NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY

DEEMED TO BE UNIVERSITY UNDER DISTINCT CATEGORY

(AN AUTONOMOUS INSTITUTION UNDER MINISTRY OF ELECTRONICS & IT, GOVT. OF INDIA)

Curriculum and Syllabi for

B.Tech - Computer Science and Engineering (Artificial Intelligence and Machine Learning)

(2025-26 Scheme)

Main Campus at Ropar and Constituent Units at Aizawl, Agartala, Aurangabad, Calicut, Gorakhpur, Imphal, Itanagar, Kekri, Kohima, Patna and Srinagar

Vision

To be a nationally recognized center of excellence in Artificial Intelligence and Machine Learning, producing competent, ethical, and innovative engineers to lead the digital future.

Mission

1. To deliver quality education in Computer Science with a strong emphasis on AI and ML technologies.
2. To equip students with theoretical knowledge and practical skills required to develop intelligent and adaptive systems.
3. To promote interdisciplinary research, innovation, and entrepreneurship.
4. To instill ethical values, teamwork, and communication skills for holistic professional development.
5. To enable graduates to address societal challenges through responsible and inclusive AI applications.

Program Education Objectives

Graduates of the B.Tech (AI & ML) program will:
1. Apply their knowledge and skills to develop intelligent software and hardware systems in a variety of domains.
2. Pursue successful careers in industry, academia, or entrepreneurial ventures in the field of AI and emerging technologies.
3. Demonstrate professional ethics, leadership, and communication abilities in multidisciplinary environments.
4. Engage in lifelong learning and advanced studies to remain current with technological advancements.
5. Contribute responsibly to societal and environmental well-being through innovative AI-driven solutions.

Program Outcomes

Upon successful completion of the program, students will be able to:
1. Apply Engineering Knowledge: Apply mathematics, computer science, and AI principles to solve complex engineering problems.
2. Design Intelligent Solutions: Design and develop AI-based systems that meet specified requirements in diverse domains.
3. Modern Tool Usage: Utilize AI/ML tools, libraries, and platforms for development, analysis, and deployment.
4. Ethics and Communication: Demonstrate ethical responsibility, teamwork, and effective communication skills.
5. Lifelong Learning: Engage in continuous learning and adapt to rapidly evolving AI technologies and practices.

Program Specific Outcomes

Graduates of the B.Tech (AI & ML) program will be able to:
1. Build, train, and evaluate machine learning and deep learning models for solving real-world problems.
2. Design and implement AI applications in areas such as computer vision, natural language processing, robotics, and healthcare.
3. Integrate data engineering, cloud computing, and MLOps practices to deploy scalable AI systems.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Audit Courses (without grade or credit)	AU	0
Value Added Courses	VA	6
Professional Elective Courses	PE	20
Open Elective Courses	OE	12
Ability Enhancement Courses	AE	8
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	31
Program Core Courses	PC	88
Total Credits (Common track)		**165**

EVALUATION AND ASSESSMENT

1. Performance of a student in a semester shall be evaluated through continuous Class assessment, Tutorial/Lab assessment, Mid-Semester Examination (MSE) and End-Semester Examination (ESE). Both the MSE and ESE shall be the University examination and will be conducted as notified by the Controller of Examinations (CoE) of the University.

2. The continuous assessment shall be based on assignments, tutorials, paper presentation / quizzes/ viva-voce / flipped classes, lab work / projects / fieldwork and attendance, etc.

3. The MSE/ESE shall be comprising of written papers.

4. The overall assessment of the students will be done as per the following scheme:

S. No.	Assessment Type	Marks Weightage
1.	Mid Semester Examination	20
2.	End Semester Examination	40
3.	Continuous Assessment - Practical /Lab/ Tutorial	25
4.	Continuous Assessment - Theory	15
Total		**100**

For more details, please refer the applicable ordinance for this programme and Assessment SOPs.

CURRICULUM

Semester 1						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-101	DASH250001	Mathematical Foundations for AI	3	1	0	4
EE-BTCAIM-102	DCSE250113	Programming in Python	2	0	4	4
PC-BTCAIM-103	DOEE250033	Basic Electrical & Electronics Engineering	2	1	2	4
PC-BTCAIM-104	DOAI250044	Probability and Statistical Techniques	3	1	0	4
AE-BTCAIM-105	DASH250016	Communication Skills	1	0	2	2
OE-BTCAIM-106	Elective	Open Elective				4

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-201	DASH250028	Discrete Mathematics	3	1	0	4
PE-BTCAIM-202	Elective	Program Elective				4
PC-BTCAIM-203	DOEE250059	Digital Electronics	3	0	2	4
EE-BTCAIM-204	DCSE250055	Data Structures	3	0	2	4
VA-BTCAIM-205	DASH250030	Economics for Engineers	1	1	0	2
OE-BTCAIM-206	Elective	Open Elective				4

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-301	DCSE250060	Database Management Systems	2	0	4	4
PC-BTCAIM-302	DCSA250057	Object Oriented Programming using C++	3	0	2	4
PC-BTCAIM-303	DOEE250044	Computer Organization and Architecture	3	0	2	4
PC-BTCAIM-304	DOAI250015	Data Analytics and Visualization	2	0	4	4
AE-BTCAIM-305	DASH250026	Design Thinking	1	0	2	2
PE-BTCAIM-306	Elective	Program Elective				4
AU-BTCAIM-307	Elective	Audit Elective				0

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-401	DCSE250120	Theory of Computation	3	1	0	4
PC-BTCAIM-402	DOAI250022	Data Science and Machine Learning	3	0	2	4
PC-BTCAIM-403	DCSE250102	Operating Systems	3	0	2	4
VA-BTCAIM-404	DASH250104	Technical Presentation and Report Writing	0	0	4	2
OE-BTCAIM-405	Elective	Open Elective				4
EE-BTCAIM-406	EE	Minor Project-I				4

Semester 5						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-501	DOAI250012	Business Intelligence	2	0	4	4
PC-BTCAIM-502	DOAI250062	Applied Ethics for AI	3	1	0	4
PC-BTCAIM-503	DCSE250035	Computer Networks	3	0	2	4
AE-BTCAIM-504	DOEE250038	Capstone Project planning	1	0	2	2
PE-BTCAIM-505	Elective	Program Elective				4
EE-BTCAIM-506	EE	Internship				4

Semester 6						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-601	DOAI250041	Natural Language Processing	3	0	2	4
PC-BTCAIM-602	DCSA250020	Computer Vision	3	0	2	4
PC-BTCAIM-603	DOAI250053	Attention Based Architectures	3	0	2	4
AE-BTCAIM-604	DOEE250037	Capstone Project execution and Report writing	1	0	2	2
VA-BTCAIM-605	DASH250091	Professional Values and Ethics	2	0	0	2
PE-BTCAIM-606	Elective	Program Elective				4

Semester 7						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BTCAIM-701	DCSE250004	Advanced Data Structures and Algorithms	3	0	2	4
PC-BTCAIM-702	DCSA250071	Software Engineering	3	1	0	4
PC-BTCAIM-703	DOAI250027	Generative AI	3	0	2	4
PC-BTCAIM-704	DCSA250031	Full Stack Web Development	3	0	2	4
PE-BTCAIM-705	Elective	Program Elective				4
EE-BTCAIM-706	EE	Mini Project				5

Semester 8						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-BTCAIM-801	EE	Project and Thesis				10

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DASH250099	Social Networking and Mining	3	1	0	4	
DCSA250003	Mobile Application Development using Android	2	1	2	4	
DCSA250059	Object-Oriented Programming with Java	2	0	4	4	
DCSA250068	Soft Computing	3	0	2	4	
DCSA250081	Web Development using Python and Django	2	0	4	4	
DCSE250020	Cloud Computing	3	0	2	4	
DOAI250018	Data Mining	3	0	2	4	
DOAI250026	Data Analysis Using Spreadsheet	3	0	2	4	
DOAI250029	Artificial Intelligence with Python	2	0	4	4	
DOAI250043	Pattern Recognition	3	0	2	4	
DOAI250046	R Programming	2	0	4	4	
DOAI250047	Recommender Systems	3	1	0	4	
DOAI250048	Reinforcement Learning	3	0	2	4	
DOAI250065	AI for Cybersecurity	3	0	2	4	
DOEE250121	Internet of Things	3	0	2	4	
DOEE250127	IOT: Technologies and Applications	3	0	2	4	

Elective Courses for OE Category
Students may opt courses under MOOC, NPTEL, SWAYAM, NSQF/NCVET aligned courses or other relevant technical subjects not included in their core curriculum. The exercise of option shall be subject to the Course Schedule of the respective Constituent Unit, the credit requirements and availability of seats. Guidelines for choosing MOOC/Online courses are given separately and shall have to be adhered to.

Elective Courses for AU (Audit) Category					
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**
DASH240027	Disaster Management	2	0	0	0
DASH240047	English for Research Paper Writing	2	0	0	0
DASH240052	Environmental Science	3	0	0	0
DASH240085	Pedagogy Studies	2	0	0	0
DASH240086	Personality Development through Life Enlightenment Skills	2	0	0	0
DASH240097	Sanskrit for Technical Knowledge	2	0	0	0
DASH240103	Stress Management by Yoga	2	0	0	0
DASH240104	Technical Presentation and Report Writing	0	0	2	0
DASH240106	Value Education	2	0	0	0
DASH240124	Constitution of India - AU	2	0	0	0
DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0

DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0
DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0
DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0
DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

The choice of all type of Electives available shall be as per the Course Schedule of the respective Constituent Unit.

Guidelines for Massive Open Online Courses (MOOC) / approved Online Courses

1. Relevant Swayam, MOOC, NPTEL, NIELIT NSQF and any other online courses approved by UGC/AICTE shall also be allowed as Open Electives(OE).

2. One credit of MOOC course should be minimum of 30 hours.

3. A student can choose any approved online course as Open Elective, subject to minimum credit requirements.

4. The students willing to opt OE under MOOC in their next semester, will submit their request to the MOOC Coordinator at their respective campus.

5. Once approved, the campus shall display the list of accepted courses.

6. The student has to submit certificate issued by the institution offering the MOOCs course, along with the number of credits and grades, to get credits transferred into his/her marks certificate issued by NDU.

7. The transfer of credits shall be as adopted by NDU.

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DASH250001	Mathematical Foundations for AI	3	1	0	4

Course Outcomes:

CO1 : Apply principles of linear algebra to understand AI models and representations.
CO2: Analyze and compute probabilistic models using random variables,distributions, and Bayes' theorem for AI/ML applications.
CO3: Evaluate datasets using statistical inference techniques including estimation, confidence intervals, and hypothesis testing to make data-driven decisions.
CO4 : Evaluate learning performance and entropy-based decision making using information theory.
CO5 : Model real-world AI problems using mathematical tools and numerical methods.

Module 1:

Linear Algebra for AI (9 Hours)
Vectors (vector spaces, norms, loop), matrices, matrix operations, rank, inverse, determinant, eigenvalues and eigenvectors, orthogonality, singular value decomposition (SVD), applications in Principal Component Analysis (PCA), word embeddings, image recognition.

Module 2:

Probability (9 Hours)
Basic rules and axioms events, sample space, dependent and independent events, conditional probability, Random variables- continuous and discrete, expectation, variance, distributions- joint and conditional, Bayes' Theorem, Popular distributions- binomial, Bernoulli, poisson, exponential, Gaussian

Module 3:

Statistics (8 Hours)
Fundamentals of Data: Collection, Summarization, and Visualization; Sampling and Sampling Distributions, Central Limit Theorem; Methods of Estimation, Unbiased estimators; Confidence Interval Estimation: Z-interval, t-interval; Hypothesis Testing, Types of Errors, Rejection Region Approach and p-value Approach.

Module 4:

Information Theory (8 Hours)
Measure of Information, Entropy, Source coding theorem - Shannon-Fano codes & Huffman codes, Discrete Memoryless channel, Mutual information, Channel Capacity, Shannon-Hartley theorem

Module 5:

Mathematical Tools for AI Applications (11 Hours)
Markov chains and Markov Decision Processes (MDPs), matrix factorization in recommender systems, basic game theory, numerical methods: interpolation, numerical integration, applications in Non-linear Programming (NLP), reinforcement learning, and generative models.

Labs / Practicals:

N/A

References:

1.S. Kumaresan – Linear Algebra: A Geometric Approach, PHI Learning.
2.B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers.
3.Pankaj Jalote – Mathematical Foundations of Computer Science, Wiley India.

4.K. D. Joshi – Foundations of Discrete Mathematics, New Age International.
5.Marc Deisenroth, A. Faisal, C. Ong – Mathematics for Machine Learning, Cambridge University Press.
6.Shai Shalev-Shwartz & Ben-David – Understanding Machine Learning: From Theory to Algorithms.

Course Code	Course Name	L	T	P	Cr
DCSE250113	Programming in Python	2	0	4	4

Course Outcomes:

CO1: Remember syntax rules, basic operators, and built-in features of Python 3.

CO2: Understand guiding principles and coding conventions from Zen of Python.

CO3: Apply object-oriented programming paradigms in design and practice.

CO4: Analyze the runtime efficiency and resource utilization of Python programs.

CO5: Evaluate the outcomes of programs manually with dry runs and trace tables.

CO6: Create modular and maintainable programs following Pythonic practices.

Module 1:

Constants, types, identifiers, keywords, comments, print, input, help functions.
Operator precedence and associativity, arithmetic operators (**, *, /, //, %, +, -).
Binary number system, bin function, bitwise operators (shift, &, ^, |), bit masking.
Relational and equality operators (== vs "is"), not, short-circuit evaluation (and, or).

Module 2:

Selection: if, elif, else, mutual exclusion, indent (space vs tab), nesting, pass.
Iteration: while loop, range of integers, in operator, for loop, break vs continue.
Function: parameters, arguments, return value, global, recursion, lambda, closure.
Modules: from, import, as, __name__, top-level module, avoiding circular imports.

Module 3:

ASCII and UTF-8, ord and chr functions, escape sequence, strings, docstring, regex.
Mutability, id function, object vs reference, is vs ==, interning, datetime module.
Ordered: list, tuple, negative indexing, slicing, sort with key, pack and unpack.
Unordered: set, dictionary (keys, values, items), subscript vs get method, None.

Module 4:

Class definition, self parameter in methods, constructor, attributes, instantiation.
Dot notation, type function, __str__ vs __repr__, methods for built-in operators.
Static attributes, annotations, private name mangling, getter and setter methods.
Inheritance, object, super, C3 linearization, ambiguous MRO, abstract base class.

Module 5:

Syntax errors vs runtime exceptions, traceback, try, except, else, finally, raise.
Matching except clause, except with multiple exceptions, finally with return.
Built-in exceptions, user-defined exception classes, exceptions with arguments, as.
Command-line arguments, file opening modes (read/write/append), with statement.

Labs / Practicals:

01. Test the precedence and associativity of arithmetic operators.

02. Test the bitwise operators and verify the results with bin function.

03. Test the relational operators (with and without chaining).

04. Test the short-circuit evaluation of and, or operators using function calls as operands.

05. Convert from metric prefix (KB/MB/GB/TB) to binary prefix (KiB/MiB/GiB/TiB).

06. Use if-elif-else ladder to print the grade as per the score obtained in a subject.

07. Find the mean, median, mode, variance, and standard deviation for a list of scores.

08. Print values using additive/subtractive rules of roman numerals: I, V, X, L, C, D, M.

09. Use string repetition with multiplication to generate patterns without nested loops.

10. Use nested loops to generate asymmetric and symmetric patterns of arbitrary size.

11. Check whether a given string is a palindrome (or not) without reversing the string.

12. Test positional arguments, default arguments, and keyword arguments for functions.

13. Write a function to check whether an indexed collection of numbers is already sorted.

14. Write a function for primality testing, and use it to find first n pairs of twin primes.

15. Write a function to generate the Collatz sequence (up to 1) for a positive integer.

16. Generate the terms of Fibonacci sequence using iterative and recursive approaches.

17. Print optimal sequence of moves for a given instance of “Towers of Hanoi” puzzle.

18. Find all combinations to make a given payment from a limited set of coin types.

19. Check if a letter is a vowel using two approaches: if-elif-else and in operator.

20. Find vowel count in a string using a lookup table for case-insensitive vowel check.

21. Write a regular expression to check whether a string is a valid email ID (or not).

22. Use case-insensitive and dot-insensitive matching to check if two email IDs are same.

23. Display a countdown timer in HH:MM:SS format using carriage return ("\r").

24. Get current time and find clockwise angles of hour hand, minute hand, second hand.

25. Find the day of week for any date in Gregorian calendar using doomsday algorithm.

26. Define Matrix class and implement these operators: unary +, unary -, +, -, *, ==.

27. Define Person class in a module and extend it with Student class in another module.

28. Incrementally generate roll numbers in Student constructor using a static counter.

29. Populate a list with Student instances and write their attributes in a new CSV file.

References:

1. "Python Tutorial" by Guido van Rossum and the Python development team
https://bugs.python.org/file47781/Tutorial_EDIT.pdf

2. "Think Python: How to Think Like a Computer Scientist (Second Edition)" by Allen Benjamin Downey
https://greenteapress.com/thinkpython2/thinkpython2.pdf

3. "Python for Everybody: Exploring Data in Python 3" by Charles Russell Severance
https://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf

4. "Beej's Guide to Python Programming" by Brian Hall
https://beej.us/guide/bgpython/pdf/bgpython_a4_c_2.pdf

5. "Learning Python: Powerful Object-Oriented Programming (Sixth Edition)" by Mark Lutz
https://learning-python.com

6. "Fluent Python: Clear, Concise, and Effective Programming (Second Edition)" by Luciano Ramalho
https://www.fluentpython.com

Course Code	Course Name	L	T	P	Cr
DOEE250033	Basic Electrical & Electronics Engineering	2	1	2	4

Course Outcomes:

CO1: Analyze basic DC and AC electrical circuits using Ohm's Law, Kirchhoff's Laws, and network theorems.
CO2: Understand and explain power generation methods, transmission, and distribution, and describe the components of an electrical power system.
CO3: Identify and describe the construction and working principles of transformers, DC machines, and induction motors.
CO4: Analyze semiconductor devices like diodes, BJTs, and their applications in rectifiers, clippers, and amplifiers.
CO5: Explain the structure and operation of MOSFETs and CMOS logic circuits and their use in embedded and VLSI systems.

Module 1:

Introduction to Electrical Systems and DC Circuits
Basic concepts of electric current, voltage, power, and energy. Ohm's Law, Kirchhoff's Laws (KCL and KVL). Series and parallel circuits. Mesh and nodal analysis. Star-delta transformations. Energy sources: ideal and practical voltage and current sources.
Introduction to Power Generation and Distribution: Elementary idea of generation methods (thermal, hydro, solar, nuclear), transmission and distribution of electrical power; structure of power system; role of substations and transformers.

Module 2:

AC Circuits and Measurement
Sinusoidal voltage and current: amplitude, frequency, phase, time period. RMS and average values, form factor, peak factor. Phasor representation of sinusoidal quantities. AC analysis of pure R, L, and C components. RLC series and parallel circuits. Power in AC circuits: real, reactive, and apparent power. Power factor and its significance. Basics of three-phase systems.
Basic measuring instruments: voltmeter, ammeter, wattmeter, and energy meter.

Module 3:

Electrical Machines and Domestic Wiring
Magnetic circuits and electromagnetic induction. Transformers: principle, construction, EMF equation, efficiency, and applications. DC Machines: construction, types (motor/generator), working principle and applications. Three-phase induction motors: construction and working.
Basic concepts of domestic wiring: types of wiring, star case wiring, fuses, circuit breakers, earthing.

Module 4:

Semiconductor Devices and Applications
Semiconductors: intrinsic and extrinsic types. PN junction diode: V-I characteristics, rectifiers (half-wave and full-wave), clippers and clampers. Zener diode: breakdown and voltage regulation. BJT: construction, input-output characteristics in CE mode, application as a switch and amplifier. Basic introduction to optoelectronic devices (LED, photodiode).

