO3	3	2								1		
CLO4	1		2			3						
CLO5	3	2									1	
CLO6				1		2					3	

Syllabus

UNIT 1: Physical Layer

Modulation Techniques: Amplitude, Frequency and Phase, ADSL, Multiplexing Techniques: Frequency division multiplexing, Time division multiplexing, wave length division multiplexing, Differential PCM, Switching Techniques: Circuit, message and packet switching.

UNIT 2: Data Link Layer

PPP, PPPoE, MAC Layer: Ethernet (incl. manchester encoding), Switched Ethernet, VLANs, Spanning Tree Protocol.

UNIT 3: Network Layer – Data Plane

Internet Protocol Addressing: CIDR, Internet Protocol Datagram (including fragmentation and reassembly, routing options), IP Forwarding Algorithm, ARP, ICMP (including ICMP Redirect, ICMP Path MTU discovery, ICMP Destination Unreachable options).

UNIT 4: Transport Layer

UDP, TCP sliding window protocol, TCP connection establishment, TCP reliability including cumulative and delayed acknowledgements, Nagle algorithm, Karn's algorithm for RTT and RTO estimation, TCP AIMD Congestion Control Algorithm, TCP half-close connections including TCP keepalive timer and probe timer, TCP Fast Retransmit and Fast Recovery.

UNIT 5: Network Layer – Control Plane

Distance Vector Algorithm and Routing Information Protocols V1 and V2, Link State Algorithm and Open Shortest Path First Protocol (OSPF).

UNIT 6: Application Layer

Domain Naming System (DNS) and Dynamic Host Configuration Protocol (DHCP), Network Management using SNMP.

Reference Books

1. James F. Kurose and Keith W. Ross. Computer Networking: A top-down approach, 6th edition, Pearson Education.
2. Douglas Comer. Computer Networks and Internets Sixth Edition, 2014. ISBN 0133587932/9780133587937, Pearson Education.
3. Douglas Comer. Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 6th edition, 2013. ISBN-10: 0-13-608530-X ISBN-13: 9780136085300, Pearson Education.
4. Kevin R. Fall and W.Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols, 2/E, 2012, ISBN-10: 0321336313 ISBN-13: 9780321336316, Pearson Education.
5. Radia Perlman. Interconnections: Bridges, Routers, Switches, and Internetworking Protocols, 2/E, 2000, ISBN-10: 0201634481 ISBN-13: 9780201634488.Pearson Education.

Course title: Algorithms

Code: IE304

Type: PCC/DSC

Credits: 4

Semester: 5

Course	Algorithms	**Credits**	4
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Assess the inherent structure/hardness of a problem (Evaluate) CLO-2: Select an appropriate strategy to solve a problem (Understand) CLO-3: Design an algorithm that suits the time complexity requirements of the problem. (Create) CLO-4: Estimate the time and space complexities of an algorithm along with the necessary mathematical proofs when necessary. (Evaluate) CLO-5: Devise algorithms by choosing appropriate data structures (Create)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	2	3				1						
CLO2		1	2		3							
CLO3	2		3	1								
CLO4	1		2	3								
CLO5	2		1			3						
CLO6	2	3				1						

Syllabus

UNIT 1: Analysis of Algorithms

Asymptotic Notation; Best, worst and average case analysis of algorithms; Solving recurrence relations using substitution method, generating functions, Master's theorem etc. Warm-up to complexity analysis: Heap data structure, priority queue application, Best, worst and average case analysis of a few sorting algorithms like heap sort, insertion, bubble, selection, counting and radix sort algorithms. Strategies for problem solving.

UNIT 2: Divide and Conquer Strategy

Time complexity analysis for Merge Sort and Quick Sort Algorithms

UNIT 3: Greedy Strategy

Theoretical foundation of greedy strategy: Matroids Algorithms for solving problems like Knapsack Problem (Fractional), Minimum Spanning Tree problem; Shortest Paths, Job Scheduling, Huffman's code etc along with proofs of corrections and complexity analysis.

