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.Sc (AI and ML)

(2025-26 Scheme)

रा.इ.सू.प्रौ.सं

NIELIT

Deemed to be University

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

Vision

To be a leading center of excellence in AI and Machine Learning, advancing education, research, and innovation while cultivating ethical, globally competent professionals who drive transformative change through intelligent systems.

Mission

The mission of the B.Sc. (AI/ML) program is to equip students with strong theoretical foundations and hands-on skills in Artificial Intelligence and Machine Learning. The program aims to cultivate a research-driven and ethically grounded mindset, empowering graduates to lead innovation, address complex real-world challenges, and contribute meaningfully to industry, technology, and society.

Program Education Objectives

PEO1: Acquire foundational knowledge in Artificial Intelligence and Machine Learning.
PEO2: Develop intermediate-level knowledge and skills in AI & ML for problem-solving.
PEO3: Demonstrate initiative and teamwork in advancing innovation across AI-driven product development, applied research, and technology-focused industries.
PEO4: Develop the ability to adapt and thoughtfully address challenges in the evolving field of AI and ML, including technical, ethical, and managerial aspects.

Program Outcomes

PO1: Demonstrate proficiency in Machine learning model building and fine tuning.
PO2: Apply ML algorithms to solve complex real world problems.
PO3: Exhibit effective communication skills in conveying technical concepts orally and in writing.
PO4: Engage in collaborative projects and demonstrate the ability to work effectively in a team.
PO5: Considering ethical aspects in applying AI for professional and societal contexts.
PO6: Possess a comprehensive understanding in recent trends in AI & ML.

Program Specific Outcomes

PSO1: Apply foundational and advanced concepts of Artificial Intelligence and Machine Learning to design, develop, and evaluate intelligent systems and solutions across domains.
PSO2: Demonstrate the ability to collect, preprocess, analyse, and interpret structured and unstructured data using appropriate tools, algorithms, and statistical techniques.
PSO3: Engage in problem identification, literature analysis, and solution formulation for real-world challenges, fostering a research-oriented mindset and innovative thinking.
PSO4:Recognize and address ethical, societal, and legal implications of AI/ML technologies, ensuring fairness, transparency, accountability, and inclusivity in AI systems.
PSO5: Exhibit adaptability to emerging trends and technologies in AI/ML, preparing for lifelong learning, higher studies, and careers in industry, research, and entrepreneurship.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	16
Ability Enhancement Courses	AE	8
Professional Elective Courses	PE	20
Audit Courses (without grade or credit)	AU	0
Value Added Courses	VA	6
Open Elective Courses	OE	12
Program Core Courses	PC	88
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE - H	3
Professional Elective Courses	PE - H	12
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE - HWR	15
Total Credits (Common track)		**150**
Total Credits (Common Track [150] + Honours Track [15])		**165**
Total Credits (Common Track [150] + Honours with Research Track [15])		**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-BSCAIM-101	DOEE250063	Digital Logic and Circuits	3	0	2	4
PC-BSCAIM-102	DOAI250023	Data Visualization	2	0	4	4
PC-BSCAIM-103	DOAI250030	Introduction to Artificial Intelligence	3	1	0	4
OE-BSCAIM-104	Elective	Open Elective				4
VA-BSCAIM-105	DASH250019	Communicative English	1	0	2	2
AU-BSCAIM-106	Elective	Audit Elective				0

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BSCAIM-201	DCSE250108	Problem Solving and Programming using C	3	0	2	4
PC-BSCAIM-202	DCSA250057	Object Oriented Programming using C++	3	0	2	4
PE-BSCAIM-203	Elective	Program Elective				4
OE-BSCAIM-204	Elective	Open Elective				4
AE-BSCAIM-205	DASH250084	Organizational Behaviour	2	0	0	2
EE-BSCAIM-206	EE	Project with Seminar				8