Module 5:

MOSFET and CMOS Technology
MOSFET: structure, operation of nMOS and pMOS, output and transfer characteristics.

MOSFET as a switch and amplifier. CMOS logic: complementary MOS operation, CMOS inverter, static and dynamic behavior. Advantages of CMOS technology: low power, high noise margin, scalability. Applications of MOSFETs and CMOS in logic circuits, embedded systems, and VLSI design.

Labs / Practicals:

Electrical Engineering Experiments

1. Verification of Ohm's Law and Kirchhoff's Laws (KVL & KCL)
o Study linear resistive networks and validate current-voltage relationships.
2. Measurement of Power and Power Factor in a Single-Phase AC Circuit
o Calculate real, reactive, and apparent power using wattmeters and determine power factor.
3. Series and Parallel RLC Circuit Analysis
o Observe resonance and calculate impedance, current, and phase angle.
4. Load Test on Single-Phase Transformer
o Measure efficiency and voltage regulation under different loads.
5. Speed Control and Operation of a DC Motor
o Study working, direction control, and variation of speed with field/armature voltage.
6. Star Case Wiring and Testing of Light/Fan Circuit

• Perform practical wiring of a basic domestic setup using star configuration; verify connections and functionality.

Electronics Engineering Experiments

7. V-I Characteristics of PN Junction Diode
o Plot forward and reverse bias curves; identify cut-in and breakdown voltages.
8. Half-Wave and Full-Wave Rectifier with and without Filter
o Measure output DC voltage, ripple factor, and waveform using CRO.
9. Zener Diode as Voltage Regulator
o Test voltage regulation with varying load and input voltage.
10. Input and Output Characteristics of BJT in CE Configuration
o Measure current gain (β) and plot characteristics for amplification.
11. MOSFET Characteristics and CMOS Inverter Operation

References:

1. V.K. Mehta, Rohit Mehta – Principles of Electrical Engineering and Electronics, S. Chand Publishing
2. D.P. Kothari, I.J. Nagrath – Basic Electrical and Electronics Engineering, McGraw-Hill Education
3. M.S. Sukhija, T.K. Nagsarkar – Basic Electrical and Electronics Engineering, Oxford University Press

Course Code	Course Name	L	T	P	Cr
DOAI250044	Probability and Statistical Techniques	3	1	0	4

Course Outcomes:

CO1: Summarize and represent raw data using graphical tools, central tendency, dispersion, and shape measures for effective data interpretation.
CO2: Understand probability concepts and apply probability functions to analyse discrete and continuous random variables.
CO3: Identify and apply standard probability distributions and generating functions to solve problems involving random events and statistical modelling.
CO4: Analyse joint distributions, apply transformation techniques, and interpret multivariate dependence using covariance, correlation, and conditional expectations.
CO5: Compute and interpret moments, covariance and correlation matrices, and understand Central Limit Theorem.

Module 1:

Descriptive Statistics & Data Representation
Variables and Raw Data
Graphical Representation: Plots, charts, histograms, frequency polygons
Frequency Distributions: Relative, cumulative, and frequency curves
Measures of Central Tendency: Mean, median, mode
Measures of Dispersion: Range, standard deviation, quartile deviation, mean/median absolute deviation
Measures of shape: Skewness and Kurtosis

Module 2:

Probability Fundamentals and Random Variables
Introduction to Probability: Sample space, events, axioms
Random Variables: Definition, types (discrete/continuous), independence
Probability Functions: PMF (Probability Mass Function), PDF (Probability Density Function), CDF (Cumulative Distribution Function)
Joint Probability Distributions: Marginal and conditional distributions
Expectation and Variance: Linearity of expectation, covariance, correlation

Module 3:

Probability Distributions and Generating Functions
Discrete Distributions: Binomial, Poisson (properties, applications)
Continuous Distributions: Normal, Uniform, Exponential, Gamma, Beta
Relationship between Binomial and Normal, Poisson and Normal distributions
Generating Functions: Probability generating functions (PGF), Moment generating functions (MGF), Characteristic functions (properties and applications)

Module 4:

Multivariate Distributions and Transformations
Bivariate Random Variables: Joint distributions, conditional distributions.
Transformation Techniques: Single variable , Multiple variables (Jacobian method)
Conditional Expectation and Variance
Covariance and Correlation.

Module 5:

Moments: Raw vs. central moments, interpretation of Moments.
Covariance Matrix: Definition, properties, and eigenvalues.
Correlation Matrix, Central Limit Theorem.

Labs / Practicals:

N/A

References:

1. Fundamentals of Mathematical Statistics by S.C. Gupta & V.K. Kapoor
2. Probability, Random Variables, and Stochastic Processes by Athanasios Papoulis and S. Unnikrishna Pillai (4th Ed.)
3. An Introduction to Probability and Statistics by Vijay K. Rohatgi & A.K. Md. Ehsanes Saleh (3rd Ed.)
4. Statistics for Management by Richard I. Levin & David S. Rubin.
5. Applied Multivariate Analysis by S. Dasgupta.

Course Code	Course Name	L	T	P	Cr
DASH250016	Communication Skills	1	0	2	2

Course Outcomes:

CO1 : Understand the need and importance of English language.
CO2 : Develop proficiency in the language.
CO3 : Able to use technology to enrich communication skills.
CO4 : Gain self-confidence with improved command over English.
CO5 : Apply English communication skills effectively in academic, professional, and social environments.

Module 1:

Communication Fundamentals
Communication ;Meaning, Process and importance of Communication, Types and Channels of communications , Scope and Significance of Listening ,Speaking, Reading, Writing Skills.

Module 2:

Writing Skills
Basics of Grammar: Parts of Speech, Uses of Tenses, Active Passive, Narration.

Module 3:

Vocabulary Building
Word Formation, Synonyms , Antonyms, Words often Confused, One Word Substitutes, Idioms and Phrasal verbs, Abbreviations of Scientific and Technical Words.

Module 4:

Speaking Skills
Introduction to Phonetic Sounds & Articulation, Word Accent, Rhythm and Intonation, Interpersonal Communication, Oral Presentation, Body Language and Voice Modulation (Para linguistics and Non Verbal), Negotiation and Persuasion, Group Discussion, Interview Techniques (Telephonic and Video conferencing).

Module 5:

Technical Writing
Job application, CV writing, Business Letters, Notices, Report Writing, Minutes, E-mail Etiquette, Blog Writing.

Labs / Practicals:

1. Introducing Oneself, Exercise on Parts of Speech & Exercise on Tense.
2. Exercise on Agreement, Narration, Active Passive Voice & Dialogue Conversation.
3. Exercise on Writing Skills and Listening Comprehension.
4. Practice of Phonemes, Word Accent, Intonation, JAM Session.
5. Individual Presentation, Extempore and Picture Interpretation.
6. Vocabulary Building Exercises (One Word Substitute, Synonyms, Antonyms, Words Often Confused etc.) & Group Discussion.

References:

1. "The Essence of Effective Communication", Ludlow R. and Panton F., Pubs: Prentice Hall.
2. “Effective Communication Skills”, Kulbhushan Kumar, Khanna Publishing House.
3. “A University Grammar of English”, Quirk R. and Sidney G., Pubs: Pearson Education.
4. “High School English Grammar”, Wren and Martin, Pubs: S. Chand & Company Ltd.
5. “Essentials of Business Communication”, Guffrey M.E., Pubs: South-Western College Publishing.
6. “Technical Communication: Principles and Practice”, Raman M. and Sharma S., Pubs: Oxford University Press.
7. “Effective Business Communication”, Rodrigues M.V., Pubs: Concept Publishing Company, Delhi.

8. “English Vocabulary in Use”, McCarthy M. and Felicity O’ Dell.

Course Code	Course Name	L	T	P	Cr
DASH250028	Discrete Mathematics	3	1	0	4

Course Outcomes:

CO1 : Apply propositional and predicate logic in the formalization of mathematical and computer science problems.
CO2 : Solve number-theoretic and combinatorial problems using modular arithmetic and binomial identities.
CO3 : Implement counting strategies and solve discrete problems using pigeonhole and inclusion-exclusion principles.
CO4 : Analyze and model partially ordered sets and apply lattice and Boolean algebra in digital logic design.
CO5 : Understand and apply graph theory concepts to real-world problems such as routing and the Travelling Salesman Problem.

Module 1:

Logic and Set Theory (8 hrs)
Propositional and Predicate Logic: Statements, Predicates and quantifiers, nested quantifiers, Logical connectives, truth tables, logical equivalences
Set Theory: Sets and set operations, Venn diagrams, set identities
Functions: definitions, domain, codomain, range, Inverse and composition of functions.

Module 2:

Number Theory and Arithmetic (8 hrs)
Basic Number Theory: Division algorithm, divisibility, Prime numbers and prime factorization, Greatest Common Divisor (GCD) and Least Common Multiple (LCM)
Modular Arithmetic: Congruence relations and properties, Applications in computing and cryptography.

Module 3:

Combinatorics and Binomial Principles (6 hrs)
Permutations and Combinations, Pigeonhole Principle: Basic and generalized form, Applications in problem-solving, Inclusion-Exclusion Principle, Binomial Theorem (without proof): General term, middle term, Applications in algebra and combinatorics, Pascal's Identity: Construction and properties.

Module 4:

Lattices and Boolean Algebra (10 hrs)
Partially ordered sets (Posets), Hasse diagrams
Lattices: Definition and basic properties, Bounded, distributive, and complemented lattices
Boolean algebra: Definition, identities, and operations
Boolean functions: Sum of Products (SOP), Product of Sum (POS) forms and simplification basics.

Module 5:

Relations and Graph Theory (8 hrs)
Relations: Types and properties: reflexivity, symmetry, transitivity, Representations of relations, Equivalence relations
Graphs: Basic terminology and types of graphs, Graph representations: adjacency matrix, incidence matrix, adjacency list, Connectivity, Eulerian and Hamiltonian paths and circuits, Travelling Salesman Problem (TSP): introduction, formulation.

Labs / Practicals:

N/A

References:

1. Rosen Kenneth H., Discrete Mathematics and its Applications, McGraw Hill.
2. Balakrishnan S., Discrete Mathematics, Laxmi Publications.
3. Babu Ram, Discrete Mathematics, Pearson Education / Vikas Publishing.
4. Kolman Bernard, Busby Robert C. and Ross Sharon Cutler, Discrete Mathematical Structures, Pearson Education.
5. Tremblay J.P. and Manohar R., Discrete Mathematics, McGraw Hill.
6. Deo Narsingh, Graph Theory with Applications, PHI Learning.
7. Liu C.L., Elements of Discrete Mathematics, McGraw Hill.

Course Code	Course Name	L	T	P	Cr
DOEE250059	Digital Electronics	3	0	2	4

Course Outcomes:

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

Module 1:

Number System and Logic Gates: Representation and interconversion among binary, octal, decimal, and hexadecimal number systems, Binary arithmetic operations: addition, subtraction, multiplication, and division, Binary Coded Decimal (BCD), Excess-3, and Gray codes, Signed and unsigned binary number representation,1's and 2's complement techniques for binary subtraction, Code conversions: Binary to BCD, Excess-3, and Gray codes, and vice versa, Logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR, Universal gates and implementation of basic logic functions using universal gates.

Module 2:

Boolean Algebra and Logic Simplification: Boolean laws and theorems, DeMorgan’s theorems, SOP and POS forms, Canonical forms, Karnaugh Map (2, 3, 4 variables) – minterms and maxterms, Quine-McCluskey method (basic idea), Logic expression simplification and gate minimization.

Module 3:

Combinational Logic Circuits: Adders: Half adder, full adder, ripple carry adder, carry look-ahead adder, Subtractors: Half and full subtractors, Comparators, Multiplexers (MUX) and De-multiplexers (DEMUX) Encoders and Decoders, BCD to 7-segment decoder, Design of combinational logic using MUX, DEMUX.

Module 4:

Sequential Logic Circuits: Latches and Flip-flops: SR, D, T, JK, Master-Slave, Characteristic tables and excitation tables, Edge and level triggering, Timing diagrams, Counters: Asynchronous (ripple) and Synchronous counters, Up/down counters, modulo-N counters, Shift registers: SISO, SIPO, PISO, PIPO, Ring counter and Johnson counter, Mealy and Moore Finite State Machines (FSMs).

Module 5:

Memory & Logic Families: Classification of memories: RAM, ROM, PROM, EPROM, EEPROM, Memory decoding and address space, Overview of programmable logic devices: PLA, PAL, CPLD, FPGA (introductory), Logic families: TTL, CMOS. Characteristics: propagation delay, power dissipation, noise margin, fan-out, fan-in.

Labs / Practicals:

1. Verification of Logic Gates: Implement and verify the truth tables of AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.
2. Universal Gates Implementation: Realize basic logic functions using only NAND or NOR gates.
3. Boolean Expression Simplification: Simplify given Boolean expressions and implement the minimized circuits using logic gates.
4. Design of Half Adder and Full Adder: Construct and test circuits for half and full adders using logic gates.
5. Design of Half Subtractor and Full Subtractor: Implement and verify half and full subtractor circuits.
6. Multiplexer and Demultiplexer: Design and test 4:1 MUX and 1:4 DEMUX circuits.
7. Flip-Flop Implementations: Implement and observe SR, D, T, and JK flip-flops using basic gates or ICs.
8. Counters: Design and test asynchronous and synchronous up/down counters (e.g., mod-8, mod-10).
9. Shift Registers: Construct and analyze SIPO, SISO, PIPO, and PISO shift registers.

10. BCD to 7-Segment Display: Design a decoder to display BCD inputs on a 7-segment LED display.

References:

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

Course Code	Course Name	L	T	P	Cr
DCSE250055	Data Structures	3	0	2	4

Course Outcomes:

CO1: Remember the different types of linear and non-linear data structures.

CO2: Understand how structuring of data can reduce the average access time.

CO3: Apply concrete data structures to implement abstract data structures.

CO4: Analyze the fundamental operations supported by various data structures.

CO5: Evaluate the number of steps required to do an operation on a data structure.

CO6: Create efficient implementations that are suitable for real-world applications.

Module 1:

Multi-dimensional arrays, addressing in row-major and column-major orderings.
Searching: linear search, binary search, uniform binary search, interpolation search.
Sorting by comparison: selection, bubble, insertion, Shell sort, quicksort, mergesort.
Non-comparison sorting: counting sort, pigeonhole sort, radix sort (LSD and MSD).

Module 2:

Linked lists types: linear and circular, unidirectional and bidirectional.
Operations on linked lists: create, insert, delete, traverse, reverse, find.
Open address hashing, collision, linear probing, quadratic probing, double hashing.
Hashing with separate chaining, load factor, applications of hash tables.

Module 3:

Abstract data structures, concrete implementation using arrays and linked lists.
Stacks: LIFO access, operations (push, pop, top), call stack, expression evaluation.
Queues: FIFO access, operations (enqueue, dequeue, peek), linear queue, ring buffer.
Double-ended queue (input-restricted, output-restricted), priority queue, DEPQ.

Module 4:

Graphs (undirected and directed), adjacency matrix, adjacency list, weighted graphs.
Graph traversals: breadth-first search, depth-first search, iterative deepening search.
Graph connectivity, directed acyclic graphs, topological sorting, spanning trees.
Arborescence, branching factor, binary trees (skew, strict, complete, perfect).

Module 5:

Tree traversal: pre-order (NLR), in-order (LNR), post-order (LRN), reverse variants.
Binary heap (min-heap and max-heap), Fibonacci heap, binomial heap, heapsort.
Binary search tree (BST), sorting with BST, threaded BST (single and double).
Disjoint-set data structure, operations: makeset, find, union (by rank or by size).

Labs / Practicals:

Each data structure and supporting algorithms should be implemented in C or Python.

01. Use arrays to store univariate polynomials for doing addition and multiplication.

02. Write a function to find min/max element and use it to implement selection sort.

03. Implement an adaptive variant of bubble sort that runs in linear time for best case.

04. Implement insertion sort with a type-agnostic design (like qsort from stdlib.h).

05. Perform non-comparative sorting with counting sort and radix sort (LSD and MSD).

06. Create unidirectional and bidirectional linked lists with linear and circular variants.

07. Create open address hash tables with linear probe, quadratic probe, double hashing.

08. Create hash table with an array of pointers for separate chaining using linked lists.

09. Implement stack, queue (linear and circular), deque using arrays and linked lists.

10. Convert among prefix, infix, postfix expressions, and evaluate the postfix notation.

11. Perform breadth-first traversal, depth-first traversal, topological sort on a digraph.

12. Implement priority queue using binary max heap, supporting insertion and deletion.

13. Heapify an array of strings in linear time and do heap sort in lexicographic order.

14. Perform insert and delete on a binary search tree threaded for in-order traversal.

15. Implement disjoint-set data structure supporting makeset, find, union operations.

References:

1. "Think Data Structures" by Allen Benjamin Downey
https://greenteapress.com/thinkdast/thinkdast.pdf

2. "Open Data Structures: An Introduction" by Pat Morin
https://www.aupress.ca/app/uploads/120226_99Z_Morin_2013-Open_Data_Structures.pdf

3. "An Open Guide to Data Structures and Algorithms" by Paul W. Bible and Lucas Moser
https://pressbooks.palni.org/anopenguidetodatastructuresandalgorithms

4. "Computer Science II" by Chris Bourke
https://cse.unl.edu/~cbourke/ComputerScienceTwo.pdf

5. "Data Structures and Algorithms" by Alfred Vaino Aho, John Edward Hopcroft, Jeffrey David Ullman
https://dl.acm.org/doi/10.5555/577958

6. "Data Structures and Algorithm Analysis in C (Second Edition)" by Mark Allen Weiss
https://dl.acm.org/doi/10.5555/248735

Course Code	Course Name	L	T	P	Cr
DASH250030	Economics for Engineers	1	1	0	2

Course Outcomes:

CO1 : Understand basic economic concepts, including demand, supply, cost, and market structures.
CO2 : Analyze the economic feasibility of engineering projects using cost-benefit and break-even analysis.
CO3 : Evaluate different investment alternatives using engineering economic tools like NPV, IRR, and Payback Period.
CO4 : Apply knowledge of macroeconomic indicators and fiscal/monetary policies in the context of engineering.
CO5 : Develop the ability to assess the economic impact of engineering decisions in real-world scenarios.

Module 1:

Basic Economic Concepts
Introduction to Economics: Micro vs Macro, Utility, Demand and Supply Analysis, Elasticity of Demand and Supply, Equilibrium, Law of Diminishing Returns, Opportunity Cost.

Module 2:

Cost and Production Analysis
Types of Costs: Fixed, Variable, Marginal, Average, Total, Cost-Output Relationship, Break-even Analysis, Law of Variable Proportions, Returns to Scale, Cost-Volume-Profit Analysis.

Module 3:

Market Structures and Pricing
Types of Market: Perfect Competition, Monopoly, Oligopoly, Monopolistic Competition, Pricing Strategies, Price Discrimination, Government Policies and Pricing.

Module 4:

Engineering Economics
Time Value of Money, Present Worth and Future Worth, Net Present Value (NPV), Internal Rate of Return (IRR), Payback Period, Comparison of Investment Alternatives, Depreciation and Inflation Considerations.

Module 5:

Macroeconomic Concepts
National Income, GDP, GNP, Inflation, Deflation, Unemployment, Fiscal and Monetary Policy, Role of RBI and Government in Economic Planning, Economic Indicators Relevant to Engineers.

Labs / Practicals:

N/A

References:

1. R. Panneerselvam, Engineering Economics, PHI Learning.
2. D.M. Mithani, Managerial Economics, Himalaya Publishing House.
3. M.L. Jhingan, Microeconomic Theory, Vrinda Publications, Latest Edition.
4. Chan S. Park, Contemporary Engineering Economics, Pearson Education.
5. Paul A. Samuelson and William D. Nordhaus, Economics, McGraw Hill.