UNIT 4: Dynamic Programming Strategy

Identify situations in which greedy and divide and conquer strategies may not work. Understanding of optimality principle. Technique of memorization. Applications to problems like Coin change, 0/1 and 0/n- Knapsack, Shortest Paths, Optimal Binary Search Tree (OBST), Chained Matrix Multiplication, Traveling Salesperson Problem (TSP) etc.

UNIT 5: Backtracking and Branch & Bound Strategies

State space tree construction, traversal techniques and solving problems like 0/1 and 0/n knapsack, TSP, Applications of Depth First Search: Topological sorting, Finding strongly connected components and game problems.

UNIT 6: Theory of NP-Completeness

Complexity classes of P, NP, NP-Hard, NP-Complete, Polynomial reductions, Cook's theorem. Discussion of problems: Satisfiability (SAT), CNF-SAT, Min-Vertex Cover, Max-Clique, Graph Colouring, NP-Completeness proofs.

Reference Books

1. Introduction to Algorithms-T.Cormen, C.E.Leiserson, R.L.Rivest, PHI, 3rdEdition 2009.
2. Algorithms- R.Johnsonbaugh and M.Schaefer, Pearson, 2004.
3. Fundamentals of Algorithmics - G.Brassard and P.Bratley, PH, 1996
4. The Algorithm Design Manual- Steven S. Skiena, Springer, 2009

Course title: Compiler Design

Code: IE360

Type: PCC/DSC

Credits: 3

Semester: 6

Course	Compiler Design	**Credits**	3
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			

CLO-1: Discuss different phases of the compilation process (Understand)

CLO-2: Identify appropriate compiler for the given problem (Analyze)

CLO-3: Design a prototype for a small language (Create)

CLO-4: Write programs from the knowledge gained about the compilation (Apply)

CLO-5: Write the Implementation of the Lexical analyzer, parser and code generator using tools such as Lex, Bison etc (Apply)

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	2		1								
CLO2			3	2	1							
CLO3			3		2	1						
CLO4			2		3		1					
CLO5			2		3		1					

Syllabus

UNIT 1: Compiler Structure

Analysis- synthesis model of compilation, various phases of a compiler, tool-based approach to compiler construction.

UNIT 2: Lexical Analysis

Interface with input program, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, implementation, regular definition, transition diagrams.

UNIT 3: Syntax Analysis

CFGs, ambiguity, error detection and recovery, top-down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom-up parsing, operator precedence parsing, LR parsers (SLR, LALR, LR).

UNIT 4: Syntax Directed Translation

Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top-down evaluation of attributes, L- and S-attributed definitions, Type checking

UNIT 5: Run time Environments

Source language issues, storage organization, activation tree, activation record, stack allocation of activation records, parameter passing mechanisms, symbol tables, dynamic storage allocation techniques.

UNIT 6: Intermediate Code Generation

Intermediate representations, translation of declarations, assignments, control flow, boolean expressions and procedure calls. Implementation issues.

UNIT 7: Code Generation and Instruction Selection

Issues, basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from DAGs, peep hole optimization.

UNIT 8: Code Optimization

Principal Sources of Optimization, Optimization of basic blocks, Introduction to Data flow Analysis (Reaching Definitions and Live Variable Analysis).

Reference Books

1. AW Appel, M Ginsburg: Modern Compiler Implementation in C, Cambridge University Press
2. K Cooper, L Torczon: Engineering a Compiler, Morgan Kaufmann
3. J.P. Tremblay, P.G. Sorenson: Theory and Practice of Compiler Writing, McGraw Hill

Suggested Assignment

Programming assignments based on lexical analysis, construction of predictive/operator precedence/SLR parsing table and parser, symbol table, dynamic storage allocation strategies, syntax directed translation, data flow analysis.