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BSCAIM-301	DCSE250053	Data Structure using C++	2	0	4	4
PC-BSCAIM-302	DCSE250060	Database Management Systems	2	0	4	4
PC-BSCAIM-303	DOAI250044	Probability and Statistical Techniques	3	1	0	4
OE-BSCAIM-304	Elective	Open Elective				4
VA-BSCAIM-305	DASH250056	Financial Accounting & Management	2	0	0	2
EE-BSCAIM-306	EE	Mini Project with Seminar				4

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BSCAIM-401	DCSA250071	Software Engineering	3	1	0	4
PC-BSCAIM-402	DCSE250026	Computer Architecture	3	1	0	4
PC-BSCAIM-403	DCSA250065	Programming in Java	3	1	0	4
PC-BSCAIM-404	DCSA250084	Web Technologies	3	0	2	4
PE-BSCAIM-405	Elective	Program Elective				4
AU-BSCAIM-406	Elective	Audit Elective				0

Semester 5

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-BSCAIM-501	DCSE250035	Computer Networks	3	0	2	4
PC-BSCAIM-502	DCSA250011	Computer Oriented Numerical Methods	3	1	0	4
PC-BSCAIM-503	DCSE250042	Introduction to Operating System	3	1	0	4
EE-BSCAIM-504	EE	Internship				4
PE-BSCAIM-505	Elective	Program Elective				4
AE-BSCAIM-506	DASH250026	Design Thinking	1	0	2	2
AU-BSCAIM-507	Elective	Audit Elective				0

Semester 6						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BSCAIM-601	DOAI250032	Machine Learning	3	0	2	4
PC-BSCAIM-602	DCSE250029	Computer Graphics	3	1	0	4
PE-BSCAIM-603	Elective	Program Elective				4
AE-BSCAIM-604	DASH250016	Communication Skills	1	0	2	2
AE-BSCAIM-605	DOEE250038	Capstone Project planning	1	0	2	2
VA-BSCAIM-606	DASH250104	Technical Presentation and Report Writing	0	0	4	2

Semester 7						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BSCAIM-701	DCSE250120	Theory of Computation	3	1	0	4
PC-BSCAIM-702	DOAI260041	Natural Language Processing	2	0	4	4
PC-BSCAIM-703	DOAI250014	Data Analysis using Python	2	0	4	4
PE-BSCAIM-704	Elective	Program Elective				4
PE-BSCAIM-705 (H)	Elective	Program Elective				4
EE-BSCAIM-705 (HWR)	EE	Minor Project				4
AU-BSCAIM-706	Elective	Audit Elective				0

Semester 8						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BSCAIM-801	DOAI250024	Deep Learning	2	0	4	4
PC-BSCAIM-802	DCSE250044	Cyber Security	3	0	2	4
PE-BSCAIM-803 (H)	Elective	Program Elective				4
EE-BSCAIM-803 (HWR)	EE	Research Project/ Dissertation				11
PE-BSCAIM-804 (H)	Elective	Program Elective				4
EE-BSCAIM-805 (H)	EE	Dissertation				3

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4	
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4	
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4	
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4	
DCSA250017	Linux Environment	3	0	2	4	
DOAI250019	Data Mining and Warehousing	3	0	2	4	
DCSE250020	Cloud Computing	3	0	2	4	203
DCSE250043	Cryptography and Network Security	3	0	2	4	203
DCSA250068	Soft Computing	3	0	2	4	405
DCSE250011	Augmented Reality and Virtual Reality	3	0	2	4	405
DOAI250012	Business Intelligence	2	0	4	4	405
DCSE250081	Fuzzy Logic and Fuzzy Sets	3	1	0	4	505
DOAI250051	Text Analysis	3	1	0	4	505
DOAI250065	AI for Cybersecurity	3	0	2	4	505
DOAI250048	Reinforcement Learning	3	0	2	4	604
DCSE250045	Cyber Threat Intelligence	3	0	2	4	704
DOAI250027	Generative AI	3	0	2	4	704
DCSA250029	Digital Image Processing	3	0	2	4	705
DOAI250009	Big Data Analytics	3	0	2	4	705
DOAI250021	Data Science	3	0	2	4	705
DCSA250003	Mobile Application Development using Android	2	1	2	4	803
DCSA250009	Blockchain Technology	3	1	0	4	803
DCSA250054	Multimedia & Animation Techniques	3	0	2	4	803
DASH250015	Cognitive Systems	3	1	0	4	804
DCSA250020	Computer Vision	3	0	2	4	804