Course Code	Course Name	L	T	P	Cr
DCSE250060	Database Management Systems	2	0	4	4

Course Outcomes:

CO1. Understand the basic database concepts, data models, schemas and instances.
CO2. Learn database design using Entity Relationship model and learn the use of constraints and relational algebra operations.
CO3. Gain proficiency in SQL and construct queries using SQL.
CO4. Understand the importance of normalization in database design.
CO5. Understand transaction processing, concurrency control and recovery

Module 1:

Introduction to Databases, File systems vs. Database systems, Characteristics of DBMS, Applications of DBMS, Advantages of using a DBMS, Database users and administrators, Three-level architecture of DBMS (ANSI/SPARC architecture), Data independence, Database schema, Instance, Generalization, Specialization, Structure of a DBMS, Overview of DBMS software (e.g., MySQL, SQL Server, Oracle, PostgreSQL)

Module 2:

Introduction to Data models: Hierarchical, Network, Relational, Object-oriented, Data Modeling using Entity Relationship (ER) model: entities, entity types, attributes, relationships, relationship types, E/R diagram notation, examples.
Relational data model: Concepts, schema, instances, keys, constraints, Relational algebra: Select, project, union, set difference, Cartesian product, joins, ER to Relational mapping, Relational algebra and relational calculus, Tuple Relational Calculus, Domain Relational Calculus

Module 3:

Structured Query Language (SQL): Basics of SQL, DDL, DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, Functions - aggregate functions, Built-in functions – numeric, date, string functions, set operations, sub-queries, correlated sub-queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types. Transaction control commands – Commit, Rollback, Save point, Cursors, Indexes, stored procedures, Triggers

Module 4:

Normalization – Introduction, Non loss decomposition and functional dependencies, First, Second, and third normal forms – dependency preservation, Boyce/Codd normal form.
Higher Normal Forms - Introduction, Multi-valued dependencies and Fourth normal form, Join dependencies and Fifth normal form.

Module 5:

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

Labs / Practicals:

1. Create a database, define tables using DDL, apply constraints (Primary, Foreign, Not Null, Unique).
2. Insert, update, delete, and retrieve records using SQL DML commands.
3. Write SQL queries using WHERE, ORDER BY, GROUP BY, HAVING, LIKE, etc.
4. Write SQL queries to demonstrate the use of INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, and SELF JOIN using appropriate sample tables.
5. Write nested subqueries with IN, EXISTS, and ANY/ALL.

6. Draw ER diagram for a sample application (e.g., Library or Hospital), map it to relational schema.
7. Normalize sample unstructured data up to 3NF or BCNF. Implement before-and-after schemas in SQL.
8. Create and execute stored procedures/functions for common operations.

References:

1. Silberschatz, Korth, Sudarshan – Database System Concepts, McGraw Hill Year: 2020 (7th edition).
2. Elmasri and Navathe – Fundamentals of Database Systems, Pearson Edition 2016
3. Raghu Ramakrishnan, Johannes Gehrke – Database Management Systems, McGraw Hill
4. Database Systems Concepts Author H.f.Korth and Silberschatz Publisher McGraw Hill
5. Database System Author Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom Publisher Pearson Edition 2nd Edition
6. C.J. Date – An Introduction to Database Systems, Pearson

Course Code	Course Name	L	T	P	Cr
DCSA250057	Object Oriented Programming using C++	3	0	2	4

Course Outcomes:

CO1. Describe the different features of Object Oriented Programming.
CO2. Write programs to Implement the concepts of classes and objects.
CO3. Create new classes using the concepts of inheritance.
CO4. Apply knowledge of Polymorphism to solve real life problems.
CO5. Implement exception handling mechanism.

Module 1:

Introduction to Object Oriented Programming: Object Oriented Paradigm Objects and Classes, Features Object oriented Programming, Structured Vs Object Oriented Development, Features of Object Oriented Languages, Applications ofObjectOriented Programming, Merits and Limitations of Object Oriented Programming.Basic Data types, Basic Type modifiers, Derived Data types, Variables, Storage class specifiers, Initializing variables, Operators, Unformatted Console and stream I/O Functions, Formatted Console I/O Functions

Module 2:

Classes and Objects: Classes ,Class Members and Creating Objects, Member functions, Member Access Specifiers (public, private, protected), Static class member, Inline Functions, Arrays within a Class and Array of Objects, Passing Objects as function arguments and returning object from a function,Constructors, Overloaded Constructors, Null Contradictor, Copy Constructor, Destructors Constraints on Constructors and Destructors

Module 3:

Inheritance: Base and Derived classes, Accessing Base class members and Access Control, Overriding member functions, Multi Level, Multiple, Hierarchical& Hybrid Inheritance, Virtual Base Class.

Module 4:

Polymorphism: Fundamental of Polymorphism, Overloading Functions, Overloading Operators (Unary, binary, string manipulation using operator), Pointer to object and derived class, 'This' pointer, Virtual Functions, Early and Late Binding, Rules of Virtual Functions, Pure Virtual Function, Friend Functions.

Module 5:

File Handling, Exception Handling & Templates: Basic File Operations, File Handling, Classes for file stream operation, Opening and Closing Files, File modes File, Introduction to Exception Handling, Catching Class Types, Multiple Catch Handlers, Exception Specification, Generic Functions/Function Templates, Template Arguments.

Labs / Practicals:

1. Write a C++ program to demonstrate class and object creation.
2. Implement constructors and destructors in a C++ class.
3. Demonstrate function overloading and operator overloading in C++.
4. Create a program to implement single and multilevel inheritance.
5. Use virtual functions to implement runtime polymorphism.
6. Design a template class for stack or queue operations.
7. Demonstrate file handling using file streams in C++.
8. Implement a program using exception handling with multiple catch blocks.

References:

1. Object Oriented Programming With C++, E Balagurusamy
2. Object oriented programming in C++ , Robert Laffore
3. Introduction To Programming With C++, Diane Zak
4. Object oriented programming with C++ Reema Thareja

Course Code	Course Name	L	T	P	Cr
DOEE250044	Computer Organization and Architecture	3	0	2	4

Course Outcomes:

CO1: Explain processor fundamentals, instruction representation, and hardware operations including arithmetic and logical functions.
CO2: Analyze instruction set architectures, addressing modes, and computer arithmetic with parallelism and SIMD extensions.
CO3: Design basic datapath and pipelined architectures, and evaluate instruction-level parallelism and hazards in modern processors.
CO4: Demonstrate memory mapping techniques, cache design, memory hierarchy, and virtual memory mechanisms.
CO5: Evaluate memory management strategies including RAID levels, disk storage, multithreading, and multiprocessor systems.

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

Module 1 & 2: Processor Basics and Instruction Set
1. Simulation of Basic Arithmetic Operations in MIPS Assembly
o Objective: Write MIPS assembly code to perform addition, subtraction, multiplication, and division of two integers.
o Tool: MIPS Simulator (e.g., QtSpim, MARS)
2. Binary and Hexadecimal Conversion & Representation
o Objective: Implement a program (in C or Assembly) to convert between binary, hexadecimal, and decimal formats; demonstrate signed and unsigned number representations.

3. Instruction Encoding and Decoding
o Objective: Manually encode and decode sample MIPS instructions and verify using a simulator.

Module 3: Datapath and Pipelining
4. Design a Basic Arithmetic Logic Unit (ALU) Using VHDL/Verilog
o Objective: Implement an ALU capable of performing AND, OR, ADD, SUB, and SLT operations.
o Tool: Xilinx Vivado / ModelSim / Logisim
5. Simulate Single-Cycle and Multi-Cycle Datapaths
o Objective: Model a simplified processor datapath (single-cycle and pipelined) and compare performance.
o Tool: Digital Simulator / Logisim / Ripes (RISC-V simulator)
6. Pipeline Hazard Detection and Resolution
o Objective: Simulate and analyze data and control hazards in a pipelined processor; demonstrate stalling and forwarding.

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

References:

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

Course Code	Course Name	L	T	P	Cr
DOAI250015	Data Analytics and Visualization	2	0	4	4

Course Outcomes:

CO1: Understand the fundamentals of data analytics, data preprocessing, and feature engineering techniques.
CO2: Perform efficient data manipulation, aggregation, and wrangling using Pandas and NumPy.
CO3: Implement numerical computing techniques for high-level mathematical operations and image data analysis.
CO4: Create effective data visualizations using Matplotlib, Seaborn, and Pandas for data-driven insights.
CO5: Utilize advanced visualization tools like Excel, Tableau, and Power BI for interactive and real-time data representation.

Module 1:

Fundamentals of Data Analytics and Preprocessing
Definition, importance, and applications of data analytics. Introduction to structured and unstructured data, data collection, data storage, and preprocessing techniques. Data ingestion, handling missing values, data cleaning, transformation, and feature engineering. Overview of NumPy and Pandas libraries for data analysis, including DataFrames, Series, indexing, slicing, and filtering.

Module 2:

Data Manipulation and Aggregation
Operations on Pandas DataFrames and Series, data wrangling techniques, handling categorical and numerical data, working with missing and duplicate values. Data aggregation, applying functions in Pandas, groupby operations, pivot tables, and cross-tabulations. Merging and concatenating datasets, time series analysis, and handling large datasets efficiently.

Module 3:

Numerical Computing with NumPy
Creating and manipulating multidimensional arrays, broadcasting, vectorized operations, and high-level mathematical functions. Array slicing, fancy indexing, filtering, and statistical analysis using NumPy. Working with structured arrays, handling image data, and performing linear algebra operations for data science applications.

Module 4:

Data Visualization Techniques
Importance of data visualization, choosing appropriate visualization methods, visualizing data insights and findings. Data visualization using Matplotlib: line plots, bar charts, scatter plots, histograms, and subplots. Advanced visualization using Seaborn: pair plots, violin plots, heatmaps, and regression plots. Using Pandas for quick data visualization and integrating visualization tools in data science workflows.

Module 5:

Tools and Advanced Visualization Techniques
Visualization using Excel: charts, pivot tables, and dashboards. Introduction to interactive visualization tools such as Tableau and Power BI. Creating real-time and dashboard-based visualizations, interactive plots using Plotly, and geographic data visualization with Folium. Case studies and real-world applications of data analytics and visualization.

Labs / Practicals:

1. Implement data wrangling techniques like reshaping and pivoting using Pandas.
2. Create Data Frames and Series in Pandas and perform basic data manipulation operations.
3. Load datasets from different file formats (CSV, Excel, JSON) into Pandas.
4. Perform basic exploratory data analysis (EDA) using summary statistics and visualizations.
5. Create data visualizations using Matplotlib (e.g., line plots, histograms, bar charts).
6. Create advanced visualizations using Seaborn (e.g., heatmaps, pair plots, violin plots).
7. Implement box plots, scatter plots, and correlation matrices for data analysis.
8. Perform data aggregation and group-by operations to analyze grouped data.
9. Merge multiple Data Frames using different join operations in Pandas.
10. Implement linear regression model and visualize the results using Matplotlib.
11. Create and customize subplots using Matplotlib to present multiple visualizations.
12. Perform data aggregation using pivot tables and cross-tabulation in Pandas.
13. Create interactive plots using Plotly for web-based visualization.
14. Create a word cloud from a textual dataset to visualize the frequency of words.
15. Perform principal component analysis (PCA) for dimensionality reduction and visualize the result.
16. Visualize time series data and trends using Matplotlib and Seaborn.
17. Use Excel for data visualization, such as creating charts, graphs, and dashboards.
18. Implement correlation analysis and create a heatmap for understanding relationships between variables

References:

1. Python – Data Visualisation by Samuel Burns
2. Storytelling with Data: A Data Visualization Guide for Business Professionals by Cole NussbaumerKnaflic
3. Data Science for Business by Foster Provost and Tom Fawcett:
4. Data Visualization: A Practical Introduction by Kieran Healy
5. Hands-On Data Visualization: Interactive Storytelling from Spreadsheets to Code by Jack Dougherty and IlyaIlyankou
6. Data Visualization with Excel Dashboards and Reports by Dick Kusleika
7. Beautiful Visualization: Looking at Data Through the Eyes of Experts edited by Julie Steele and Noah Iliinsky

Course Code	Course Name	L	T	P	Cr
DASH250026	Design Thinking	1	0	2	2

Course Outcomes:

CO1 : Understand the core principles and process of design thinking including empathy, ideation, and prototyping.
CO2 : Apply user-centric research methods to identify and frame real-world problems.
CO3 : Generate and evaluate innovative ideas through brainstorming and collaborative design activities.
CO4 : Build and test low-fidelity prototypes to explore solution possibilities.
CO5 : Present and communicate design solutions effectively using appropriate tools and storytelling techniques.

Module 1:

Introduction to Design Thinking
Definition and Importance of Design Thinking, Comparison with Traditional Problem Solving, Mindsets for Innovation, Stanford d.school Design Process, Applications in Engineering, Business, and Social Sectors.

Module 2:

Empathize – Understanding Users
Importance of Empathy, Observation and Interview Techniques, User Persona Creation, Empathy Mapping, Journey Maps, Capturing Insights from Field Research.

Module 3:

Define and Ideate
Problem Definition and Framing: Point of View (POV) Statements, Ideation Techniques (Brainstorming, SCAMPER, Mind Mapping), Idea Evaluation and Selection Criteria.

Module 4:

Prototype and Test
Prototyping Tools and Techniques, Rapid Prototyping, Paper Prototypes, Role-Playing, Usability Testing, Feedback Integration and Iteration.

Module 5:

Communication and Implementation
Storyboarding, Pitching Ideas, Design Thinking for Entrepreneurship, Real-life Case Studies, Ethical and Sustainable Design Considerations.

Labs / Practicals:

1. Conduct user interviews and observations to develop empathy maps.
2. Create user personas and problem statements.
3. Brainstorm solutions for a chosen problem and organize ideas using mind maps.
4. Develop low-fidelity prototypes for the selected ideas.
5. Test prototypes with users and gather feedback for refinement.
6. Present project using storyboards and pitch decks.
7. Group project: Apply the design thinking process to a real-world challenge and document the journey.
8. Use tools like Canva, Miro, or Figma for visualizing and presenting ideas.
9. Role-play-based usability tests.
10. Peer review and reflection journal on the design process.

References:

1. Tim Brown, Change by Design, Harper Business.
2. Rolf Faste, Design Thinking Methodology, Stanford University Materials.
3. Jeanne Liedtka & Tim Ogilvie, Designing for Growth: A Design Thinking Toolkit for Managers, Columbia

Business School Publishing.
4. Nigel Cross, Design Thinking: Understanding How Designers Think and Work, Berg Publishers.
5. Plattner, Meinel, Leifer, Design Thinking: Understand – Improve – Apply, Springer.

Course Code	Course Name	L	T	P	Cr
DCSE250120	Theory of Computation	3	1	0	4

Course Outcomes:

CO1. Explain the basic concepts of automata, formal languages, and grammar types.
CO2. Design finite automata (deterministic and non-deterministic) and regular expressions for recognizing regular languages.
CO3. Design regular expressions for describing simple patterns and languages.
CO4. Understand and design pushdown automata for basic context-free languages.

Module 1:

Mathematical Preliminaries: Sets, Relations and Functions (Brief Discussion), Graphs, Trees. ,Strings and their properties: Definition, operation on strings, palindrome, prefix & suffix of a string, Levi theorem (Statement only), Terminal & Non-terminal symbol

Module 2:

Definition of an Automaton: Definition of finite Automaton, Block diagram of finite Automaton, Transition system, Properties of Transition Functions, Acceptability of a string by Finite Automaton. Definition of DFA and NDFA, The equivalence of DFA and NDFA. Mealy and Moore machine.

Module 3:

Formal Language: Concept of a language, Definition of a grammar, Language generated by a grammar (definition with application). Chomsky classification of languages (definition), Relation between the classified languages. Recursive and recursively enumerable set (definitions).

Module 4:

Regular Sets & Regular Grammar: Definition of Regular expression and regular set, Identities of regular expressions
Relation between regular expression and finite automata, Transition system containing /\-moves (application), Conversion of Non-deterministic systems to deterministic system (application), Construction of finite automata equivalent to a regular expression (with application)

Module 5:

Context-Free Languages & Pushdown Automata: Introduction – Definition – Derivation trees (Definitions & application) – Ambiguity in
CFG, Basic definition of PDA

Labs / Practicals:

NA

References:

1. "Foundations of Computation (Second Edition)" by Carol Critchlow and David Eck
https://math.hws.edu/FoundationsOfComputation/FoundationsOfComputation_2.3.2_8.5x11.pdf

2. "Automata Theory: An Algorithmic Approach" by Francisco Javier Esparza Estaun and Michael Blondin
https://michaelblondin.com/automata/files/book_authors.pdf

3. "Automata and Computability" by Dexter Campbell Kozen
https://www.cs.cornell.edu/kozen/Papers/481.pdf

4. "Introduction to Automata Theory, Languages, And Computation (First Edition)" by John Edward Hopcroft and Jeffrey David Ullman
http://infolab.stanford.edu/~ullman/ialc.html

5. "Introduction to the Theory of Computation (Third Edition)" by Michael Fredric Sipser
https://dl.acm.org/doi/10.5555/524279

6. "An Introduction to Formal Languages and Automata (Fifth Edition)" by Peter Linz
https://dl.acm.org/doi/10.5555/1995326

Course Code	Course Name	L	T	P	Cr
DOAI250022	Data Science and Machine Learning	3	0	2	4

Course Outcomes:

CO1: Understand the theoretical and practical fundamentals of data science, machine learning, and handling large-scale data.

CO2: Analyze and apply various machine learning algorithms including advanced supervised, unsupervised, and deep learning methods.

CO3: Utilize Python libraries such as pandas, scikit-learn, matplotlib, seaborn, TensorFlow, and PyTorch for model building and analysis.

CO4: Perform robust exploratory data analysis (EDA), feature engineering, and statistical testing for complex datasets.

CO5: Evaluate machine learning models using rigorous statistical metrics and advanced validation strategies including cross-validation, ROC, AUC, and model explainability tools.

CO6: Design and implement end-to-end data science solutions for real-world, domain-specific problems, with considerations of fairness, interpretability, and deployment strategies.