Course title: Software Engineering and Lab

Code: IE351 and IE355

Type: PCC/DSC

Credits: 3 + 2

Semester: 6

Course	Software Engineering	**Credits**	3
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Explain the models of software development process (Understand) CLO-2: Evaluate the appropriateness of different models of software development for their application in various domains (Evaluate). CLO-3: Apply the requirements engineering to software systems. (Apply) CLO-4: Describe Software Architectures (understand). CLO-5: Assess the applicability of software architectures for various combinations of non-functional requirements (Evaluate level). CLO-6: Apply object oriented and structured and structured paradigms to design software systems (Apply). CLO-7: Apply testing strategy to test software applications (Apply).			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	3	2				1						
CLO2	2			3	1							
CLO3						3				1	2	
CLO4	3	2		1								
CLO5				3	1		2					
CLO6	1		3		2							
CLO7	1		3								2	

Syllabus

UNIT 1: Introduction to Software Engineering

Need of software engineering, systems engineering, challenges in software engineering, Software process models, quality characteristics of software systems, Ethics in Software Engineering.

UNIT 2: Requirements Engineering

Requirements engineering process, requirements specification, structured and object-oriented analysis.

UNIT 3: Software Design

Architectural design, detailed design, Structured and object-oriented design, user interface design.

UNIT 4: Software Testing

Verification, Validation, testing techniques, Testing Process.

UNIT 5: Tools and Evolution

CASE Tools, Reverse engineering, Reengineering and Configuration management.

Course	Software Engineering Lab	**Credits**	2
Course Type			
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Create user stories (Create). CLO-2: Develop test plans for test first development (Create). CLO-3: Design & develop the stories (Create). CLO-4: Create the documentation (Create). CLO-5: Develop Software requirements specification document (Create). CLO-6: Apply object oriented and structured paradigm (Apply). CLO-7: Generate test reports (Create)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1			3		2						1	
CLO2			3		2						1	
CLO3			2		3						1	
CLO4			3		2						1	
CLO5		2			3						1	
CLO6					2	3					1	
CLO7				2	3						1	

Syllabus

For a given case study/problem statement, the following deliverables are to be realized

- Define stories
- Identify tasks and develop test plan for stories/task (with the help of specifications)
- Design and develop increments
- Test the increments and release the increment
- Apply object oriented and structured modelling
- Implement the case study for plan driven approach by writing use case specification, designing the system and implementing the same.

Reference Books

1. Ian Sommerville (2016), “Software Engineering”, 10th Edition, Pearson Education Limited, Global Edition.
2. Roger S Pressman, Bruce R Maxim (2015), “Software Engineering, A Practitioner’s Approach”, 8th Edition, Tata McGraw Hill, Indian Edition
3. Grady Booch, James Rumbaugh, Ivor Jacobson (2005), “The Unified Modelling Language User Guide”, 2nd Edition, Addison Wesley Professional.US

Course title: Internet Technologies and Lab

Code: IE361 and IE363

Type: PCC/DSC

Credits: 3 + 2

Semester: 6

Course	Internet Technologies and Lab	**Credits**	3 + 2
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Apply protocols related to network application layer for internet applications such as decentralized communications and remote data sharing. (Apply) CLO-2: Analyze a web page and identify its elements and attributes, create web pages using scripting languages, cascading styles sheets, and build dynamic webpages using JavaScript. (Create) CLO-3: Develop interactive web applications using server side and database technologies (Create) CLO-4: Create schemas and documents using markup languages, design and develop Lightweight data - interchange format for exchange of data between client and server applications. (Create) CLO-5: Analyze and apply search engine services for web applications (Analyze)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1	1		2			3						
CLO2		3	1		2							
CLO3			2		1	3						
CLO4	2		1		3							
CLO5	1				3	2						

Syllabus

UNIT 1: Application Layer Protocols

HTTP, Proxy Servers, SMTP, POP, IMAP, SSH, FTP, Peer-to-Peer protocols such as BitTorrent, Distributed Hash Tables.

UNIT 2: Client-Side Technologies

HTML, CSS, PHP, JavaScript, XML, Document Object Model (DOM), Dynamic Content, Cookies.

UNIT 3: Connection to Server and Server-Side Technologies

Databases (MySQL/MongoDB) and JDBC, Servlets, JSP, NodeJS.

UNIT 4: Advanced Client-Side Technologies

Asynchronous JavaScript and XML (AJAX), JQuery, JavaScript Object Notation (JSON), Google Web Toolkit, [Ruby on Rails1], GoJS, Firebug.