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
DOEE250063	Digital Logic and Circuits	3	0	2	4

Course Outcomes:

CO1: Understand and describe the principles of digital systems and binary number operations.
CO2: Apply Karnaugh mapping (K-Map) to simplify Boolean expressions and optimize digital circuits.
CO3: Analyze and design basic combinational circuits using various digital components.
CO4: Synthesize and evaluate synchronous sequential circuits using storage elements and HDL.
CO5: Design and implement various types of registers and counters using HDL.

Module 1:

Introduction
Digital Systems Binary Numbers – Conversions – Types – Codes – Storage and Registers – Binary Logic – Boolean Algebra – Theorems and Properties – Functions – Canonical and Standard Forms – Other Logic Operations – Digital Logic Gates –Integrated Circuits

Module 2:

Gate-Level Minimization
Map Method – Four-Variable K-Map – Product-of-Sums Simplification – Don't-Care Conditions – NAND and NOR Implementation – Other Two-Level Implementations –Exclusive-OR Function – Hardware Description Language

Module 3:

Combinational Logic
Analysis Procedure – Design Procedure – Binary Adder–Subtractor – Decimal Adder – Binary Multiplier – Magnitude Comparator – Decoders – Encoders – Multiplexers – HDL Models of Combinational Circuits

Module 4:

Synchronous Sequential Logic
Storage Elements– Latches – Flip-Flops – Analysis of Clocked Sequential Circuits - Synthesizable HDL Models of Sequential Circuits – State Reduction and Assignment – Design Procedure

Module 5:

Registers and Counters
Registers– Shift Registers – Ripple Counters– Synchronous Counters – Other Counters – HDL for Registers and Counters

Labs / Practicals:

1.Binary to Decimal and vice-versa
2.Decimal to Hexadecimal and Vice-Versa
3.Digital Logic Gates
4.Simplification of Boolean Functions
5.Combinational Logic Circuits
  i.Code Converters

ii. Arithmetic (Adders, Subtractors, Multipliers, Comparators.
iii. Data Handling (Multiplexers, De-multiplexers, Encoders & Decoders).
6.Combinational Logic Circuit Design
7.Binary Adder-Subtractor Simulation
8.Decimal Adder Simulation
9.Binary Multiplier Simulation
10.Sequential Circuit Storage Elements: Flip-Flop Simulation

References:

1.M. Morris Mano, Michael D. Ciletti, “Digital design With an Introduction to the Verilog HDL”, Pearson, Sixth Edition, 2018.
2.M. Rafiquzzaman, “Fundamentals of Digital Logic and Microcomputer Design”, John Wiley & Sons, Inc., Fifth Edition, 2009.

Course Code	Course Name	L	T	P	Cr
DOAI250023	Data Visualization	2	0	4	4

Course Outcomes:

CO1: To introduce the fundamental concepts and methodologies of data visualization.
CO2: To understand various techniques for visualizing different types of data, including time series, networks, and hierarchies.
CO3: To explore the process of acquiring, parsing, and handling data for visualization.
CO4: To develop interactive data visualizations using frameworks like D3.js, Tableau, and Matplotlib.
CO5: To analyze and interpret security-related data through visualization techniques.

Module 1:

INTRODUCTION
Context of data visualization – Definition, Methodology, Visualization design objectives. Key Factors – Purpose, visualization function and tone, visualization design options – Data representation, Data Presentation, Seven stages of data visualization, widgets, data visualization tools.