Module 1:

Introduction to Data Science, Python, and Mathematical Foundations

Definition, Applications, Lifecycle, and Tools for Data Science, Introduction to Python Ecosystem (pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, TensorFlow, PyTorch), Working with Jupyter Notebook and Google Colab, Review of Python: Functions, Control Flow, OOP, Data Structures, Linear Algebra for ML (vectors, matrices, eigenvalues, SVD), Probability Theory and Bayes Theorem essentials, Calculus basics for optimization, Optimization: Gradient Descent, SGD

Module 2:

Data Handling, EDA, and Feature Engineering

Importing and Managing Datasets, Data Cleaning and Preprocessing: Handling Missing Data, Outliers, Data Wrangling with pandas and NumPy, Data Visualization (Matplotlib, Seaborn, Plotly), Feature Engineering: Encoding (One-Hot, Label Encoding), Binning, Polynomial Features, Feature Scaling: StandardScaler, MinMaxScaler, PCA and t-SNE for Dimensionality Reduction, Handling Imbalanced Datasets: SMOTE, ADASYN

Module 3:

Statistics, Probability, and Model Evaluation Metrics

Descriptive Statistics, Central Tendency, Dispersion, Probability Distributions: Gaussian, Binomial, Poisson, Hypothesis Testing, p-values, Confidence Intervals, Correlation, Covariance, Accuracy, Precision, Recall, F1-Score, ROC-AUC, Confusion Matrix, Cross-Validation Techniques (K-Fold, Stratified K-Fold, LOOCV), Model Interpretability (SHAP, LIME)

Module 4:

Machine Learning Algorithms and Advanced Topics

Linear Regression, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forest, Gradient Boosting (XGBoost, LightGBM), Naïve Bayes, Support Vector Machines (SVM), Clustering: K-Means, Hierarchical, DBSCAN, Anomaly Detection Techniques, Dimensionality Reduction Techniques: PCA, LDA, t-SNE, UMAP, Bias-Variance Tradeoff, VC-Dimension (Conceptual), Curse of Dimensionality

Module 5:

Deep Learning, Ethics, and Model Deployment

Neural Networks (MLP, Activation Functions, Backpropagation), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), LSTM, Transfer Learning (using pretrained models), Hyperparameter Tuning: Grid Search, Random Search, Bayesian Optimization, AI Ethics, Bias, Fairness in ML Models, Explainable AI (XAI), Flask, Streamlit for ML apps, Cloud Deployment (Heroku, AWS, GCP basics), Mini-Project on Domain-Specific Dataset

Labs / Practicals:

1. Python Programming for Data Science and Analysis (NumPy, pandas, Matplotlib, Seaborn)

2. Advanced EDA and Visualization on Real Datasets

3. Machine Learning Model Building (Regression, Classification) using scikit-learn

4. Unsupervised Learning: Clustering, Dimensionality Reduction

5. Advanced Model Evaluation and Explainability (SHAP, LIME)

6. Deep Learning with TensorFlow / PyTorch: CNN Basics

7. End-to-End Project with Model Deployment (Streamlit / Flask)

8. Group Project: Addressing a Real-World Problem with Ethical Considerations

References:

1. Python for Data Analysis by Wes McKinney

2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron

3. Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido

4. Pattern Recognition and Machine Learning by Christopher Bishop

5. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

6. Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong

7. Official documentation for pandas (https://pandas.pydata.org/)

8. Official documentation for scikit-learn (https://scikit-learn.org/)

9. Official documentation for seaborn (https://seaborn.pydata.org/)

10. Official documentation for TensorFlow (https://www.tensorflow.org/)

11. Official documentation for PyTorch (https://pytorch.org/)

Course Code	Course Name	L	T	P	Cr
DCSE250102	Operating Systems	3	0	2	4

Course Outcomes:

CO1: Understand the basics of operating systems like kernel, shell, types and services of operating systems.
CO2:Understand the concept of program, process and thread and analyse various CPU scheduling Algorithms.
CO3: Describe and analyse the memory management and its allocation policies.
CO4: Understand the issues related to file system interface and implementation.
CO5: Understand disk management and explain disk scheduling algorithms for better utilization of external memory.
6: Configure OS in an efficient and secure manner.

Module 1:

Introduction: Overview of Operating System, basic concepts, UNIX/LINUX Architecture, Kernel, services and systems calls, system programs.

Module 2:

Process Management and Memory management: Process Management: Process concepts, operations on processes, IPC, Process Scheduling, Multithreaded programming. Memory management: Memory allocation, Swapping, Paging, Segmentation, Virtual Memory, various faults.

Module 3:

File management: Concept of a file, access methods, directory structure, file system mounting, file sharing and protection, file system structure and implementation, directory implementation, free space management, efficiency and performance. Different types of file systems.

Module 4:

I/O System: Mass storage structure - overview, disk structure, disk attachment, disk scheduling algorithms, swap space management, RAID types.

Module 5:

OS Security: Authentication, Access Control, Access Rights, System Logs

Labs / Practicals:

1. Revision practice of various commands like man, cp, mv, ln, rm, unlink, mkdir, rmdir, etc and many more that were learnt in IT Workshop course and later.
2. Implement two way process communication using pipes.
3. Implement message queue form of IPC
4. Implement shared memory and semaphore form of IPC
5. Simulate the CPU scheduling algorithms - Round Robin, SJF, FCFS, priority
6. Simulate all FIFO Page Replacement Algorithm using C program
7. Simulate all LRU Page Replacement Algorithms using C program
8. Simulate Paging Technique of Memory Management
9. Practice various commands/utilitiessuch as catnl, uniq, tee, pg, comm, cmp, diff, tr, tar, cpio, mount, umount, find, umask, ulimit, sort, grep, egrep,fgrep cut, paste, join, du, df , ps, who, etc and many more.

References:

1. "Operating System Concepts (Tenth Edition)" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne
https://os-book.com/OS10/index.html

2. "Operating Systems: Internals and Design Principles (Ninth Edition)" by William Stallings
http://williamstallings.com/OperatingSystems

3. "Modern Operating Systems (Fifth Edition)" by Andrew Stuart Tanenbaum and Herbert Bos
https://dl.acm.org/doi/10.5555/2655363

4. "Operating System Design: The Xinu Approach (Second Edition)" by Douglas Earl Comer
https://www.oreilly.com/library/view/operating-system-design/9781498712439

5. "The Design of the UNIX Operating System" by Maurice J. Bach
https://dl.acm.org/doi/10.5555/8570

6. "Advanced Programming in the UNIX Environment (Third Edition)" by William Richard Stevens and Stephen A. Rago
http://www.apuebook.com/apue3e.html

7. "Beej's Guide to Interprocess Communication" by Brian Hall
https://beej.us/guide/bgipc/pdf/bgipc_a4_c_2.pdf

Course Code	Course Name	L	T	P	Cr
DASH250104	Technical Presentation and Report Writing	0	0	4	2

Course Outcomes:

CO1 : Understand the principles and formats of technical communication including different types of reports and presentations.
CO2 : Prepare clear, concise, and well-structured technical documents for academic and professional purposes.
CO3 : Design and deliver effective oral presentations using appropriate tools and visual aids.
CO4 : Demonstrate proficiency in grammar, technical vocabulary, and formal style required in technical writing.
CO5 : Collaborate in teams to produce technical content and receive peer feedback constructively.

Module 1:

Introduction to Technical Communication
Principles and characteristics of technical communication, Purpose and types of technical documents, Features of a good technical report, Audience analysis.

Module 2:

Writing Skills for Technical Documents
Structure of reports: Title Page, Abstract, Table of contents, Introduction, Body, Conclusion, and References. Writing instructions, lab reports, Project Reports, and Proposals. Use of charts, graphs, and tables.

Module 3:

Grammar and Language in Technical Writing
Use of formal tone and objective language, sentence construction, common errors, punctuation, and vocabulary building for technical writing. Editing and proofreading techniques.

Module 4:

Preparing Technical Presentations
Planning a presentation, structuring content, use of slides, visual aids, and multimedia tools (e.g., PowerPoint, Prezi). Non-verbal communication and handling audience questions.

Module 5:

Practice and Evaluation
Student individual and group presentations on technical topics, peer reviews, instructor feedback. Writing mini technical reports based on lab/project work or case studies.

Labs / Practicals:

1. Analyzing Samples of Technical Documents
Objective: Identify the structure and features of different technical documents (e.g., lab report, project report, instruction manual).
2. Writing a Lab Report Based on Practical Work
Objective: Write a complete lab report including abstract, objectives, procedure, observations, and conclusion.
3. Creating a Project Proposal Document
Objective: Prepare a short technical proposal on a selected topic including problem statement, objectives, methodology, and timeline.
4. Designing Visual Elements (Graphs, Charts, Tables)
Objective: Use data to create visual aids using Excel/PowerPoint and integrate them into a technical report.
5. Grammar and Proofreading Workshop
Objective: Identify and correct errors in sample texts focusing on sentence construction, punctuation, and vocabulary.
6. Preparing and Delivering an Individual Presentation
Objective: Create and present a 5-minute talk on a technical topic using PowerPoint or Prezi.

7. Group Presentation on a Technical Case Study
Objective: Collaboratively research and present findings on a real-world technical issue or innovation.
8. Peer Review and Feedback Session
Objective: Evaluate and provide constructive feedback on peers' written reports and oral presentations.
9. Editing and Improving a Poorly Written Report
Objective: Revise and rewrite a given flawed report to meet academic/professional standards.
10. Writing a Mini Technical Report on a Chosen Topic
Objective: Submit a short report (3–5 pages) on a lab experiment, research article, or emerging technology.

References:

1. Meenakshi Raman and Sangeeta Sharma – Technical Communication: Principles and Practice, Oxford University Press.
2. M. Ashraf Rizvi – Effective Technical Communication, Tata McGraw-Hill.
3. Sharma, R.C. and Krishna Mohan – Business Correspondence and Report Writing, Tata McGraw-Hill.
4. Lesikar, Raymond V. et al. – Basic Business Communication, Tata McGraw-Hill.
5. David F. Beer and David McMurrey – A Guide to Writing as an Engineer, Wiley India.

Course Code	Course Name	L	T	P	Cr
DOAI250012	Business Intelligence	2	0	4	4

Course Outcomes:

CO1:Understand BI Fundamentals: Explain the purpose, structure, and role of Business Intelligence systems in data-driven decision-making, while distinguishing BI from data analytics.

CO2:Apply BI Tools and Techniques: Perform data analysis using spreadsheet software, manage data with MySQL, and execute SQL queries to support BI processes.

CO3:Create Visualizations with Power BI: Build and customize interactive dashboards and reports in Power BI to effectively communicate insights through data visualizations.

CO4:Integrate Machine Learning with BI: Identify how Machine Learning enhances BI through automation, customer segmentation, and pattern discovery, while recognizing associated challenges.
CO5:Implement Predictive Analytics: Apply supervised and unsupervised Machine Learning algorithms, such as linear regression, K-Nearest Neighbors, and Decision Trees, and evaluate model accuracy for BI applications.

Module 1:

Meaning and purpose of BI, Structure of BI systems: data sources, ETL (Extract, Transform, Load), data warehouses, and reporting, Importance of data-driven decision-making through BI, Similarities and differences between BI and data analytics

Module 2:

BI analysis using spreadsheet software (e.g., Microsoft Excel), Using MySQL for data management, Understanding SQL: writing and executing queries (SELECT, JOIN, GROUP BY, etc.)

Module 3:

Introduction to Power BI: features and capabilities, Building interactive dashboards and reports, Data modeling and DAX (Data Analysis Expressions) in Power BI, Enhancing BI capabilities through visualizations

Module 4:

Differences between ML and BI, Applications of ML in BI: automating tasks, customer segmentation, uncovering hidden patterns, handling unstructured data, Challenges of integrating ML into BI (e.g., data quality, complexity, interpretability)

Module 5:

Machine Learning for Predictive Analytics, Supervised vs. unsupervised learning, Linear regression and classification techniques, ML algorithms: K-Nearest Neighbors, Naïve Bayes, Decision Trees, Implementing ML models and evaluating accuracy (e.g., confusion matrix, precision, recall)

Labs / Practicals:

Lab 1: Data Analysis with Excel
Use Microsoft Excel to perform basic BI analysis, including pivot tables, charts, and conditional formatting on a sample sales dataset.

Lab 2: SQL Query Execution in MySQL
Set up a MySQL database, create tables for customer and order data, and write SQL queries (SELECT, JOIN, GROUP BY) to extract insights.

Lab 3: Creating a Power BI Dashboard
Import a dataset into Power BI, create visualizations (bar charts, pie charts), and build an interactive dashboard

to display sales performance.

Lab 4: Data Modeling with DAX in Power BI
Use Power BI to create calculated columns and measures using DAX to analyze profit margins and sales trends in a retail dataset.

Lab 5: Customer Segmentation with Excel
Apply clustering techniques in Excel (e.g., using filters and pivot tables) to segment customers based on purchase history and demographics.

Lab 6: Automating BI Tasks with Power BI
Automate data refresh and report generation in Power BI using a sample dataset to simulate a real-world BI workflow.

Lab 7: Linear Regression for Sales Prediction
Implement a linear regression model in Python (using scikit-learn) to predict future sales based on historical data and evaluate model accuracy.

Lab 8: Classification with Naïve Bayes
Use Python to apply a Naïve Bayes classifier on a customer churn dataset to predict which customers are likely to leave, and assess performance using a confusion matrix.

Lab 9: Decision Trees for Pattern Discovery
Build a decision tree model in Python to identify patterns in a marketing campaign dataset, visualizing the tree and calculating precision and recall.

Lab 10: K-Nearest Neighbors for Customer Segmentation
Implement a K-Nearest Neighbors algorithm in Python to segment customers based on purchasing behavior, and visualize clusters using a scatter plot.

References:

1. Business Intelligence Guidebook: From Data Integration to Analytics by Rick Sherman
2. Business Intelligence: A Managerial Perspective on Analytics by Ramesh Sharda, Dursun, Efraim
3. Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications by Larissa T. Moss and Shaku Atre
4. The State of AI in Business Intelligence (e-book)
5. The Definitive Guide to DAX: Business Intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel by Marco Russo and Alberto Ferrari
6. Introduction to Machine Learning with Python, O'reilly
7. Understanding Machine Learning: From Theory To Algorithms by Shai Shalev-Shwartz
8. Learning MySQL by Hugh E. Williams (O'reilly
9. Learning SQL: Generate, Manipulate, and Retrieve Data by Alan Beaulieu

Course Code	Course Name	L	T	P	Cr
DOAI250062	Applied Ethics for AI	3	1	0	4

Course Outcomes:

CO1: Understand the foundations of AI ethics, common pitfalls, and the role of ethics in the AI lifecycle.
CO2: Apply ethical AI design principles such as fairness, transparency, autonomy, and privacy.
CO3: Analyze and compare ethical frameworks and theories for real-world AI applications.
CO4: Evaluate explainability, interpretability, and data privacy in AI systems.
CO5: Assess post-deployment ethical challenges, conduct AI impact assessments, and design governance frameworks.

Module 1:

Introduction to AI Ethics & Responsible AI:Introduction to AI and AI Ethics, The importance of ethical AI for the greater good, Role of ethics in responsible AI development
Common ethical pitfalls in AI: bias, discrimination, privacy violations, opacity, unintended consequences,The AI project lifecycle and where ethics fits

Module 2:

Principles and Approaches for Ethical AI Design:Principles of Ethical AI:Human-centered AI,Transparency and explainability,Fairness and non-discrimination
Autonomy,Beneficence and non-maleficence,Privacy and data protection,Approaches to ethical AI design:
Rule-based Top-down vs bottom-up approaches Ethics-first development methodology

Module 3:

Ethical Frameworks and Theories in Practice:Introduction to ethical frameworks and their relevance in AI,Virtue-based ethics in AI,Bioethics framework for AI healthcare and social impact domains,Utilitarianism and Kantian ethics,Comparative analysis and practical application of frameworks

Module 4:

Explainability,Interpretability,and Data Privacy:Explainable AI (XAI): concepts, tools, and importance,Interpretability vs explainability: impact on decision-making and trust,Data protection principles in AI,Ethical handling of sensitive data,Privacy risks and consequences of data breaches
Implementing AI systems with privacy by design

Module 5:

Ethical Deployment, Impact Assessment, and Governance:Ethical issues post-deployment of AI solutions,Mitigating unintended consequences,Conducting AI impact assessments,Creating and evaluating a Code of AI Ethics,Institutional and organizational ethics in AI,Critiquing current ethics codes (IEEE, EU AI Act, etc.),Capstone project: creation of an ethics-first AI solution

Labs / Practicals:

Tutorial 1: AI Ethics Case Study Analysis
Objective: Analyze real-world ethical failures in AI (e.g., COMPAS algorithm, facial recognition bias)
Tutorial 2: AI Lifecycle Mapping with Ethics
Objective: Map ethics checkpoints to the stages of the AI development lifecycle
Tool: Draw.io or Lucidchart
Tutorial 3: Evaluating AI for Ethical Principles

Objective: Evaluate an AI use case (e.g., recruitment system or loan approval AI) for compliance with ethical principles
Tool: Google Forms/Excel checklist for peer review
Tutorial 4: Ethics-First AI Prototyping (Design Lab)
Objective: Design a conceptual AI application with ethical principles embedded from the start
Tutorial 5: Ethical Reasoning Role-Play
Objective: Apply ethical theories (utilitarianism, Kantianism, virtue ethics) to resolve AI dilemmas
Tutorial 6: Framework Application Case Study
Objective: Analyze a controversial AI scenario using multiple frameworks
Tutorial 7: Building a Simple Explainable AI Model
Objective: Use a simple machine learning model and explain its predictions
Tool: Python (scikit-learn, SHAP, LIME in Colab)
Tutorial 8: Privacy Risk Assessment Exercise
Objective: Identify potential privacy risks in an AI application (e.g., healthcare chatbot or surveillance AI)
Tool: Custom DPIA form or Google Docs template
Tutorial 9: Creating a Code of AI Ethics
Objective: Draft a customized AI ethics code for a hypothetical organization
Tutorial 10: Capstone Project – Building an Ethics-First AI Solution
Objective: Design a complete ethical AI solution for a social or business problem

References:

1. “Artificial Intelligence: A Guide for Thinking Humans”
Author: Melanie Mitchell
Publisher: Penguin
2. “Ethics of Artificial Intelligence and Robotics” (Stanford Encyclopedia of Philosophy)
Editor: Edward N. Zalta (free online)
3. “Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy”
Author: Cathy O’Neil
Publisher: Crown Publishing
4. “Ethics of Artificial Intelligence”
Authors: Markus D. Dubber, Frank Pasquale, Sunit Das
Publisher: Oxford University Press

Course Code	Course Name	L	T	P	Cr
DCSE250035	Computer Networks	3	0	2	4

Course Outcomes:

CO1: Understand Network Basics – Learn network models, topologies, and transmission media.
CO2: Analyse Data Link Layer – Evaluate error detection, MAC protocols, and wireless communication.
CO3: Implement Network Layer Functions – Apply IP addressing, subnetting, and routing algorithms.
CO4: Apply Transport & Application Layer Concepts – Compare TCP/UDP and analyse key protocols.
CO5: Demonstrate Security & Emerging Technologies – Identify threats, implement cryptography, explore SDN, IoT, and cloud networking.

Module 1:

Introduction to Computer Networks Overview of Computer Networks, Network Types: LAN, WAN, MAN, PAN, Network Topologies: Bus, Star, Ring, Mesh, Hybrid, OSI Model & TCP/IP Model – Layers and Functions, Transmission Media: Wired & Wireless Technologies, Switching Techniques: Circuit, Packet, and Message Switching

Module 2:

Data Link Layer & MAC Protocols
Error Detection and Correction: Parity, Checksum, CRC, Hamming Code, Flow Control Mechanisms: Stop-and-Wait, Sliding Window Protocol, MAC Protocols: ALOHA, CSMA/CD, CSMA/CA, Ethernet Standards (IEEE 802.3), VLANs, Spanning Tree Protocol, Wireless LAN (Wi-Fi), Bluetooth, and Mobile Networks

Module 3:

Network Layer & Routing Algorithms
IPv4 and IPv6 Addressing, Subnetting, and CIDR, Routing Algorithms: Distance Vector, Link-State, and Hybrid Routing, Routing Protocols: RIP, OSPF, EIGRP, and BGP, Congestion Control Mechanisms and Quality of Service (QoS), Network Address Translation (NAT) and DHCP

Module 4:

Transport Layer & Application Layer Protocols
TCP and UDP: Connection Establishment, Flow Control, and Error Control, Congestion Control: TCP Reno, TCP Tahoe, and Fair Queuing, Application Layer Protocols: HTTP, HTTPS, FTP, SMTP, POP3, IMAP, DNS and URL Resolution, Peer-to-Peer Networks, Security in Transport & Application Layers: TLS, SSL, Firewalls

Module 5:

Network Security and Emerging Trends
Cryptography Fundamentals: Symmetric and Asymmetric Encryption, Authentication and Digital Signatures, Network Security Attacks: DoS, DDoS, Phishing, Man-in-the-Middle, Wireless and IoT Security Challenges, Software-Defined Networking (SDN) and Cloud Networking, Introduction to 5G, Edge Computing, and Quantum Networking

Labs / Practicals:

Preparing

References:

1. "Computer Networking: A Top Down Approach (Eighth Edition)" by James F. Kurose and Keith W. Ross

https://gaia.cs.umass.edu/kurose_ross/about.php

2. "An Introduction to Computer Networks (Second Edition)" by Peter Lars Dordal
https://intronetworks.cs.luc.edu/current2/ComputerNetworks.pdf

3. "Computer Networking: Principles, Protocols and Practice (Third Edition)" by Olivier Bonaventure
https://github.com/cnp3/ebook/releases/download/draft-3rd/CNP3-2021.pdf

4. "Computer Networks: A Systems Approach (Sixth Edition)" by Larry L. Peterson and Bruce S. Davie
https://github.com/SystemsApproach/book/releases/download/v6.1/book.pdf

5. "Computer Networks (Fifth Edition)" by Andrew Stuart Tanenbaum and David J. Wetherall
https://www.oreilly.com/library/view/computer-networks-fifth/9780133485936

6. "Internetworking with TCP/IP: Principles, Protocols, and Architecture (Sixth Edition)" by Douglas Earl Comer
https://www.oreilly.com/library/view/internetworking-with-tcpip/9780137464197

7. "UNIX Network Programming: Volume 1 (Third Edition)" by William Richard Stevens, Bill Fenner, and Andrew M. Rudoff
https://www.oreilly.com/library/view/the-sockets-networking/0131411551

8. "Computer and Network Organization: An Introduction" by Maarten van Steen and Henk Sips
https://www.distributed-systems.net/index.php/books/computer-and-network-organization

9. "Beej's Guide to Network Programming" by Brian Hall
https://beej.us/guide/bgnet/pdf/bgnet_a4_c_2.pdf

Course Code	Course Name	L	T	P	Cr
DOEE250038	Capstone Project planning	1	0	2	2

Course Outcomes:

CO1: Understand the importance and structure of capstone projects in solving real-world problems

CO2: Formulate a project problem statement and define clear goals and deliverables

CO3: Conduct effective literature reviews and gather requirements using proper techniques

CO4: Design a project plan including timelines, responsibilities, and risk management

CO5: Prepare project documentation and presentation for reviews and approval

Module 1:

Introduction to Capstone Projects (3 Hours)
What is a capstone project?, Importance of project-based learning in curriculum, Structure and life cycle of a capstone project, Academic and industry expectations, Overview of domains (AI, IoT, Web, Security, Data Science, etc.)