UNIT 5: Advanced Web Applications

Search Engines and their algorithms, Google Maps and building your own Google Maps, Keyhole Markup Language (KML) on Google Earth.

Reference Books

1. Core Servlets and Java Server Pages (JSP), by Marty Hall, Prentice Hall, 2nd edition (2003).
2. Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX by Elliotte Rusty Harold, Addison-Wesley Pub Co; 1st edition, 2002. (Available online at http://cafeconleche.org/books/xmljava/
3. Glee Harrah Cady, Pat McGregor: Mastering the Internet, BPB, Sybex 1996.
4. Alan Simpson: HTML Publishing Bible, IDG Books, Comdex Computer Publishing, A Division of Pusthak Mahal, 1996.
5. Bryan Pfaffenberger: Publish on the Web, AP Professional, 1996.
6. Clayton Walnum: Java by Example, Que 1996.
7. Marty Hall: The Core Web Programming, Prentice-Hall, 1998.
8. J. Niederst: Web Design in a Nutshell, O'Reilly Associates

Course title: Humanities – I (Applied Linguistics, etc.)

Code:

Type: HSMC/HSC

Credits: 3

Semester: 6

Course	Humanities – I (Applied Linguistics, etc.)	**Credits**	3
Course Type	**HSMC/HSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			

Course title: Essentials of AI

Code: IE362

Type: PCC/DSC

Credits: 4

Semester: 6

Course	Essentials of AI	**Credits**	4
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation and learning. (Apply) CLO-2: Demonstrate awareness and a fundamental understanding of various applications of AI in intelligent agents, expert systems, artificial neural networks and other machine learning models. (Apply) CLO-3: Demonstrate proficiency in applying scientific method to models of machine learning. (Apply) CLO-4: Apply selected basic AI techniques; judge applicability of more advanced techniques. (Apply) CLO-5: Design and develop a system that act intelligently and learns from experience (Create) CLO-6: Demonstrate an ability to share in discussion of AI, its current scope and limitations and societal implications. (Apply).			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1			2	3	1							2
CLO2			3		1	2						3
CLO3	2		3			1				2		3
CLO4			3	2	1							3
CLO5	3	1				2				3	1	
CLO6	2	1				3				2	1	

Syllabus

UNIT 1: Introduction to AI

Introduction to artificial intelligence, History of AI, Agents & Environment, The Structure of Agents, Agent programs, simple reflex agents, Model based agents, Goal based agents, utility based agents, Learning agents. Proposing and evaluating AI applications, Case study: Google Duplex.

1) Russell & Norvig, "Chapter 1: Introduction" in Artificial Intelligence: A Modern Approach, Third Edition
2) Chui, et. al, "Notes From the AI Frontier," McKinsey Global Institute, April 2018
3) Amadeo, R., June 27, 2018, "Talking to Google Duplex: Google's human-like phone AI feels revolutionaryfile://localhost/"

UNIT 2: Search & Planning

Problem spaces and search, Knowledge and rationality, Uninformed search Strategies (BFS, Uniform-cost search, DFS, Depth-limited search, iterative deepening, Bidirectional search), Heuristic search strategies (Hill climbing, Simulated Annealing, , A*, memory bounded heuristic search), Heuristic Functions, Local search algorithms, Searching with partial observations, searching with non-deterministic actions, Online search, Search and optimization (gradient descent) , Adversarial search (minmax, Alhha-Beta pruning, stochastic games, partially observable games sate-of-the art game progrma), Planning and scheduling, Case studies: Playing chess, manufacturing scheduling

1) "Chapters 3, 4 : Solving Problems by Searching," "Chapter 5: Adversarial search", "Chapter 10.2-10.5: Planning", "Chapter 11: Planning and Acting in the Real World" in Russell & Norvig, Artificial Intelligence: A Modern Approach, Third Edition

UNIT 3: Knowledge Representation & Reasoning

Logic and inference, Propositional theorem proving, Propositional Model checking, Agents based on Propositional Logic, First Order logic, Knowledge Engineering in first-order logic, unification & lifting, Forward chaining & Backward chaining Resolution, logic programming, Ontologies, Bayesian reasoning, Temporal reasoning, Case study: Medical diagnosis.