Module 2:

VISUALIZING DATA METHODS
Mapping - Time series - Connections and correlations - Scatterplot maps - Trees, Hierarchies and Recursion - Networks and Graphs, Info graphics

Module 3:

VISUALIZING DATA PROCESS
Acquiring data, - Where to Find Data, Tools for Acquiring Data from the Internet, Locating Files for Use with Processing, Loading Text Data, Dealing with Files and Folders, Listing Files in a Folder, Asynchronous Image Downloads, Advanced Web Techniques, Using a Database, Dealing with a Large Number of Files. Parsing data - Levels of Effort, Tools for Gathering Clues, Text Is Best, Text Markup Languages, Regular Expressions (regexps), Grammars and BNF Notation, Compressed Data, Vectors and Geometry, Binary Data Formats, Advanced Detective Work.

Module 4:

INTERACTIVE DATA VISUALIZATION
Drawing with data – Scales – Axes – Updates, Transition and Motion – Interactivity - Layouts – Geomapping – Exporting, Framework – D3.js, tableau, Matplotlibsns.

Module 5:

SECURITY DATA VISUALIZATION
Port scan visualization - Vulnerability assessment and exploitation - Firewall log visualization -Intrusion detection log visualization -Attacking and defending visualization systems - Creating security visualization system.- Visual Data Analysis Techniques, Interaction Techniques - Systems and Applications.

Labs / Practicals:

1. Graphical Integrity & Perception
Create accurate vs misleading visuals; understand scale, axis distortion.

2. Data-Ink Ratio & Redesign
Apply Tufte's principles to redesign charts by reducing non-data ink.

3. Bar & Dot Plots
Visualize amounts using grouped bars, stacked bars, dot plots, and heatmaps.

4. Distribution Visualization
Use histograms, density plots, Q-Q plots to explore data distribution.

5. Pie & Proportional Charts
Visualize proportions using pie charts and donut charts; compare with bar plots.

6. Time Series Visualization
Plot individual and multiple time series with different response variables.

7. Geospatial Visualization
Create scatterplot maps, choropleth maps using geographic data.

8. Multivariate Data Visualization
Use scatterplot matrices, bubble charts, parallel coordinates for multidimensional data.

9. Network & Hierarchical Visualization
Build tree maps, sunburst charts, and network graphs for relationships.

10. Visual Analytics Dashboard
Create interactive dashboards in Tableau with filters, KPIs, and insights.

11. Python Library Comparison
Compare matplotlib, seaborn, plotly, and bokeh using the same dataset.

References:

1. Edward R. Tufte, The Visual Display of Quantitative Information, 2nd edition, Graphics Press, 2007
2. Claus O. Wilke, Fundamentals of Data Visualization, O'Reilly, 2019
3. Ben Fry, Visualizing Data: Exploring and Explaining Data with the Processing Environment, O'reilly, 1st edition, 2008
4. A. Kerren, J. T. Stasko, J. Fekete, C. North, Information Visualization, Springer, 2008
5. Alex Campbell, Data Visualization: Ultimate Guide to Data Mining and Visualization, 2020
6. A. Loth, Visual Analytics with Tableau, Willey, 2019
7. Kieran Healy, Data Visualization: A Practical Introduction, Princeton University Press, 2018
8. Alexandru C. Telea, Data Visualization: Principles and Practice, Second Edition, CRC Press, 2015

Course Code	Course Name	L	T	P	Cr
DOAI250030	Introduction to Artificial Intelligence	3	1	0	4

Course Outcomes:

CO1: Understand AI fundamentals, state-space search, and heuristic problem-solving methods.
CO2: Represent knowledge using logic, rules, and predicate calculus.
CO3: Apply reasoning methods under uncertainty including probabilistic and fuzzy logic approaches.
CO4: Explain key concepts in natural language processing and basics of machine learning.
CO5: Demonstrate the design of expert systems and AI-based planning algorithms.