Module 2:

Problem Identification and Scope Definition (3 Hours)
Identifying real-world problems and research gaps, Selecting domain/topic of interest, Defining project goals, objectives, and scope, Formulating problem statements and success criteria, Role of innovation, feasibility, and scalability

Module 3:

Literature Survey and Requirement Analysis (3 Hours)
Techniques for conducting a literature review, Finding, evaluating, and citing scholarly sources, Analyzing existing solutions and identifying gaps, Gathering functional and non-functional requirements, Stakeholder analysis and documenting needs

Module 4:

Project Design and Planning Tools (3 Hours)
Work Breakdown Structure (WBS), Creating Gantt charts and milestone planning, Task allocation and team responsibilities, Project risk assessment and mitigation, Introduction to tools: Trello, Jira, MS Project

Module 5:

Documentation and Review Process (3 Hours)
Preparing project proposal and synopsis, Interim and final review presentation tips, Writing effective technical documentation, Introduction to plagiarism checkers and citation managers, Rubrics for evaluation and assessment criteria

Labs / Practicals:

1. Domain exploration and problem identification

2. Writing a problem statement and objective matrix

3. Conducting a sample literature review using Scopus/Google Scholar

4. Drafting requirement specification document (SRS)

5. Creating Gantt charts and project schedule in MS Excel or online tools

6. Group review presentations for faculty panel

7. Final submission of project proposal and documentation

References:

1. Project Management Body of Knowledge (PMBOK Guide) – PMI

2. Practical Research: Planning and Design by Paul D. Leedy & Jeanne Ormrod

3. How to Write a Thesis by Umberto Eco

4. Technical Report Writing Today by Daniel Riordan

5. IEEE and APA style referencing guides

6. Zotero, Mendeley, Grammarly, Overleaf tools

Course Code	Course Name	L	T	P	Cr
DOAI250041	Natural Language Processing	3	0	2	4

Course Outcomes:

CO1: Understand the history, stages, challenges, and applications of NLP in real-world systems.
CO2: Apply morphological analysis, finite state methods, stemming, and n-gram models for language processing.
CO3: Implement POS tagging, CFG parsing, and sentence-level syntactic analysis using statistical and ML-based methods.
CO4: Analyze word sense disambiguation techniques using machine learning and dictionary-based approaches.
CO5: Apply discourse analysis techniques including reference resolution, pronoun interpretation, and coherence modeling.

Module 1:

History of NLP; Generic NLP system; Levels of NLP; Knowledge in language processing problem; Ambiguity in natural language; Stages in NLP; Challenges of NLP; Role of machine learning; Brief history of the field; Applications of NLP: Machine translation, Question answering system, Information retrieval, Text categorization, text summarization & Sentiment analysis

Module 2:

Morphology analysis survey of English morphology, inflectional morphology & derivational morphology; Regular expressions; Finite automata; Finite state transducers (FST); Morphological parsing with FST; Lexicon free FST, Porter stemmer, N-Grams, N-gram language model, N-gram for spelling correction.

Module 3:

Part-of-Speech tagging (POS); Lexical syntax tag set for English (Penn Treebank); Rule based POS tagging; Stochastic POS tagging; Issues: Multiple tags & words, unknown words, class-based n-grams, HM Model ME, SVM, CRF; Context Free Grammar; Constituency; Context free rules & trees; Sentence level construction; Noun Phrase; Coordination; Agreement; Verb phrase & sub categorization

Module 4:

Attachment for fragment of English sentences, noun phrases, verb phrases, prepositional phrases; Relations among lexemes & their senses; Homonymy, Polysemy based disambiguation & limitations, Robust WSD; Machine learning approach and dictionary-based approach.

Module 5:

Discourse reference resolution; Reference phenomenon; Syntactic & semantic constraints on co reference; Preferences in pronoun interpretation; Algorithm for pronoun resolution; Text coherence; Discourse structure

Labs / Practicals:

Module 1: Introduction to NLP & Applications
1. Experiment 1: Demonstration of NLP Applications
o Implement basic NLP applications like sentiment analysis, text summarization, question answering, and information retrieval using pre-trained models or APIs.
2. Experiment 2: Text Preprocessing Techniques
o Perform tokenization, stop-word removal, stemming, and lemmatization on sample English texts.

Module 2: Morphology and Language Models
3. Experiment 3: Morphological Analysis using Stemming and Lemmatization
o Analyze inflectional and derivational forms of English words using stemmers and lemmatizers.
4. Experiment 4: Implementation of N-Gram Language Models

o Build unigram, bigram, and trigram models from a text corpus and compute probabilities of word sequences.
5. Experiment 5: Spelling Correction using N-Gram Probabilities
o Implement a simple spelling corrector using edit distance and N-gram based word probability estimation.

Module 3: POS Tagging and Grammar Parsing
6. Experiment 6: POS Tagging using Rule-Based and Statistical Methods
o Perform part-of-speech tagging using both rule-based and statistical approaches; evaluate accuracy on tagged corpora.
7. Experiment 7: Constituency Parsing using Context-Free Grammar (CFG)
o Define a CFG and parse given sentences to generate parse trees showing sentence structure.

Module 4: Word Sense Disambiguation
8. Experiment 8: Word Sense Disambiguation using Dictionary-Based Methods
o Use the Lesk algorithm to identify correct word senses in different sentence contexts.
9. Experiment 9: Word Sense Disambiguation using Supervised Learning
o Implement a supervised machine learning model to classify the correct sense of an ambiguous word using labeled data.

Module 5: Discourse and Reference Resolution
10. Experiment 10: Co-reference Resolution and Discourse Analysis
• Resolve pronouns and noun phrase references in a passage and analyze coherence and discourse structure.

References:

1. James Allen. Natural Language Understanding. The Benajmins/Cummings Publishing Company Inc. 1994. ISBN 0-8053-0334-0.
2. Tom Mitchell. Machine Learning. McGraw Hill, 1997. ISBN 0070428077.
3. Cover, T. M. and J. A. Thomas: Elements of Information Theory. Wiley. 1991. ISBN 0-471-06259-6.

Course Code	Course Name	L	T	P	Cr
DCSA250020	Computer Vision	3	0	2	4

Course Outcomes:

CO1: Explain the fundamentals of digital image formation, transformations, and low-level image processing techniques such as filtering, enhancement, and restoration.
CO2: Apply various feature extraction and representation methods (edges, corners, histograms, texture, SIFT, SURF, etc.) for image analysis and recognition.
CO3: Analyze depth estimation, multi-view geometry, and segmentation techniques to perform object detection and 3D reconstruction.
CO4: Evaluate motion analysis approaches including optical flow, spatio-temporal modeling, and background subtraction for dynamic scene interpretation.
CO5: Design computer vision solutions by integrating shape-from-X methods, reflectance models, and photometric techniques for real-world applications.

Module 1:

Digital Image Formation and low-level processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing, introduction to computer vision

Module 2:

Feature Extraction: Shape, histogram, color, spectral, texture, Feature analysis, feature vectors, distance /similarity measures, data preprocessing, Edges - Canny, LOG, DOG; Scale-Space Analysis-Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT; Line detectors (Hough Transform), Orientation Histogram, SIFT, SURF, GLOH, Corners - Harris and Hessian Affine.

Module 3:

Depth estimation and Multi-camera views: Perspective, Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Binocular Stereopsis: Camera and Epipolar Geometry; Auto-calibration.
Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Module 4:

Motion Analysis: Optical Flow, KLT, Spatio-Temporal Analysis, Background Subtraction and Modeling, Dynamic Stereo; Motion parameter estimation.

Module 5:

Shape from X: Light at Surfaces; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges Albedo estimation; Photometric Stereo; Phong Model; Reflectance Map

Labs / Practicals:

Preparing

References:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.
3. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition,

Cambridge University Press, March 2004.
4. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.
5. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.

Course Code	Course Name	L	T	P	Cr
DOAI250053	Attention Based Architectures	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of attention mechanisms and their evolution in neural networks.

CO2: Analyze different types of attention, including soft, hard, self, and multi-head attention.

CO3: Explore architectures like Transformers and BERT that rely heavily on attention.

CO4: Implement and train models using attention for tasks such as translation, summarization, and classification.

CO5: Evaluate attention-based models using benchmarks and performance metrics.

CO6: Investigate recent advancements and trends in attention research, including cross-modal and sparse attention.

Module 1:

Introduction to Attention Mechanisms (9 Hours)
Motivation behind attention in deep learning, From RNNs to attention-based models, Types of attention: global vs local, soft vs hard, Attention score functions and alignment, Applications of basic attention in NLP and vision

Module 2:

Self-Attention and the Transformer Model (9 Hours)
Self-attention mechanism and scaled dot-product attention, Positional encoding and sequence modeling, Architecture of the Transformer (Encoder, Decoder), Multi-head attention and layer normalization, Comparison with RNN and CNN models

Module 3:

Pre-trained Language Models (9 Hours)
Transfer learning and fine-tuning in NLP, BERT architecture and training objectives (MLM, NSP), Variants of BERT: RoBERTa, ALBERT, DistilBERT, GPT series and causal attention, Use cases in question answering, classification, and summarization

Module 4:

Vision and Cross-Modal Transformers (9 Hours)
Vision Transformer (ViT) and attention in image processing, Cross-attention in multi-modal systems, CLIP and ALIGN for vision-language tasks, Audio and video processing with attention models, Transformer models in reinforcement learning

Module 5:

Advanced Topics and Future Directions (9 Hours)
Sparse and efficient attention mechanisms (Longformer, Linformer), Attention in graph neural networks, Interpretability and explainability of attention, Challenges with scaling and training Transformers, Research directions and ethical considerations

Labs / Practicals:

1. Implementing basic attention from scratch in PyTorch or TensorFlow

2. Building a Transformer encoder for text classification

3. Fine-tuning BERT for named entity recognition

4. Working with Hugging Face Transformers library

5. Implementing image classification using Vision Transformers

6. Multi-modal learning using cross-attention models

7. Mini-project on applying attention to a real-world NLP or vision dataset

References:

1. Attention Is All You Need, Vaswani et al. (2017)

2. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin et al.

3. The Illustrated Transformer by Jay Alammar

4. Transformers for Natural Language Processing by Denis Rothman

5. Hugging Face documentation and TensorFlow Tutorials

6. Research papers and arXiv preprints on attention and Transformer variants

Course Code	Course Name	L	T	P	Cr
DOEE250037	Capstone Project execution and Report writing	1	0	2	2

Course Outcomes:

CO1: Identify and define real-world problems relevant to the domain of study.
CO2: Plan and execute a project using systematic methodologies and tools.
CO3: Apply domain-specific knowledge and technical skills to develop a working prototype or solution.
CO4: Analyze project outcomes using qualitative or quantitative evaluation methods.
CO5: Prepare professional project documentation and technical reports.
CO6: Present the project work clearly and confidently in oral and written formats.

Module 1:

Capstone Project Overview and Planning (6 Hours)
Introduction to capstone projects, Selection of project domain and problem statement formulation, Setting goals, defining scope, and identifying deliverables, Team formation and project proposal preparation, Timeline and milestone planning using project management tools

Module 2:

Literature Review and Requirement Gathering (6 Hours)
Techniques for conducting effective literature surveys, Identifying knowledge gaps and research opportunities, Understanding functional and non-functional requirements, Stakeholder analysis and requirement documentation, Tools for requirement tracking

Module 3:

Project Design and Implementation (6 Hours)
Architectural design and technology stack selection, Data collection and preprocessing if applicable, System modeling and design diagrams such as DFD, ERD, UML, Implementation of proposed system, iterative testing, Version control practices using Git or similar tools

Module 4:

Project Testing, Evaluation, and Validation (6 Hours)
Unit testing, integration testing, and user acceptance testing, Performance metrics and result validation, Benchmarking against existing solutions, Handling feedback and project improvement, Final demonstration preparation

Module 5:

Report Writing and Presentation Skills (6 Hours)
Structure of technical project reports, Abstract, introduction, methodology, results, conclusion, Referencing and citation styles, Formatting, grammar, and plagiarism checking tools, Preparing effective presentations and viva voice tips

Labs / Practicals:

1. Drafting a capstone project proposal and review
2. Conducting a detailed literature review using tools like Google Scholar or Scopus
3. Requirement analysis and documentation
4. Designing project architecture and implementation using suitable technologies
5. Maintaining a project logbook and Gantt chart

6. Preparing interim progress reports and getting feedback
7. Creating final technical report and PowerPoint presentation

References:

1. How to Write a Thesis by Umberto Eco
2. Project Management Body of Knowledge PMBOK Guide by PMI
3. Practical Research Planning and Design by Paul D Leedy and Jeanne Ellis Ormrod
4. Technical Report Writing Today by Daniel G Riordan
5. IEEE and APA referencing style manuals
6. Tools like Zotero, Mendeley, Grammarly, and LaTeX

Course Code	Course Name	L	T	P	Cr
DASH250091	Professional Values and Ethics	2	0	0	2

Course Outcomes:

CO1: Describe profession, professionalism, its challenges, effective communication, and the role of regulatory bodies and professional organizations related to engineering.
CO2: Analyze professional values such as integrity, responsibility, respect, fairness, and accountability, and demonstrate their application in professional practice.
CO3: Apply knowledge of ethics and bioethics in ethical decision-making in collaboration with professional teams.
CO4: Evaluate responsibilities related to safety, risk management, and intellectual property in professional practice.
CO5: Examine global ethical issues and professional responsibilities in technological and research contexts.

Module 1:

Values
Definition and characteristics of values, value clarification, personal and professional values, professional socialization: integration of professional and personal values, importance of professional values in engineering and technology, core values – integrity, honesty, trust, respect, responsibility, fairness, accountability, ethical conduct in workplace relationships and decision-making.

Module 2:

Engineering Ethics
Senses of engineering ethics, variety of moral issues, types of inquiry, moral dilemmas, moral autonomy, Kohlberg's theory, Gilligan's theory, consensus and controversy, models of professional roles, theories about right action – self-interest, customs and religion, application of ethical theories, valuing time, cooperation, commitment, case studies.

Module 3:

Engineering as Social Experimentation
Engineering as social experimentation, framing the problem, determining the facts, codes of ethics, clarifying concepts, application issues, common ground, general principles – utilitarian thinking and respect for persons, case studies.

Module 4:

Engineers Responsibility for Safety and Risk
Safety and risk, assessment of safety and risk, risk-benefit analysis, reducing risk, safety and the engineer – designing for safety, Intellectual Property Rights (IPR).

Module 5:

Global issues
Globalization, cross-cultural issues, environmental ethics, computer ethics, computers as instruments of unethical behavior, computers as objects of unethical acts, autonomous systems, computer codes of ethics, weapons development, ethics and research, analyzing ethical problems in research, case studies.

Labs / Practicals:

N/A

References:

1. M. Govindarajan, S. Natarajan, V.S. Senthil Kumar – Engineering Ethics Includes Human Values, PHI Learning Pvt. Ltd.
2. Charles E. Harris, Michael S. Pritchard, Michael J. Rabins – Engineering Ethics, Cengage Learning.
3. Mike W. Martin, Roland Schinzinger – Ethics in Engineering, Tata McGraw Hill.

4. A.R. Aryasri, Dharanikota Suyodhana – Professional Ethics and Morals, Maruthi Publications.
5. A. Alavudeen, R. Kalil Rahman, M. Jayakumaran – Professional Ethics and Human Values, Laxmi Publications.
6. D.R. Kiran – Professional Ethics and Human Values.
7. P.S.R. Murthy – Indian Culture, Values and Professional Ethics, BS Publications

Course Code	Course Name	L	T	P	Cr
DCSE250004	Advanced Data Structures and Algorithms	3	0	2	4

Course Outcomes:

CO1: Analyze the performance of basic data structures and sorting/searching techniques.
CO2: Implement and apply advanced trees, heaps, and hash tables to solve complex problems.
CO3: Solve problems using graph algorithms and greedy strategies.
CO4: Design algorithms using divide and conquer, backtracking, and dynamic programming.
CO5: Develop efficient and scalable solutions using appropriate algorithmic techniques.

Module 1:

Review of Fundamental Concepts
Pointers and Dynamic Memory Allocation, Arrays, Dynamically Allocated Arrays, Structures, Algorithm Specification, Data Abstraction, Performance Analysis, Performance Measurement, Time and space complexity

Module 2:

Basic Data Structures and Advanced Trees
Stacks, Queues, Linked Lists, Hash Tables, Searching and Sorting Techniques: Linear Search, Bubble Sort, Selection Sort, Insertion Sort, Binary Trees, Binary Tree Traversals, Applications of Binary Trees, Threaded Binary Trees, Binary Search Trees, Advanced Trees: AVL Trees: Rotations, insertion and deletion, Red-Black Trees, Splay Trees, B Trees, B+ Trees, Binary Heap: Min/Max heap operations, Priority Queue applications.