1) Readings in Russell & Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Propositional logic: Chapter 7, First-order logic: Chapter 8, Sections 8.1.2 - 8.2, 8.4, 9.1-9.5, Knowledge representation: Sections 12.1-12.5, 12.7, Quantifying uncertainty: Ch. 13, Probabilistic reasoning: Ch. 14.1-14.2 ○ (Optional) Probabilistic reasoning over time: Ch. 15

UNIT 4: Machine Learning & Supervised Methods

What is machine learning? Supervised vs. unsupervised learning, The theory of learning, Regression --linear, logistic, ridge. Classification – decision trees, SVM, random forests, Model performance evaluation, non-parametric models, Ensemble learning.

UNIT 5: Machine Learning: Unsupervised Methods

Dimensionality reduction: PCA, Clustering-- k-means, hierarchical clustering. Semi-supervised methods (GANs), Reinforcement learning, Choosing among machine learning techniques.

1) Chapter 18 in Russell & Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition
2) Chapter 5.8: "Unsupervised Machine Learning," in Goodfellow, I.,Bengio, Y. and Courville A., Deep Learning, 2016.
3) Russell & Norvig, Chapter 21 "Reinforcement Learning" in Artificial Intelligence: A Modern Approach, 3rd Edition

UNIT 6: Natural Language Understanding

Intro to natural language understanding, Language Models, Information retrieval, Information Extraction, Phase Structure grammars, Syntactic Analysis, Augmented grammars and semantic interpretation, Machine translation, Sentiment analysis, Hidden Markov Models, Chatbots, Natural language generation, Speech synthesis, Case study: Google Duplex (revisited)

1) Russell & Norvig, "Chapter 22: Natural Language Processing" in Artificial Intelligence: A Modern Approach, Third Edition.
2) Collobert et al. "Natural Language Processing (Almost) from Scratch," Journal of Machine Learning Research, 2011 available at https://arxiv.org/pdf/1103.0398.pdf
3) (Optional) G Golderg, Y. Neural Network Methods for Natural Language Processing Synthesis Lectures on Human Language Technologies, April 2017, freely available monograph at https://doi.org/10.2200/S00762ED1V01Y201703HLT037
4) (Optional) Feldman, R, "Sentiment Analysis Tutorial, IJCAI-13, 2013, http://ijcai13.org/files/tutorial_slides/tf4.pdf
5) Russell & Norvig, "Chapter 15.3: Hidden Markov Models" and "Chapter 22: Natural Language for Communication" in Artificial Intelligence: A Modern Approach, Third Edition.

UNIT 7: AI in the Enterprise, Ethical & Legal Considerations in AI

Privacy, Bias, AI and the future of work, Appropriate uses of AI, Infrastructure for AI: Parallel & distributed computing for scalability, Resolving technical trade-offs, Case Study: Uber & Facebook

1) Beyer, D, "AI and Machine learning in industry," 2017, download from http://www.oreilly.com/data/free/ai-machine-learning-in- industry.csp?cmp=tw-data-free-article-lgen_tw_free_ebook_as
2) Jerome, J, "Why AI may be the next big privacy trend,"
3) https://iapp.org/news/a/why-artificial-intelligence-may-be-the-next-big-privacy- trend/, 2016
4) Burt, A . "How will the GDPR impact machine learning?", May 16, 2018, "https://www.oreilly.com/ideas/how-will-the-gdpr-impact-machine learning
5) Vanian, J "Unmasking A.I.'s Bias Problem," Fortune, June 25, 2018, http://fortune.com/longform/ai-bias-problem/ NSTC, "Preparing for the Future of AI," October 2016, Brynjolfsson, E and Mitchell, T. "What can machine learning do?