Module 1:

Introduction and Problems, State Space Search & Heuristic Search Techniques:
What is AI? The AI Problems, The Underlying Assumption, What Is An AI
Techniques, The Level of The Model, Criteria For Success, Some General References, One Final Word, Defining The Problems as a State Space Search, Production Systems, Production Characteristics, Production System Characteristics, And Issues In The Design Of Search Programs, Additional Problems, Generate And-Test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means Ends Analysis.

Module 2:

Knowledge representation:
Knowledge Representation Issues -Representations And Mappings, Approaches To Knowledge Representation; Using Predicate Logic - Representation Simple Facts In Logic, Representing Instance And Isa Relationships, Computable Functions And Predicates, Resolution; Representing Knowledge Using Rules -Procedural Versus Declarative Knowledge, Logic Programming, Forward Versus Backward Reasoning.

Module 3:

Reasoning:
Symbolic Reasoning Under Uncertainty -Introduction To Non-monotonic Reasoning, Logics For Non-monotonic Reasoning; Statistical Reasoning -Probability And Bays' Theorem, Certainty Factors And Rule-Base Systems, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic; Weak Slot-and-Filler and Strong Slot-and-Filler Structures - Semantic Nets, Frames, Conceptual Dependency, Scripts, CYC

Module 4:

Natural Language Processing and Machine learning and Neural Networks:
Introduction, Syntactic Processing, Semantic Analysis, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking; Definition and examples of machine learning, supervised learning, unsupervised learning, reinforcement learning, introduction to neural networks.

Module 5:

Expert Systems and AI planning systems:
Representation using domain knowledge, Expert System shell, knowledge acquisition; Definition and examples of Planning Systems, planning as search, operator-based planning, propositional planning.

Labs / Practicals:

NA

References:

1. “Artificial Intelligence” -By Elaine Rich And Kevin Knight (2nd Edition) Tata Mcgraw-Hill
2. “Artificial Intelligence: A Modern Approach”, Stuart Russel, Peter Norvig, PHI
3. Introduction to AI & Expert Sys., Pitterson, D.N

Course Code	Course Name	L	T	P	Cr
DASH250019	Communicative English	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 : Expand vocabulary and usage for effective communication.

Module 1:

Communication: Theory and Practice
Basics of communication: Introduction, meaning and definition, process of communication etc.
Types of communication: formal and informal, verbal, non-verbal and written Barriers to effective communication.

Art of Effective communication: Clarity, Conciseness, Active listening, Non – verbal communication, Consider your audience, Feedback, Technology, Simplification of words
7 Cs for effective communication (considerate, concrete, concise, clear, complete, correct, courteous).

Module 2:

Writing Skills
Parts of Speech, Active and Passive Voice , Tenses , Punctuation.

Module 3:

Reading Comprehension
Comprehension, vocabulary enhancement and grammar exercises based on reading of the following texts:
Section-1
Malgudi Days: R.K. Narayan
The Room on Roof: Ruskin Bond
“The Gift of the Magi” by O. Henry
“Uncle Podger Hangs a Picture” Jerome K. Jerome
Section-2
“ On His Blindness” by John Miltion
Stopping by Woods on a Snowy Evening by Robert Frost,
Where the Mind is Without Fear by Rabindranath Tagore,
The Solitary Reaper by William Wordsworth.

Module 4:

Writing Skills
Effective writing: essentials of effective writing, various steps in effective writing.
The art of précis writing: Job application, business letter Drafting e-mail, notices, minutes of a meeting, Resume writing.
Filling-up different forms such as banks and on-line forms for placement etc.

Module 5:

Vocabulary
Words often confused, One-word substitution, Idioms and phrases.
Glossary of administrative terms.