Module 3:

Divide and Conquer, Backtracking, String Matching algorithms
Merge Sort, Quick Sort, Binary Search, Backtracking: N-Queens, Graph Coloring, String matching algorithms: KMP, Rabin-Karp

Module 4:

Graph Algorithms
Graph Representations: Adjacency list, Adjacency Matrix, BFS, DFS, Shortest Path: Dijkstra’s, Greedy algorithms: Activity Selection, Huffman Coding Minimum Spanning Tree: Prim’s and Kruskal’s Algorithms

Module 5:

Dynamic Programming
Basic concepts: Overlapping subproblems, optimal substructure, 0/1 Knapsack, Longest Common Subsequence, Matrix Chain Multiplication

Labs / Practicals:

Preparing

References:

1. "Introduction to Algorithms (Fourth Edition)" by Thomas H. Cormen, Charles Eric Leiserson, Ronald Linn Rivest, Clifford Seth Stein
https://mitpress.mit.edu/9780262046305/introduction-to-algorithms

2. "A Second Course in Algorithms" by Timothy Avelin Roughgarden
https://timroughgarden.org/w16/l

3. "Algorithm Design" by Jon Michael Kleinberg and Eva Tardos
https://dl.acm.org/doi/10.5555/1051910

4. "Computational Intractability: A Guide to Algorithmic Lower Bounds" by Erik D. Demaine, William Ian Gasarch, and Mohammad Taghi Hajiaghayi
https://hardness.mit.edu/drafts/2025-02-02.pdf

5. "The Design and Analysis of Algorithms" by Dexter Campbell Kozen
https://www.cs.cornell.edu/kozen/Papers/daa.pdf

6. "The Art of Computer Programming" by Donald Ervin Knuth
https://www-cs-faculty.stanford.edu/~knuth/taocp.html

Course Code	Course Name	L	T	P	Cr
DCSA250071	Software Engineering	3	1	0	4

Course Outcomes:

CO1: Understand Software Engineering Concepts – Explain core principles, ethical responsibilities, software processes, and critical system properties.
CO2: Analyze Software Requirements – Differentiate functional and non-functional requirements and apply requirements engineering processes.
CO3: Apply System Modeling & Project Management – Develop system models and manage software projects, including planning, scheduling, and risk management.
CO4: Design & Develop Software – Implement architectural and object-oriented design, agile methodologies, and software evolution strategies.
CO5: Perform Software Testing & Cost Estimation – Conduct verification, validation, software testing, team management, and cost estimation techniques

Module 1:

Overview - Introduction, FAQs about software engineering, Professional and ethical responsibility; Socio-technical systems - Emergent system properties, Systems engineering; Critical systems and Software processes - Critical systems, A simple safety-critical system, System dependability, Availability and reliability; Software processes - Software process models, Process iteration, Process activities, The Rational Unified Process, Computer-Aided Software Engineering.

Module 2:

Requirements - Software requirements, Functional and non-functional requirements, User requirements, System requirements, Interface specification, The software requirements document; Requirements engineering processes - Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

Module 3:

System models and Project Management - System models, Context models, Behavioural models, Data models, Object models, Structured methods; Project management - Management activities, Project planning, Project scheduling, Risk management

Module 4:

Software Design - Architectural design, Architectural design decisions, System organisation, Modular decomposition styles, Control styles; Object-oriented design - Objects and object classes, An object-oriented design process, Design evolution; Development - Rapid software development, Agile methods, Extreme programming, Rapid application development; Software evolution - Program evolution dynamics, Software maintenance, Evolution processes, Legacy system evolution.

Module 5:

Verification and validation - Verification and validation, Planning verification and validation, Software inspections, Automated static analysis, Verification and formal methods; Software testing - System testing, Component testing, Test case design, Test automation; Management - Managing people, Selecting staff, Motivating people, Managing groups, The People Capability Maturity Model; Software cost estimation, Software productivity, Estimation techniques, Algorithmic cost modelling, Project duration and staffing

Labs / Practicals:

N/A

References:

1.Ian Sommerville: Software Engineering, 8th Edition, Pearson Education, 2007.
2.Roger S. Pressman: Software Engineering - A Practitioner’s Approach, 7th Edition, Tata McGraw Hill, 2007.
3.Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India, 2009.

Course Code	Course Name	L	T	P	Cr
DOAI250027	Generative AI	3	0	2	4

Course Outcomes:

CO1: Understand the scope, applications, and ethical challenges of Generative AI across domains.
CO2: Explain traditional and deep learning-based approaches to language modeling and LLM architectures.
CO3: Analyze transformer models, GPT architectures, and their real-world use cases.
CO4: Apply prompt engineering techniques to design effective prompts for generative models.
CO5: Evaluate ethical implications, address bias, and explore real-world applications and future trends in Generative AI.

Module 1:

- Definition and Scope of Generative AI
- Applications of Generative AI in Various Domains
- Importance of Generative AI in Business, Healthcare, Art, and Science
- Ethical Considerations and Challenges in Generative AI
- Introduction to Language Models and Their Role in AI

Module 2:

- Traditional Approaches to Language Modeling
- Deep Learning-Based Approaches in NLP
- Overview of Popular LLM Architectures: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) Networks, Autoencoders & Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs)

Module 3:

- Introduction to Transformers and Self-Attention Mechanism
- Understanding GPT and Its Significance in AI
- Pre-training and Fine-Tuning Processes in GPT
- Architecture and Working of GPT Models
- Overview of GPT Variants and Their Use Cases (GPT-3, GPT-4, ChatGPT, etc.)

Module 4:

- Understanding the Concept and Significance of Prompt Engineering
- Strategies for Designing Effective Prompts
- Best Practices for Prompt Engineering in Generative AI
- Experimenting with Different Prompting Techniques
- Real-World Examples and Applications of Prompt Engineering

Module 5:

- Ethical Implications of Generative Models
- Addressing Bias and Fairness in AI Systems
- Ensuring Responsible Use and Deployment of Generative Models
- Use Cases in NLP, Content Generation, and Creative Applications
- Case Studies Highlighting Successful Implementations
- Potential Future Applications and Emerging Trends in Generative AI

Labs / Practicals:

Practical 1: Exploring Generative AI Use Cases
In this practical, students will explore the diverse applications of Generative AI across domains such as healthcare, business, art, and science. They will select one domain and research real-world use cases where Generative AI has made significant contributions. Students will summarize their findings in a short report or

presentation, focusing on the problem solved, the type of generative model used, and the impact of the solution.

Practical 2: Write a Python program to implement a basic Recurrent Neural Network (RNN) using TensorFlow or PyTorch. Train the model on a small text dataset and generate simple text sequences to demonstrate how the model learns patterns in the input.

Practical 3: Use a Python-based tool like TensorFlow Hub or PyTorch to load a pre-trained GAN model and generate images from random noise. Display multiple generated images and experiment by modifying the latent input vectors.

Practical 4: Write a Python program to build an Autoencoder model for text reconstruction. Provide the model with noisy or incomplete input sentences and train it to reconstruct the original version, demonstrating its application in denoising.

Practical 5: Write a Python program to implement a Variational Autoencoder (VAE) for generating synthetic text or image data. Visualize how sampling from different parts of the latent space produces different outputs.

Practical 6: Write a Python program to demonstrate the self-attention mechanism used in Transformer models. Simulate a simple attention block and visualize the attention scores across different input tokens.

Practical 7: Using OpenAI GPT Models (GPT-3 / GPT-4 / ChatGPT)
In this practical, students will interact with large language models such as GPT-3 or GPT-4 via OpenAI Playground or API. They will experiment with generating content such as poems, emails, or stories by modifying parameters like temperature, top-p, and prompt structure. The goal is to understand how GPT models behave and how output quality can be influenced by prompt design.

Practical 8: Designing Effective Prompts for LLMs
Students will learn and apply the concepts of prompt engineering by crafting prompts for different tasks such as summarization, translation, storytelling, and question-answering. They will compare the results of various prompt designs and document the impact of changes in wording, context, and formatting on the model's response.

References:

1. Generative Deep Learning by David Foster, 2nd Edition, O'Reilly.
2. Neural Network Methods for Natural Language Processing by Goldberg, Morgan & Claypool Publishers.

Course Code	Course Name	L	T	P	Cr
DCSA250031	Full Stack Web Development	3	0	2	4

Course Outcomes:

CO1: Explain full stack development concepts, client-server model, and use essential development tools effectively.
CO2: Develop interactive web pages using HTML, CSS, JavaScript events, form validation, and dynamic content.
CO3: Build React applications with components, JSX, props, state, interactive forms, and conditional rendering.
CO4: Create backend APIs using Node.js and Express, connect with frontend, and exchange JSON data.
CO5: Integrate frontend and backend features into a functional mini project with data persistence.

Module 1:

What is full stack development? Difference between front-end and back-end, Understanding how websites work (client/server model),Introduction to tools: VS Code, Chrome, Node.js, GitHub.

Module 2:

Simple web page structure using HTML tags, CSS for styling layout and elements JavaScript basics: variables, events, interaction, Handling user actions like clicks and input, Creating and validating simple forms, Dynamic content display using JavaScript.

Module 3:

Introduction to React and benefits, Setting up a React project, Understanding components and JSX, Passing data with props and using useState, Creating interactive forms, /Handling input changes and button clicks, Rendering lists and conditional content.

Module 4:

Introduction to Backend and APIs, What is Node.js? What is a server? Creating a simple Express serve, Sending and receiving JSON. Simple REST API with Express, Create routes: GET, POST, Connecting Backend to Frontend: Calling backend from React using fetch, Displaying backend data in UI, Submitting form data to backend.

Module 5:

Developing a mini project Using the Features: Frontend form (React), Backend data saving (Node), displaying data list.

Labs / Practicals:

preparing

References:

1. "Full Stack Development with React and Node.js",By David Choi
2. "Pro MERN Stack" (2nd Edition),By Vasan Subramanian
3. JavaScript and JQuery: Interactive Front-End Web Development",By Jon Duckett
4. "Learning Full-Stack JavaScript Development",By Eric Bush

Course Code	Course Name	L	T	P	Cr
DASH250099	Social Networking and Mining	3	1	0	4

Course Outcomes:

CO1. Apply graph theory concepts and use tools like Python, NetworkX, and Gephi to model and analyze real-world social networks.
CO2. Compute network metrics, detect communities, predict links, simulate information diffusion, and identify anomalies in complex networks.
CO3. Develop and implement graph representation learning and deep learning models (such as Graph Neural Networks) using frameworks like PyTorch Geometric.
CO4. Critically assess ethical concerns related to privacy, fairness, and transparency in social network mining.
CO5. Demonstrate proficiency in social network analysis through hands-on projects using real-world datasets.

Module 1:

Foundations of Social Network Analysis:
Introduction to Social Networks and their applications,Graph theory basics: nodes, edges, types of graphs,Basic network modeling using Python and NetworkX,Network Measures: Degree, Betweenness, Closeness, Eigenvector Centrality,Clustering Coefficient, Diameter, Density

Module 2:

Structural Patterns and Community Detection:Network Growth Models:Erdos-Rényi, Barabási-Albert, Watts-Strogatz,Link Analysis:PageRank, HITS, Eigenvector centrality, Community Detection – Part I & II:Modularity, Louvain Method,Label Propagation, Spectral Clustering

Module 3:

Predictive and Dynamic Network Analysis: Link Prediction Techniques:Common Neighbors, Jaccard, Adamic-Adar,Supervised Link Prediction,Network Dynamics & Cascades:Independent Cascade Model,Linear Threshold Model,Influence maximization,Anomaly Detection:Types of anomalies: point, structural, temporal,Outlier detection in graphs

Module 4:

Graph Representation and Deep Learning:Introduction to Deep Learning for Graphs,Graph Embeddings:DeepWalk, Node2Vec, Graph Neural Networks (GNNs):Graph Convolutional Networks (GCN):GraphSAGE, GAT,Node Classification & Link Prediction using GNNs

Module 5:

Applications, Ethics, and Case Studies:Real-world case studies:Twitter, Facebook, Citation networks, Political networks,Applications in recommendation, influence analysis, and fraud detection,Ethical issues in social network mining:Privacy, fairness, transparency,Conclusion and future trends

Labs / Practicals:

N/A

References:

1. Social Network Analysis, Tanmoy Chakraborty, Wiley
2. Network Science, Albert-Lazzlo Barabasi
3. Social Network Analysis: Methods and Applications, Stanley Wasserman, Katherine Faus

Course Code	Course Name	L	T	P	Cr
DCSA250003	Mobile Application Development using Android	2	1	2	4

Course Outcomes:

CO1: Understand the architecture, lifecycle, and development environment of Android applications.
CO2: Design and build responsive user interfaces using Android layout and widget components.
CO3: Develop Android applications using activities, intents, and persistent data storage.
CO4: Integrate multimedia, location, and device services such as sensors and GPS in Android apps.
CO5: Deploy Android applications and apply debugging, testing, and publishing techniques.

Module 1:

Introduction to Android and Development Environment
Overview of Mobile App Development Landscape
Introduction to Android OS: History, Features, and Architecture
Android Studio: Installation and Setup
Android Project Structure (Manifest, Java/Kotlin files, Res folder)
Activity Lifecycle
Hello World App
Emulator and Device Testing Basics

Module 2:

User Interface Design and Layout Management
UI Components: TextView, EditText, Button, ImageView, CheckBox, RadioButton, Spinner
Layouts: LinearLayout, RelativeLayout, ConstraintLayout, FrameLayout
Event Handling and Listeners
Toasts, Dialog Boxes, and Snackbars
RecyclerView and Adapters
Menus: Options Menu, Context Menu

Module 3:

Activities, Intents, and Data Persistence
Intents: Explicit and Implicit
Activity Communication (startActivityForResult)
Fragments: Introduction and Usage
Shared Preferences and Internal Storage
SQLite Database: CRUD Operations
Room Persistence Library (Basics)

Module 4:

Advanced Features and Device Integration
Multimedia: Audio, Video Playback
Camera Integration
Accessing Device Sensors: Accelerometer, Gyroscope
Location and Maps: GPS and Google Maps API (Basic)
Background Tasks: AsyncTask, WorkManager (Intro)
Permissions and Security

Module 5:

App Deployment, Debugging and Project Development
Debugging Tools in Android Studio
Unit Testing and UI Testing Basics
App Signing and Versioning
Publishing App to Google Play (Overview)
Mini Project / Capstone Project
Best Practices in Android Development

Labs / Practicals:

1. Create a “Hello World” Android app using Android Studio and test on emulator.
2. Design a login UI screen using different layout managers and widgets.
3. Create an app with multiple activities and pass data using intents.
4. Store user preferences using SharedPreferences and display them.
5. Develop a simple SQLite-based app for student record management.
6. Integrate Google Maps to display current location.
7. Capture image using device camera and display in the app.
8. Create a media player app that plays audio or video from the device.
9. Develop an app with sensor-based interaction, such as shake detection.
10. Final Mini Project: A complete Android app with at least one form of persistent data storage and sensor/location integration.

References:

Kushwaha, D. S., & Misra, R.
Android Application Development: A Beginner's Guide
McGraw Hill Education, 2018.

Mednieks, Z., Dornin, L., Meike, G., & Nakamura, M.
Programming Android
O’Reilly Media, 3rd Edition, 2012.

Donn Felker
Android Application Development for Dummies
Wiley Publishing, 3rd Edition, 2015.

Joseph Annuzzi Jr., Lauren Darcey, and Shane Conder
Android Wireless Application Development (Volume I & II)
Addison-Wesley, 2015.

Wei-Meng Lee
Beginning Android Programming with Android Studio
Wiley, 4th Edition, 2021.

Online Resources

Android Developers Official Documentation
https://developer.android.com

Google Codelabs – Android Development Tutorials
https://developer.android.com/codelabs

Android Jetpack (UI, Room, WorkManager)
https://developer.android.com/jetpack

Kotlin Language Documentation
https://kotlinlang.org/docs/home.html

Udacity Course: Developing Android Apps with Kotlin
https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012

Course Code	Course Name	L	T	P	Cr
DCSA250059	Object-Oriented Programming with Java	2	0	4	4

Course Outcomes:

CO1: Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
CO2: Understand dynamic memory management techniques using pointers, constructors, destructors, etc
CO3: Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
CO4: Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
CO5: Demonstrate the use of various OOPs concepts with the help of program

Module 1:

An Overview of Java
Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings, Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java's Selection Statements, Iteration Statements, Jump Statements.

Module 2:

Object and Class
Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.

Module 3:

Interface and Packages
Defining Interfaces, Extending Interfaces, Implementing Interfaces, Accessing Interface Variables. Java API Packages, Using System Packages, Naming Conventions, Creating Packages, Accessing a Package, Using a Package, Adding a Class to a Package, Hiding Classes, Static Import.

Module 4:

Exception Handling and I/O Exceptions
Handling-Fundamentals, exception types, using try and catch, multiple catch clause, nested try statements – throw, throws and finally, built-in exceptions, creating own exceptions. Input / Output Basics – Streams – Byte streams and Character streams – Reading and Writing Console – Reading and Writing Files

Module 5:

Multithreaded Programming
Creating Threads, Extending the Thread Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the 'Runnable' Interface,

Inter-thread communication. vectors, lists, maps.

Labs / Practicals:

Preparing

References:

1.Java: The Complete Reference, Twelfth Edition Paperback – 23 December 2021 by Herbert Schildt
2.Java Performance: The Definitive Guide: Getting the Most Out of Your Code by Scott Oaks

Course Code	Course Name	L	T	P	Cr
DCSA250068	Soft Computing	3	0	2	4

Course Outcomes:

After completion of course, students would be able to:
CO1: Identify and describe soft computing techniques and their roles in building intelligent machines.
CO2: Apply genetic algorithms to combinatorial optimization problems.
CO3: Explain the basic concepts, components, and paradigms of neural networks, and compare them with other information processing methods.
CO4: Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems.
CO5: Evaluate and compare solutions by various soft computing approaches for a given problem.

Module 1:

Introduction: What is computational intelligence?- Biological basis for neural networks- Biological versus Artificial neural networks- Biological basis for evolutionary computation- Behavioral motivations for fuzzy logic, Myths about computational intelligence- Computational intelligence application areas, Evolutionary computation, computational intelligence-Adoption, Types, self-organization and evolution, Historical views of computational intelligence, Computational intelligence and Soft computing versus Artificial intelligence and Hard computing.

Module 2:

Evolutionary computation concepts and paradigms- History of evolutionary computation & overview, Genetic algorithms, Evolutionary programming &strategies, Genetic programming, Particle swarm optimization, Evolutionary computation implementations-Implementation issues, Genetic algorithm implementation, Particle swarm optimization implementation.

Module 3:

Neural Network Concepts and Paradigms- What Neural Networks are? Why they are useful, Neural network components and terminology- Topologies - Adaptation, Comparing neural networks and other information Processing methods- Stochastic- Kalman filters - Linear and Nonlinear regression - Correlation - Bayes classification -Vector quantization -Radial basis functions -Preprocessing - Post processing.

Module 4:

Fuzzy Systems concepts and paradigms - Fuzzy sets and Fuzzy logic - Approximate reasoning, Developing a fuzzy controller - Fuzzy rule system implementation.

Module 5:

Performance Metrics- General issues- Partitioning the patterns for training, testing, and validation-Cross validation - Fitness and fitness functions - Parametric and nonparametric statistics, Evolutionary algorithm effectiveness metrics, Receiver operating characteristic curves, Computational intelligence tools for explanation facilities, Case Studies for implementation of practical applications in computational intelligence.

Labs / Practicals:

preparing

References:

1. David B. Fogel and Charles J. Robinson; Computational Intelligence: The experts speak. Wiley - Intersciece 2003.
2. Neuro Fuzzy & Soft Computing - J.-S.R.Jang, C.-T.Sun, E.mizutani, Pearson Education
3. Digital Neural Network - S.Y.Kung, Prentice Hall International Inc.
4. Spiking Neural Networks - Wulfram Gerstner, Wenner Kristler, Cambridge University Press.
5. Neural Networks and Fuzzy Systems: Dynamical Systems Application to Machine Intelligence - Bart Kosko,

Prentice Hall, 1992.

Course Code	Course Name	L	T	P	Cr
DCSA250081	Web Development using Python and Django	2	0	4	4

Course Outcomes:

CO1: Develop basic Python programs utilizing variables, control structures, data types, and loops to solve problems.

CO2: Apply advanced Python concepts including data structures, file handling, functions, modules.

CO3: Demonstrate Object-Oriented Programming (OOP) skills by creating classes, methods.

CO4: Design and build basic back-end web applications using the Django framework, leveraging its MVT architecture, URL routing, forms, authentication, and the admin interface.

CO5: Develop and test RESTful APIs using Django REST Framework and Postman, and implement CRUD operations with proper validation and authentication.

Module 1:

Python functions and variables,Tuples, Dictionaries, Reading and Writing text files, loops, Custom functions, Exception handling, Importing files,OOP Concept

Module 2:

Client-Server Architecture, HTTP Methods (GET, POST, PUT, DELETE), Request-Response Cycle in Web Applications, IP Address and Port Numbers, URL Routing and Path Matching, HTTP Status Codes, CSRF

Module 3:

Introduction to Back-End Web Development (Django), HTTP protocol basics, MVC/MVT model, Virtual environment setup, Installing Django, Django's take on MVT (Model, View, Template), DRY programming principles, Core files (models.py, urls.py, views.py)

Module 4:

Working with database, Setting up database connections, use of .env file,Managing users and Django admin.