1) Workforce implications," Science 22 Dec 2017: Vol. 358, Issue 6370, pp. 1530- 1534 DOI: 10.1126/science.aap8062

6) Courtland, R. "Bias detectives: the researchers striving to make algorithms fair," Nature, June 2018, https://www.nature.com/magazine- assets/d41586-018-05469-3/d41586-018-05469-3.pdf
7) Zheng, H. Wang, Y, and Molino, P. "COTA: Improving Uber Customer Care with NLP & Machine Learning," January 2018, https://eng.uber.com/cota/
8) Hermann and Del Balso, 2017, "Meet Michelangelo: Uber's Machine Learning Platform," https://eng.uber.com/michelangelo/
9) National Science and Technology Council, "Preparing for the future of AI," October 2016, https://obamawhitehouse.archives.gov/sites/default/files/whitehouse_files/microsites/ostp/NSTC/preparing_for_the_future_of_ai.pdf

Reference Books

1. Artificial Intelligence: A Modern Approach, Stuart Russel & Peter Norvig, Third Edition.
2. Essentials of Artificial Intelligence, Matt Ginsbeg.

Course title: Pattern Recognition

Code: IE456

Type: PCC/DSC

Credits: 4

Semester: 8

Course	Pattern Recognition	**Credits**	4
Course Type	**PCC/DSC**		
Course Description			
Course Objectives			
Course Learning Outcomes			
CLO-1: Describe the approaches to statistical and syntactic pattern recognition (Understand) CLO-2: Describe the design characteristics of pattern recognitions systems such as curse of dimensionality (Understand) CLO-3: Apply pattern recognition techniques to solve real world problems (Apply) CLO-4: Design simple pattern classifiers, classifier combinations ad structural pattern recognizers (Create) CLO-5: Develop pattern recognition techniques and the scientific computing environment. (Create)			

Mapped to Program Level Outcomes												
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12
CLO1												
CLO2												
CLO3												
CLO4												
CLO5												
CLO6												
CLO7												

Syllabus

UNIT 1: Introduction to Patterns and Recognition Methods

Concepts of Pattern Recognition, applications, in real world; patterns and their representations as random vectors, strings and graph, time series. Feature and data scales. Feature and pattern similarity, dissimilarity measures. Feature selection, dimensionality reduction. PCA and SVD methods. Introduction to pattern classification paradigms such as statistical, syntactical and structural pattern recognition. Nearest Neighbor Classification as a case study.

UNIT 2: Statistical Pattern Recognition

Bayes decision theory, Maximum aposteriori classification, risk and errors; Supervised learning using parametric and nonparametric approaches: Maximum Likelihood estimation, Bayesian parameter estimation approach, Parzen Windows, k-NN estimation.

UNIT 3: Linear Discriminative Classifiers

Decision Boundaries, Separability; Perceptron's, Support Vector Machines, Fisher's Linear Discriminant function, Decision Trees, Random Forests, Projections and Embeddings

UNIT 4: Unsupervised learning and other Methods

Unsupervised classification, clustering: the clustering concept, clustering strategies, a hierarchical clustering, Partitional clustering, c-means algorithm, learning vector quantization, expectation maximization and mean shift.

UNIT 5: String and Sequence Recognition Methods

Formal languages; String languages for pattern recognition: selection of pattern primitives, Matching of Strings, Edit distances, pattern grammars, basic formulation of states and sequences, Markov Models, Hidden Markov Models, Viterbi Algorithm, Baum Welch Learning, Graphs fundamentals of graph theory, basic algorithms for graphs matching

Reference Books

1. Duda, R.O., Hart, P.E., and Stork, D.G. Pattern Classification. Wiley-Interscience. 2nd Edition. 2001. The textbook has a website: http://www.rii.ricoh.com/~stork/DHS.html.
2. Andrew R. Webb, and Keith D. Copsey, Statistical Pattern Recognition, Third Edition, Wiley Interscience, 2011
3. Bishop, C. M. Pattern Recognition and Machine Learning. Springer. 2007.
4. Theodoridis, S. and Koutroumbas, K. Pattern Recognition. Edition 4. Academic Press, 2008.
5. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press. 1995.
6. Hastie, T., Tibshirani, R. and Friedman, J. The Elements of Statistical Learning. Springer. 2001.
7. Koller, D. and Friedman, N. Probabilistic Graphical Models. MIT Press. 2009.