Labs / Practicals:

Practical 1: Pronunciation & Phonetics Practice
Practice vowel & consonant sounds

Word stress and sentence stress
Intonation using short passages

Practical 2: Listening Comprehension
Listen to audio clips
Answer comprehension-based questions
Focus: keywords, inference, tone

Practical 3: Reading Comprehension
Read a prescribed text
Vocabulary, inference, theme-based questions
Grammar from passage

Practical 4: Grammar Application Workshop
Worksheets on tenses
Active/passive voice practice
Punctuation correction
Paragraph rewriting

Practical 5: Paragraph Writing
Write a paragraph using 7 Cs
Focus on clarity and coherence

Practical 6: Email & Letter Drafting
Draft a formal email
Write a job application letter
Use correct structure, tone, and formatting

Practical 7: Resume Preparation
Prepare a professional résumé
Tailor it to a sample job description
Focus on formatting & concise language

Practical 8: Précis Writing
Compress 250–300 words
Maintain key ideas and clarity
Logical flow and precision

Practical 9: Vocabulary Enhancement
Confusing words
One-word substitutions
Idioms & phrases
Synonyms / antonyms

Practical 10: Group Discussion / Oral Presentation
Participate in GD
Or deliver a 2-minute presentation
Evaluate fluency, confidence, content & pronunciation

References:

1. J.D.O'Connor. Better English Pronunciation. Cambridge: Cambridge University Press.
2. Lindley Murray. An English Grammar: Comprehending Principles and Rules. London: Wilson and Sons.
3. Kulbhushan Kumar, Effective Communication Skills, Khanna Publishing House, New Delhi.
4. Margaret M. Maison. Examine your English. Orient Longman: New Delhi.
5. M. Ashraf Rizvi. Effective Technical Communication. Mc-Graw Hill: Delhi.
6. John Nielson. Effective Communication Skills. Xlibris.
7. Oxford Dictionary.
8. Roget's Thesaurus of English Words and Phrases.

9. Collin’s English Dictionary.

Course Code	Course Name	L	T	P	Cr
DCSE250108	Problem Solving and Programming using C	3	0	2	4

Course Outcomes:

CO1. Understand computation models, algorithms, and flowcharts, and design solutions for mathematical and logical problems using structured approaches.
CO2. Develop simple C programs using variables, data types, operators, conditional statements, and iterative constructs for problem solving.
CO3. Implement arrays, strings, and modular functions to solve computational problems using recursion, algorithms, and top-down programming techniques.
CO4. Apply pointers for dynamic memory allocation, array manipulation, function arguments, and structured programming to build efficient applications.
CO5. Design and manipulate data using structures and unions, including arrays of structures, nested structures, and user-defined data management.

Module 1:

Introduction to Programming: The Basic Model of Computation, Algorithms, Flow-charts, Programming Languages, Compilation, Linking and Loading, Testing and Debugging, Documentation
Algorithms for Problem Solving :Exchanging values of two variables, summation of a set of numbers, Decimal Base to Binary Base conversion, Reversing digits of an integer, GCD (Greatest Common Division) of two numbers, Test whether a number is prime, Organize numbers in ascending order, Find square root of a number, factorial computation, Fibonacci sequence, Evaluate ‘sin x’ as sum of a series, Reverse order of elements of an array, Find largest number in an array, Print elements of upper triangular matrix, multiplication of two matrices, Evaluate a Polynomial

Module 2:

Introduction to ‘C’ Language: Character set, Variables and Identifiers, Built-in Data Types, Variable Definition, Arithmetic operators and Expressions, Constants and Literals, Simple assignment statement, Basic input/output statement, Simple ‘C’ programs.
Conditional Statements and Loops:Decision making within a program, Conditions, Relational Operators, Logical Connectives, if statement, if-else statement, Loops: while loop, do while, for loop, Nested loops, Infinite loops, Switch statement, structured Programming.

Module 3:

Arrays & Functions: One dimensional arrays: Array manipulation; Searching, Insertion, Deletion of an element from an array; Finding the largest/smallest element in an array; Two dimensional arrays, Addition/Multiplication of two matrices, Transpose of a square matrix; Null terminated strings as array of characters, Standard library string functions, Top-down approach of problem solving, Modular programming and functions, Standard Library of C functions, Prototype of a function: Formal parameter list, Return Type, Function call, Block structure, Passing arguments to a Function: call by reference, call by value, Recursive Functions, arrays as function arguments.