Module 5:

Advanced Django and REST Framework, Django URL patterns and views, Designing good URL schemes, Generic views, Django forms and validation, Authentication, Advanced form processing techniques, Creating CRUD applications in Django, Django REST Framework basics, Using Postman to test APIs.

Labs / Practicals:

1.Write a Python script that inputs two numbers and displays the sum, difference, product, and quotient using if-else statements.

2.Python Error Handling & Exceptions

3.Implement a program that reads a text file, counts the number of words and lines, and displays the result. Wrap this logic inside a reusable function.

4.Create a dictionary of student names and grades. Write a function to add new students, update grades, and list all students.

5.Define a Car class with attributes like make, model, year and a method to display car details. Instantiate the class and display the information.

6.Implement a base class Shape with an abstract method area() and two subclasses Rectangle and Circle. Compute the areas of shapes.

7.Create a new Django project and app. Implement a simple view that returns “Hello World” and map it using urls.py. Run the server and verify.

8.Crea a Django app to implement "To-Do List" app. Features like add task,view all tasks,delete tasks or any basic CRUD app

9.Design a Django model for a Product with fields like name, price, stock. Register the model in admin.py and add a few products via the Django admin interface.

10.Create a RESTful API for the Product model. Implement list and create operations, and test these endpoints using Postman.

References:

1.Python Crash Course — Eric Matthes
2.Django for Beginners — William S. Vincent
3.Django Web Development with Python — Samarth Shah
4.Django Web Development” — Vijay Joshi
5. The Definitive Guide to Django: Web Development Done Rightby Adrian Holovaty and Jacob Kaplan-Moss
6. Django 4 By Example: Build powerful and reliable Python web applications from scratchby Antonio Mele (Author), Bob Belderbos (Author)
7. Django for Beginners: Build websites with Python and Djangoby William S. Vincent
8. Beginning Django By Daniel Rubio

Course Code	Course Name	L	T	P	Cr
DCSE250020	Cloud Computing	3	0	2	4

Course Outcomes:

CO1: Explain the evolution, architecture, and key characteristics of cloud computing, including service models (IaaS, PaaS, SaaS) and deployment types (public, private, hybrid).

CO2: Describe various types and implementation levels of virtualization, including CPU, memory, storage, and network virtualization in cloud environments.

CO3: Analyze layered cloud architecture and address design challenges related to inter-cloud resource management and platform deployment.

CO4: Apply distributed and parallel programming paradigms such as MapReduce and Hadoop for cloud-based application development and understand various cloud platforms like AWS, OpenStack, and Google App Engine.

CO5: Evaluate cloud security challenges and solutions, including data and application security, VM security, and risk management strategies in cloud environments.

Module 1:

Introduction to Cloud Computing

Evolution of Cloud Computing, System Models for Distributed and Cloud Computing, NIST Cloud Computing Reference Architecture, Features of Cloud Computing, Cloud Services – IaaS, PaaS, SaaS, Cloud service Providers – Public , Private and Hybrid Clouds.

Module 2:

Introduction to component virtualization

Basics of Virtualization, Types of Virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory, I/O Devices, Desktop Virtualization, Server Virtualization, Storage Virtualization, Network Virtualization.

Module 3:

Architectural Design of Compute and Storage Clouds

Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources.

Module 4:

Parallel and Distributed Programming Paradigms

Map Reduce, Twister and Iterative MapReduce, Hadoop Library from Apache, Mapping Applications, Programming Support, Google App Engine, Amazon AWS, Cloud Software Environments - Eucalyptus, Open Nebula, OpenStack.

Module 5:

Security Overview

Cloud Security Challenges, Software-as-a-Service Security, Security Governance, Risk Management, Security Monitoring, Security Architecture Design, Data Security, Application Security, Virtual Machine Security.

Labs / Practicals:

1. Exploring Cloud Deployment and Service Models
2. Simulating Public, Private, and Hybrid Clouds
3. Virtualization of CPU and Memory
4. Network and Storage Virtualization
5. Deployment of a Layered Cloud Architecture
6. Resource Provisioning and Scaling
7. Deploying a Web Application
8. Simulating Security Attacks and Defenses in a Cloud VM
9. Implementing Access Control and Encryption on Cloud Storage

References:

[1] R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming. New Delhi, India: McGraw-Hill Education, 2013.
[2] A. T. Velte, T. J. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach. New Delhi, India: McGraw-Hill Education, 2010.
[3] K. Hwang, G. C. Fox, and J. J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington, MA, USA: Morgan Kaufmann, 2012.
[4] T. Erl, R. Puttini, and Z. Mahmood, Cloud Computing: Concepts, Technology & Architecture. Boston, MA, USA: Pearson Education, 2013.
[5] G. Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA, USA: O'Reilly Media, 2009.

Course Code	Course Name	L	T	P	Cr
DOAI250018	Data Mining	3	0	2	4

Course Outcomes:

CO1: Understand data warehousing concepts and apply frequent pattern, association, and sequential pattern mining techniques.
CO2: Perform classification, prediction, and clustering methods for different types of data.
CO3: Analyze time-series data using periodicity, trend, and similarity-based techniques.
CO4: Apply stream mining methodologies, address class imbalance, and perform graph and social network analysis.
CO5: Implement web mining techniques, distributed data mining, and explore recent trends in the field.

Module 1:

Introduction to Data Warehousing; Data Mining: Mining frequent
patterns,
association and correlations; Sequential Pattern Mining concepts, primitives,
scalable methods;

Module 2:

Unit 2:
Classification and prediction; Cluster Analysis – Types of Data in Cluster Analysis, Partitioning methods, Hierarchical Methods; Transactional Patterns and other
temporal based frequent patterns,

Module 3:

Mining Time series Data, Periodicity Analysis for time related sequence data,
Trend analysis, Similarity search in Time-series analysis;

Module 4:

Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams, Class Imbalance Problem; Graph Mining; Social Network Analysis;

Module 5:

Web Mining, Mining the web page layout structure, mining web link structure,
mining multimedia data on the web, Automatic classification of web documents

and web usage mining; Distributed Data Mining.
Recent trends in Distributed Warehousing and Data Mining, Class Imbalance Problem; Graph Mining; Social Network Analysis

Labs / Practicals:

t-I Build Data Warehouse and Explore WEKA
Page No

03
2
-Perform data preprocessing tasks and Demonstrate
performing association rule mining on data sets
-Demonstrate performing classification on data sets
-Demonstrate performing clustering on data sets
-Demonstrate performing Regression on data sets

References:

1. Jiawei Han and M Kamber, Data Mining Concepts and Techniques,, Second Edition, Elsevier Publication, 2011.

2. Vipin Kumar, Introduction to Data Mining - Pang-Ning Tan, Michael Steinbach, Addison Wesley, 2006.

3. G Dong and J Pei, Sequence Data Mining, Springer, 2007.

Course Code	Course Name	L	T	P	Cr
DOAI250026	Data Analysis Using Spreadsheet	3	0	2	4

Course Outcomes:

CO1: Understand spreadsheet interfaces, data types, entry methods, and formatting techniques.
CO2: Apply formulas, cell referencing, and commonly used functions for data manipulation.
CO3: Perform data filtering, cleaning, and summarization using Pivot Tables and advanced tools.
CO4: Create Pivot Tables, Pivot Charts, and apply best practices for effective data visualization.
CO5: Design dynamic charts, apply conditional formatting, and develop interactive dashboards.

Module 1:

Overview of spreadsheet applications (Excel, Google Sheets, LibreOffice Calc), Interface and basic operations: cells, rows, columns, worksheets Data types and data entry (text, numbers, dates, formulas). Formatting cells and data (number, text, conditional formatting)

Module 2:

Introduction to formulas and cell referencing (relative, absolute) ,Common functions,Text Functions: CONCATENATE, LEFT, RIGHT, MID, LEN, TRIM, Mathematical Functions: SUM, AVERAGE, MIN, MAX, COUNT, ROUND,Logical Functions: IF, AND, OR, NOT
Lookup Functions: VLOOKUP, HLOOKUP, XLOOKUP, INDEX & MATCH

Module 3:

AutoFilter and Advanced Filter, Custom views and data ranges, Removing duplicates and handling missing data, Using slicers for dynamic filtering (in Excel),Pivot Tables and Data Summarization .

Module 4:

Introduction to Pivot Tables, Creating Pivot Tables from raw data, Grouping data in Pivot Tables ,Using Pivot Charts, Dynamic data analysis using Pivot Table filters and slicers, Charts and Graphs for Data Visualization :Creating basic charts: Bar, Line, Pie, Column, Customizing chart elements: Titles, Axis, Legends, Data Labels, Best practices for visualizing data clearly

Module 5:

Combination charts (e.g., bar + line), Dynamic charts using named ranges and tables, Using sparklines for trend analysis, Conditional formatting with charts, Dashboards overview: integrating charts, tables, and filters.

Labs / Practicals:

1. Create a spreadsheet to store student records (Name, Roll No, Marks in 3 subjects, Total, Percentage).
2. Apply basic formatting (font style, size, color, borders, cell merge, wrap text).
3. Use conditional formatting to highlight students scoring below 40%.
4. Sort data based on Total Marks (ascending and descending).
5. Use basic formulas to calculate Total and Percentage.
(Use =SUM(), =AVERAGE(), =IF(), =MAX(), =MIN())
6. Calculate Grade using IF and nested IF statements.
7. Use text functions like LEFT, RIGHT, LEN, CONCATENATE or TEXTJOIN.
8. Date and time functions: Calculate number of days between two dates using DATEDIF() or TODAY().
9. Create a Pivot Table to summarize department-wise average marks.
10. Use filters and slicers to analyze data for specific conditions (e.g., show only students with A grade).
11. Create data validation (dropdown list) for selecting department names.
12. Create bar chart to compare marks of students across subjects.
13. Create pie chart to represent percentage distribution of grades.
14. Use sparkline charts in Google Sheets for visualizing trends in student marks.

15. Apply statistical functions like STDEV(), VAR(), MODE(), MEDIAN() on a dataset.
16. Perform correlation analysis between two data columns using CORREL().

References:

1. Data Analysis Using Excel by Shanthi Narayanan, Cengage Learning India
2. Statistical Data Analysis Using Excel by J.K. Sharma, Vikas Publishing
3. Excel for Beginners and Data Analysis by Dr. Pradeep K. Sinha, BPB Publications

Course Code	Course Name	L	T	P	Cr
DOAI250029	Artificial Intelligence with Python	2	0	4	4

Course Outcomes:

CO1: Define AI concepts, its history, types, and distinguish AI from human intelligence.
CO2: Implement and evaluate machine learning models using Python for supervised and unsupervised tasks.
CO3: Develop basic deep learning models using neural networks and understand CNNs for image classification.
CO4: Apply NLP techniques such as text preprocessing, sentiment analysis, and language modeling.
CO5: Demonstrate fundamental computer vision tasks using OpenCV for image processing and analysis.
CO6: Able to design and implement various machine learning algorithms in a range of real-world applications.

Module 1:

Introduction to Artificial Intelligence
•Definition and Goals of AI
•History and Applications of AI
•Types of AI (Narrow, General, Super)
•Intelligent Agents and Environments
•AI vs Human Intelligence
•Python for AI: Libraries Overview (NumPy, Pandas, Matplotlib, Scikit-learn)

Module 2:

Machine Learning
•Overview of Machine Learning and Types (Supervised, Unsupervised, Reinforcement)
•Data Preprocessing and Visualization
•Supervised Learning Algorithms: Linear Regression, k-NN, Decision Trees
•Unsupervised Learning: k-Means Clustering
•Model Evaluation: Accuracy, Confusion Matrix, Cross-validation

Module 3:

Introduction to Deep Learning
•Introduction to Neural Networks (Perceptrons, FeedForward Networks, Gradient Descent).
•Basic Concepts of Deep Learning (Layers, Activation Function)
•Introduction to Convolutional Neural Networks (CNNs) for Image classification.
•Simple Neural Network Implementation in Python

Module 4:

Natural Language Processing and its Applications
•Introduction to NLP
•Text Preprocessing (Tokenization, Stop Words, Lemmatization)
•Bag of Words and TF-IDF
•Sentiment Analysis using Python
•Real-world AI Applications: Chatbots, Image Recognition, Recommendation Systems

Module 5:

Computer Vision and its Applications
•Introduction to Computer Vision
•Image Processing Basics (Image Presentation, Pixels, Color Models).
•Image Filtering (Blurring, Sharpening, Edge Detection (Sobel, Canny)
•OpenCV Library for Computer Vision in Python

Labs / Practicals:

List of 10 Practical Exercises

1.Using a simple Linear Regression Model (Scikit-Learn) predict house prices based on a single feature (eg. Square Footage) using a synthetic or small real-world dataset.

2.Using KNN Model (Scikit-Learn) classify the iris dataset. Evaluate its accuracy using a train-test, split and visualize the decision boundaries (if possible, for 2 features).

3.Apply the K-Means clustering algorithm to a given dataset (eg Customer segmentation data). Determine an appropriate number of cluster using the elbow method.

4.Build a simple Feed-Forward neural network (using Tensorflow or Keras) with atleast one hidden layer to classify the MNIST handwritten digit dataset. Train the model and report its accuracy and loss.

5.Given a collection of raw text document. Perform tokenization, lowercasing and stop word removed. Implement stemming or lemmatization. Calculate word frequencies and display the top 10 most frequent word (use NLTK Library).

6.Develop a basic sentiment analysis model using NLTK and Scikit-Learn. Train on a small dataset.

7.Load an image and perform

a. Convert to Gray Scale.

b. Apply a Gaussian blur filter.

c. Detect edges using the Sobel and Canny edge detector.

d. Display the original and processed image.

8.Using a simple CNN model to perform an image classification on a dataset.

9.Using Decision Tree Model, classify the MNIST digit dataset. Evaluate the model.

10.Using logistic regression model, classify the iris dataset and evaluate the model.

References:

1."Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig

2."Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili

3."Deep Learning with Python" by François Chollet

4.Artificial Intelligence, Author: Saroj Kaushik
Publisher: Cengage Learning India

5.Programming in Python, Author: T. R. Padmanabhan
Publisher: Universities Press

6.Machine Learning, Author: V.K. Jain
Publisher: Khanna Book Publishing

7.Artificial Intelligence with Python, Author: Prateek Joshi (Indian-origin, Packt Publishing)

8.Introduction to Machine Learning, Author: Alok Sharma
Publisher: Katson Publishing House

9."Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper (NLTK Book).

10."Learning OpenCV: Computer Vision with the OpenCV Library" by Gary Bradski and Adrian Kaehler

Course Code	Course Name	L	T	P	Cr
DOAI250043	Pattern Recognition	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of pattern recognition, probability, discriminant functions, and supervised learning methods.
CO2: Apply clustering techniques for unsupervised learning and validate cluster quality.
CO3: Perform feature extraction and selection using statistical, transform-based, and dimensionality reduction methods.
CO4: Implement classification techniques using HMMs, SVMs, and feature selection methods.
CO5: Apply fuzzy logic and genetic algorithms for pattern classification in real-world case studies.

Module 1:

Overview of Pattern recognition – Basics of Probability and Statistics, Linear Algebra, Linear Transformations, Components of Pattern Recognition System, Learning and adaptation Discriminant functions – Supervised learning – Parametric estimation – Maximum Likelihood Estimation – Bayesian parameter Estimation – Problems with Bayes approach– Pattern classification by distance functions – Minimum distance pattern classifier.

Module 2:

Clustering for unsupervised learning and classification–Clustering concept – C Means algorithm – Hierarchical clustering – Graph theoretic approach to pattern Clustering – Validity of Clusters.

Module 3:

Feature Extraction and Feature Selection: Feature extraction – discrete cosine and sine transform, Discrete Fourier transform, Principal Component analysis, Kernel Principal Component Analysis. Feature selection – class separability measures, Feature Selection Algorithms - Branch and bound algorithm, sequential forward / backward selection algorithms. Principle component analysis, Independent component analysis, Linear discriminant analysis, Feature selection through functional approximation – Elements of formal grammars.

Module 4:

State Machines – Hidden Markov Models – Training – Classification – Support vector Machine – Feature Selection.

Module 5:

Fuzzy logic – Fuzzy Pattern Classifiers – Pattern Classification using Genetic Algorithms – Case Study Using Fuzzy Pattern Classifiers and Perception

Labs / Practicals:

1. Implementation of Minimum Distance Classifier
o Classify patterns using Euclidean distance-based decision rules.
2. Supervised Learning using Bayesian Classifier with Gaussian Assumptions
o Implement Bayesian classification with maximum likelihood parameter estimation.
3. Unsupervised Learning using C-Means Clustering Algorithm
o Perform clustering on multivariate data using the C-Means algorithm.
4. Hierarchical Clustering and Dendrogram Visualization
o Apply hierarchical clustering and plot dendrograms for visual analysis.
5. Feature Extraction using PCA and Kernel PCA
o Extract features and reduce dimensionality using PCA and Kernel PCA.
6. Sequential Forward Feature Selection Algorithm
o Implement and evaluate sequential forward feature selection technique.
7. Design and Training of Hidden Markov Model (HMM)
o Train an HMM and use it for classification of sequential data.

8. Support Vector Machine Classification using scikit-learn
o Train an SVM with linear and nonlinear kernels for binary/multi-class classification.
9. Pattern Classification using Fuzzy Inference Systems
o Implement a fuzzy rule-based classifier using membership functions.
10. Feature Selection using Genetic Algorithms
• Apply genetic algorithm for optimal feature subset selection in classification problems.

References:

1. Andrew Webb, “Statistical Pattern Recognition”, Arnold publishers, London,1999.
2. C.M.Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006.
3. M. Narasimha Murthy and V. Susheela Devi, “Pattern Recognition”, Springer 2011.
4. Menahem Friedman, Abraham Kandel, “Introduction to Pattern Recognition Statistical, Structural, Neural and Fuzzy Logic Approaches”, World Scientific publishing Co. Ltd, 2000.
5. Robe

Course Code	Course Name	L	T	P	Cr
DOAI250046	R Programming	2	0	4	4

Course Outcomes:

CO1. Understand the fundamentals of R programming and its environment.
CO2. Perform data manipulation and cleaning using R packages.
CO3. Visualize data effectively using R's graphical capabilities.
CO4. Apply statistical methods and data analysis techniques using R.
CO5. Implement real-world data science projects using R.

Module 1:

Introduction to R and Programming Basics
Overview of R, Installing and using packages
Basic syntax, variables, and data types
Operators, expressions, and control structures (if, else, loops)
Functions and user-defined functions

Module 2:

Working with Data Structures
Vectors, lists, matrices, arrays, and data frames
Factors and categorical data
Indexing and subsetting
Reading and writing data (CSV, Excel, JSON)
Data manipulation using dplyr, tidyr

Module 3:

Data Visualization with R
Base R plotting system
Advanced visualization with ggplot2
Customizing plots: labels, themes, and aesthetics
Creating bar charts, histograms, boxplots, scatter plots, etc.
Saving and exporting visualizations

Module 4:

Statistical Analysis
Descriptive statistics and probability distributions
Inferential statistics (t-tests, chi-square, ANOVA)
Linear regression and correlation
Handling missing values and outliers
Introduction to time series and clustering

Module 5:

Report and Interface Generation
Case studies: e.g., sales analysis, healthcare data analysis, social media trends
End-to-end data analysis workflow
Building a reproducible report using R Markdown
Introduction to Shiny for interactive dashboards

Labs / Practicals:

1. Create and assign variables of different data types: numeric, character, logical, complex.
2. Use arithmetic, relational, and logical operators.
3. Write conditional statements (if, if-else, switch) to classify numbers.
4. Write a function to compute factorial of a number using loops.
5. Generate a multiplication table using for loop.
6. Create and manipulate vectors, matrices, lists, arrays, and data frames.
7. Convert a vector into a factor and perform operations.
8. Index and subset rows/columns from a matrix and data frame.
9. Read a CSV file (e.g., students.csv) into a data frame and perform summary statistics.
10. Clean data: handle missing values using na.omit(), is.na() and replace().
11. Use dplyr functions like filter(), select(), mutate(), arrange() and summarise().
12. Plot line graphs, histograms, and scatter plots using base R.
13. Load mtcars dataset. Create a histogram of mpg and a barplot of cyl.
14. Use ggplot2 to:

a. Create a scatter plot of wt vs mpg.
b. Add color coding for different gear values.
c. Create a boxplot of mpg grouped by cyl.

15. Customize charts: axis labels, titles, legends, and themes.
16. Save your plots using ggsave().
17. Compute mean, median, mode, variance, and standard deviation for a dataset.
18. Perform a one-sample and two-sample t-test on simulated data.
19. Perform ANOVA on a built-in dataset (PlantGrowth).
20. Conduct linear regression: mpg ~ wt + hp on mtcars dataset.
21. Visualize the regression line and interpret the model summary.
22. Perform clustering (K-means) on iris dataset.
23. Choose a dataset (e.g., airquality, diamonds, or external CSV).
24. Perform the following steps:

a. Data import, cleaning, and transformation.
b. Exploratory data analysis and visualizations.
c. Statistical or predictive modeling.

25. Generate a report using R Markdown:

a. Include code, plots, and interpretations.
b. Export to HTML/PDF format.

26. Create a basic Shiny app to visualize your dataset interactively.

References:

1. R for Data Science by Hadley Wickham & Garrett Grolemund
2. The Art of R Programming by Norman Matloff
3. Hands-On Programming with R by Garrett Grolemund
4. Practical Statistics for Data Scientists by Peter Bruce & Andrew Bruce
5. CRAN Documentation and Vignettes (https://cran.r-project.org/)

Course Code	Course Name	L	T	P	Cr
DOAI250047	Recommender Systems	3	1	0	4

Course Outcomes:

CO1: Understand the fundamental concepts, types, and applications of recommender systems.
CO2: Analyze and differentiate between various recommendation approaches, including content-based, collaborative filtering, and hybrid methods.
CO3: Evaluate the performance and effectiveness of recommender systems using appropriate metrics and evaluation techniques.
CO4: Design and implement basic recommender system models for real-world datasets and application domains.

CO5: Apply ethical and practical considerations in the deployment and personalization aspects of recommender systems.

Module 1:

Foundations of Recommender Systems

Linear Algebra Notation: Matrix addition, multiplication, transposition, and inverses; covariance matrices, Introduction to Recommender Systems, Functions and applications, Challenges and limitations, Types of Recommender Systems, Content-based, collaborative, knowledge-based, and hybrid

Module 2:

Content-Based Filtering

High-Level Architecture of Content-Based Systems, Item representation techniques, User profile learning methods, Similarity computation (cosine, Jaccard), Advantages and drawbacks of content-based filtering, Case study: Book/movie recommender using TF-IDF

Module 3:

Collaborative Filtering Techniques

User-based nearest neighbor recommendation, Item-based nearest neighbor recommendation, Model-based collaborative filtering: Matrix Factorization (SVD, NMF), Preprocessing-based techniques, Attacks on collaborative filtering: types and countermeasures, Distributed and privacy-aware recommendation

Module 4:

Knowledge-Based and Hybrid Recommender Systems

Knowledge-Based Recommendation, Constraint-based and case-based reasoning, Rule-based recommendation systems, Hybrid Recommender Systems, Need and motivations for hybridization, Hybridization strategies: Monolithic, Parallel, Pipelined, Real-world hybrid systems (e.g., Netflix, Amazon)

Module 5:

Evaluation and Emerging Trends

Evaluation of Recommender Systems, Accuracy metrics: Precision, Recall, F1-score, MAE, RMSE, Evaluation methodologies: offline, online, user studies, Community and Social-Based Recommendations, Social tagging and

folksonomies, Case study: BibSonomy and HeyStaks, Trust-based Recommender Systems, Trust modeling and integration in recommendation, Computational trust and robustness, Future Trends: Explainability, fairness, and deep learning in recommendations

Labs / Practicals:

Tutorial Topics (Problem Solving / Case Discussions):

1. Matrix Algebra for Recommender Systems.

2. Cosine & Jaccard Similarity Problem Solving.

3. User Profile and Item Profile Creation.

4. Comparative Case Studies: Content-Based vs Collaborative Filtering.

5. Hands-on Matrix Factorization Exercises (SVD, NMF).

6. Evaluation Metrics: Problems on Precision, Recall, F1, MAE, RMSE.

7. Diagrammatic Explanation of Hybrid Systems.

8. Emerging Issues: Explainability, Fairness, and Ethics Discussions.

References:

1. Jannach D., Zanker M. and FelFering A., Recommender Systems: An Introduction, Cambridge University Press (2011), 1st ed
https://www.researchgate.net/publication/235910467_Recommender_Systems

2. Ricci F., Rokach L., Shapira D., Kantor B.P., Recommender Systems Handbook, Springer (2011), 1st ed.
https://www.researchgate.net/publication/227268858_Recommender_Systems_Handbook

3. Erwin Kreyzig, Advanced Engineering Mathematics, 10th Edition, Wiley, 2011.
http://www.uop.edu.pk/ocontents/AdvancedEngineeringMathematics.pdf

4. Aggarwal, C. C. Recommender Systems: The Textbook. Springer 2016
https://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal_c_c_recommender_systems_the_textbook.pdf

Course Code	Course Name	L	T	P	Cr
DOAI250048	Reinforcement Learning	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of reinforcement learning, its elements, scope, and historical development.
CO2: Apply action-value methods and strategies for solving multi-armed bandit problems.
CO3: Model real-world tasks as Markov Decision Processes and derive value-based solutions.
CO4: Implement Monte Carlo methods for prediction and control, including off-policy learning techniques.
CO5: Analyze and evaluate reinforcement learning applications through case studies in games, control, and scheduling.

Module 1:

The Reinforcement Learning Problem: Reinforcement Learning, Examples, Elements of Reinforcement Learning, Limitations and Scope, An Extended Example: Tic-Tac-Toe, Summary, History of Reinforcement Learning.

Module 2:

Multi-arm Bandits: An n-Armed Bandit Problem, Action-Value Methods, Incremental Implementation, Tracking a Nonstationary Problem, Optimistic Initial Values, Upper- Confidence-Bound Action Selection, Gradient Bandits, Associative Search (Contextual Bandits)

Module 3:

Finite Markov Decision Processes: The Agent–Environment Interface, Goals and Rewards, Returns, Unified Notation for Episodic and Continuing Tasks, The Markov Property, Markov Decision Processes, Value Functions, Optimal Value Functions, Optimality and Approximation.

Module 4:

Monte Carlo Methods: Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation, Off-Policy Monte Carlo Control, Importance Sampling on Truncated Returns

Module 5:

Applications and Case Studies: TD-Gammon, Samuel’s Checkers Player, TheAcrobot, Elevator Dispatching, Dynamic Channel Allocation, Job-Shop Scheduling.

Labs / Practicals:

Practical 1: Write a Python program to implement a basic Grid World environment, where an agent can move up, down, left, or right to reach a goal. The environment should respond with the new state and a reward after each action.

Practical 2: Write a Python program to simulate the n-Armed Bandit problem with random rewards. Implement a random action-selection strategy and calculate the average reward over multiple steps.

Practical 3: Write a Python program to implement the ε-greedy strategy in the n-Armed Bandit problem. Compare the performance with different ε values and plot the average reward over time.

Practical 4: Write a Python program to estimate action-values using sample averages in a multi-armed bandit setting. Track the convergence of action-value estimates over multiple episodes.

Practical 5: Write a Python program to implement optimistic initial values in the n-Armed Bandit problem. Show how optimism in the face of uncertainty affects the agent’s exploration behavior.

Practical 6: Write a Python program to model a simple Finite Markov Decision Process (MDP). Define a small

number of states and actions and simulate transitions and rewards for different policies.

Practical 7: Write a Python program to implement Monte Carlo prediction for estimating the value of states in an episodic environment. Use sample episodes to compute returns and update value estimates.

Practical 8: Write a Python program to implement Monte Carlo control for learning optimal policy in a Grid World environment. Use exploring starts or ε-greedy behavior policy for action selection.

Practical 9: Write a Python program to implement off-policy Monte Carlo control using importance sampling. Evaluate a target policy while following a different behavior policy and analyze convergence.

Practical 10: Write a Python program to simulate Tic-Tac-Toe as a reinforcement learning problem. Define the state, action, and reward structure and simulate random or greedy agent play to learn the winning strategy.

References:

1. Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning-An Introduction”,2nd Edition,The MIT Press,2018
2. Marco Wiering , Martijn van Otterlo Reinforcement Learning: State-of-the-Art (Adaptation, Learning, and Optimization (12)) 2012th Edition.
3. Vincent François-Lavet , Peter Henderson , Riashat Islam, An Introduction to Deep Reinforcement Learning (Foundations and Trends(r) in Machine Learning) , 2019

Course Code	Course Name	L	T	P	Cr
DOAI250065	AI for Cybersecurity	3	0	2	4

Course Outcomes:

CO1: Explain cybersecurity fundamentals, threats, attack vectors, and the role of AI/ML/DL in security.
CO2: Apply supervised and unsupervised ML techniques for threat detection and anomaly identification.
CO3: Implement deep learning models such as CNNs, RNNs, and autoencoders for malware and intrusion detection.
CO4: Utilize NLP, network traffic analysis, SIEM systems, and open-source tools for AI-driven security applications.
CO5: Evaluate challenges, ethics, adversarial ML, privacy concerns, and future trends in AI-powered cybersecurity.

Module 1:

Introduction to Cybersecurity and AI (9 Hours)

Cybersecurity fundamentals, threats, and attack vectors, Overview of AI, ML, and DL concepts, Role of AI in cybersecurity, Data sources for cybersecurity: logs, packets, traffic

Module 2:

Machine Learning in Cybersecurity (9 Hours)

Supervised and unsupervised learning techniques, Feature engineering for security datasets, Classification models for threat detection, Clustering techniques for anomaly detection

Module 3:

Deep Learning for Security Analytics (9 Hours)

Neural networks for security applications, Convolutional Neural Networks for malware classification, Recurrent Neural Networks for behavior analysis, Autoencoders for anomaly and intrusion detection

Module 4:

Use Cases and Tools in AI Cybersecurity (9 Hours)

Phishing detection using NLP techniques, Network traffic analysis using ML, SIEM systems and AI integration, Introduction to open-source tools like Snort, Zeek, and ELK stack

Module 5:

Challenges, Ethics, and Future Trends (9 Hours)

Adversarial machine learning, Bias and fairness in security AI systems, Ethical implications and privacy concerns, Emerging trends: Federated learning, Explainable AI in cybersecurity

Labs / Practicals:

1. Data preprocessing and visualization of cybersecurity datasets
2. Building and evaluating ML classifiers for intrusion detection
3. Malware classification using CNNs
4. Anomaly detection using Autoencoders
5. Phishing URL detection using text classification
6. Real-time network traffic analysis using Python and ML
7. Project: Developing a smart intrusion detection system

References:

1. Machine Learning and Security by Clarence Chio and David Freeman
2. AI in Cybersecurity by Leslie F. Sikos
3. Hands-On Machine Learning for Cybersecurity by Soma Halder and Sinan Ozdemir
4. Deep Learning for Cybersecurity by Xiaofeng Chen et al.
5. Research papers from IEEE, ACM, and arXiv related to AI and cybersecurity

Course Code	Course Name	L	T	P	Cr
DOEE250121	Internet of Things	3	0	2	4

Course Outcomes:

CO 1 Explain the concept of IoT
CO 2 Networking basics for IoT application development
CO 3 Analyze various protocols for IoT
CO 4 Design a PoC of an IoT system using various hardware platforms
CO 5 Apply data analytics and use cloud offerings related to IoT
CO 6 Analyze applications of IoT in real time scenario

Module 1:

Overview of IoT

Introduction to the Internet of Things, IoT system architecture and standards, Networking Basics - TCP/IP, Networking Basics - IP addressing basics (IPV4 and IPV6), Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks.

Module 2:

IoT hardware Platforms

IoT Design Methodology – Embedded computing logic – Microcontroller-System on Chips, Hardware platforms for prototyping IoT node- Arduino, Raspberry Pi, NodeMCU, ESP32, ARM Cortex Microcontrollers, IoT mote hardware platforms, Swadeshi RISC V based solutions, Interfacing sensors and actuators with hardware platforms, Developing IoT applications using Raspberry Pi with Python Programming.

Module 3:

IoT connectivity & Protocols

IoT Access Technologies: WiFi, Zigbee, Zwave, Bluetooth,UWB, sub1GHz, LoRaWAN, Sigfox and NB-IoT, Topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN, IoT application level protocols: MQTT, CoAP, XMPP, HTTP/Rest Services, WebSockets.

Module 4:

Data analytics, Cloud and IoT Security

No SQL Databases Vs SQL Databases, Apache web servers, JSON, Open and commercial Cloud solutions for IoT, Python Web Application Frameworks for IoT, IoT data visualisation tools, IoT Security - Need for encryption, standard encryption protocol, lightweight cryptography, Trust models for IoT, ARM Cortex Microcontroller Security, Root Security Services (RSS).

Module 5:

IoT Case studies

Smart Lighting, Smart home, Smart Agriculture, Smart farming, IoT for health care & patient monitoring, Smart and Connected Cities, Building end-to-end smart applications with TinyML.

Labs / Practicals:

Hands-on Sessions on IoT Application Development using IoT Hardware Kits (ESP32/STM32/Rpi etc.)

- IPv4 and IPv6 Implementation
- 6LoWPAN Network Implementation
- Interfacing multiple sensors and actuators with Processor Hardware using different communication protocols and develop an integrated monitoring system.
- Implementation of different IoT connectivity technologies including WiFi, Bluetooth, and BLE for various application scenarios.
- Implementation of MQTT protocol for IoT device communication
- Implementation of CoAP for resource-constrained IoT devices
- Integration of IoT devices with cloud platforms for data analytics

References:

Reference Books

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, —IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, Cisco Press, 2017
2. Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner, “Enabling things to talk – Designing IoT solutions with the IoT Architecture Reference Model”, Springer Open, 2016
3. VijayMadisetti , ArshdeepBahga, Adrian McEwen (Author), Hakim Cassimally “Internet of Things: A Hands-on-Approach” ArshdeepBahga& Vijay Madisetti, 2014.
4. Gian Marco Iodice, TinyML Cookbook: Combine artificial intelligence and ultralow-power embedded devices to make the world smarter
5. Olivier Hersent, David Boswarthick, Omar Elloumi , —The Internet of Things – Key applications and Protocols, Wiley, 2012

Course Code	Course Name	L	T	P	Cr
DOEE250127	IOT: Technologies and Applications	3	0	2	4

Course Outcomes:

CO1: Explain the evolution, architecture, functional blocks, and ecosystem of IoT with sensors, actuators, and smart objects.
CO2: Describe IoT communication protocols across physical, network, and application layers including 6LoWPAN, CoAP, and MQTT.
CO3: Apply IoT design methodologies using microcontrollers, SoCs, and hardware platforms like Arduino and Raspberry Pi.
CO4: Analyze IoT data management, structured/unstructured data, cloud services, and data analytics challenges.
CO5: Evaluate IoT applications in homes, industries, infrastructure, and Industry 4.0 through case studies.

Module 1:

FUNDAMENTALS OF IoT- Evolution of Internet of Things, Enabling Technologies, M2M Communication, IoT World Forum (IoTWF) standardized architecture, Simplified IoT Architecture, Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects.

Module 2:

IoT PROTOCOLS- IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.11ah and Lora WAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks,6LoWPAN, Application Transport Methods: SCADA, Application Layer Protocols: CoAP and MQTT.

Module 3:

DESIGN AND DEVELOPMENT- Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks IoT Platform overview: Overview of IoT supported Hardware platforms such as: Raspberry pi, Arduino Board details

Module 4:

DATA ANALYTICS AND SUPPORTING SERVICES: Data Analytics: Introduction, Structured Versus Unstructured Data, Data in Motion versus Data at Rest, IoT Data Analytics Challenges, Data Acquiring, Organizing in IoT/M2M, Supporting Services: Computing Using a Cloud Platform for IoT/M2M Applications/Services, Everything as a service and Cloud Service Models.

Module 5:

CASE STUDIES/INDUSTRIAL APPLICATIONS: IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments, Industry 4.0 concepts.

Labs / Practicals:

Practical 1: Introduction to IoT Hardware
Objective: Identify and study IoT components – sensors, actuators, microcontrollers, and communication modules.
Tools: Arduino/Raspberry Pi, DHT11, IR sensor, Relay module
Practical 2: Sensor-Actuator Communication
Objective: Interfacing temperature sensor and controlling a fan (motor or LED) based on threshold.
Concepts: Smart objects, actuator logic, sensor reading
Tools: Arduino/ESP32 + relay + DHT11
Practical 3: IoT Stack and Ecosystem Mapping (Simulation)
Objective: Map IoT architecture components to a real-world application (smart home, smart city).
Tool: Diagramming tools (e.g., Lucidchart) or simulation tools (Tinkercad)
Practical 4: Wireless Communication – LoRa/802.15.4/802.11ah (Simulation or Real)

Objective: Demonstrate communication between two LoRa or Zigbee nodes (or simulate in software).
Tools: LoRa modules (SX1278), Zigbee (XBee), or use simulators like Cooja (Contiki OS)
Practical 5: Using MQTT Protocol with Node-RED
Objective: Publish sensor data to MQTT broker and visualize it.
Tools: Node-RED, Mosquitto MQTT, Arduino/ESP32
Practical 6: Using CoAP with IoT Devices
Objective: Set up a CoAP client-server demo using libcoap or Copper plugin in Firefox.
Tools: CoAP client tools, ESP8266/ESP32 (optional), emulator
Practical 7: LED Blinking and Button Press with Arduino
Objective: Practice microcontroller programming fundamentals.
Tools: Arduino IDE + UNO board
Practical 8: Sensor Integration with Raspberry Pi
Objective: Read data from DHT11 or PIR sensor using Python on Raspberry Pi.
Tools: Raspberry Pi, Python, GPIO
Practical 9: Build a Mini IoT System
Objective: Create a complete system (e.g., smart temperature monitor) using microcontroller + Wi-Fi module
Tools: Arduino/ESP32 + DHT11 + OLED or LED display
Practical 10: Send Data to Cloud (ThingSpeak/Blynk)
Objective: Push sensor data to ThingSpeak or Blynk and visualize it.
Tools: ESP8266/ESP32, Wi-Fi, ThingSpeak account
Practical 11: Analyze IoT Data Logs
Objective: Collect real-time data from sensors, store in CSV, and perform basic analytics in Excel/Python.
Tools: Python (Pandas), Excel
Practical 12: Deploy IoT Service on Cloud (AWS IoT Core / Google Firebase)
Objective: Setup a basic cloud-based IoT application (sensor -> cloud -> dashboard)
Tools: ESP32, AWS IoT Core / Firebase Realtime DB
Practical 13: Home Automation System
Objective: Control home appliances using smartphone via IoT
Tools: ESP8266 + Relay + Mobile App (Blynk/Adafruit IO)

References:

1. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012 (for Unit2).
2. “From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence”,Jan Ho¨ ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.
3. Architecting the Internet of Things,Dieter Uckelmann, Mark Harrison, Michahelles and Florian(Eds),Springer,2011.
4. Recipes to Begin, Expand, and Enhance Your Projects,2nd Edition,Michael Margolis,Arduino Cookbook and O"Reilly Media,2011.
5. Internet of Things: A Hands-On-Approach By Arshdeep Bahga and Vijay Madisetti
6. Getting Started with the Internet of Things By Cuno Pfister (Make: Books)