Module 4:

Pointers: Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, Arrays and Pointers, pointer arrays, pointers and structures, dynamic memory allocation.

Module 5:

Structures and Unions: Structure variables, initialization, structure assignment, nested structure, structures and functions, structures and arrays: arrays of structures, structures containing arrays, unions

Labs / Practicals:

1. Write a C program to display your name, roll number, and branch.
2. Write a C program to calculate the area and perimeter of a rectangle given its length and breadth.

3. Write a C program to check whether a number is even or odd using conditional statements.
4. Write a C program to find the largest of three numbers using nested if-else.
5. Write a C program to generate and display the Fibonacci series up to n terms.
6. Write a C program to calculate the factorial of a number using a function.
7. Write a C program to perform addition, subtraction, and multiplication of two matrices.
8. Write a C program to demonstrate pointer usage by swapping two numbers using call by reference.
9. Write a C program to create a file, write data into it, and display its contents on the screen.

References:

1. Kanetkar Y, “Let us C”, BPB Publications, 2007.
2. E. Balagurusamy, “Programming with ANSI-C”, Fourth Edition,2008, Tata McGraw Hill.
3. B.W. Kernighan & D. M. Ritchie, “The C Programming Language”, Second Edition, 2001,Pearson Education
4. Pradip Dey , Manas Ghosh, “Programming in C”, Oxford University Press, 2007

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
DASH250084	Organizational Behaviour	2	0	0	2

Course Outcomes:

CO1: Understand the fundamentals and models of organizational behavior.
CO2: Analyze the influence of individual factors like personality, perception, and attitude on organizational outcomes.
CO3: Apply concepts of motivation, leadership, and group dynamics to team-based environments.
CO4: Evaluate organizational structure, culture, and communication systems in modern workplaces.
CO5: Demonstrate awareness of ethical practices, diversity, emotional intelligence, and behavioral trends in the Indian context.

Module 1:

Introduction to Organizational Behaviour
Definition, scope, and importance of OB; interdisciplinary nature; OB models (Autocratic, Custodial, Supportive, Collegial, System); OB in tech organizations and AI-driven workplaces; Relevance to individual and team effectiveness.

Module 2:

Individual Behavior and Personality
Perception – Process and Errors; attitudes and values; Personality traits – Big Five, MBTI; Emotional Intelligence; learning styles and their role in collaborative work; case examples from Indian tech/startup environments.

Module 3:

Motivation and Leadership
Motivation theories – Maslow, Herzberg, McClelland, Expectancy theory; application in project teams; Leadership theories – Trait, Behavioral, Contingency; Transactional vs. Transformational leadership; Leadership in Agile and Data-driven teams.

Module 4:

Group Behavior and Teamwork
Groups vs. teams; team formation stages (Tuckman's model); Group decision-making; Conflict types and Resolution strategies; Collaboration tools and hybrid team culture in IT/AI settings; Role clarity and diversity in AI project teams.

Module 5:

Organizational Culture, Communication, and Trends
Types and functions of organizational culture; managing cultural change; formal and informal communication channels; communication barriers and tech-enabled solutions (Zoom, Slack, etc.); work stress and coping mechanisms; ethical behavior, diversity, and inclusion in Indian workplaces.

Labs / Practicals:

N/A

References:

1. K. Aswathappa – Organizational Behaviour, Himalaya Publishing House
2. L.M. Prasad – Organizational Behaviour, Sultan Chand & Sons
3. P. Subba Rao – Organizational Behaviour, Himalaya Publishing
4. C.B. Gupta – Management and Organisational Behaviour, Sultan Chand
5. S.S. Khanka – Organizational Behaviour, S. Chand Publishing

Course Code	Course Name	L	T	P	Cr
DCSE250053	Data Structure using C++	2	0	4	4

Course Outcomes:

CO1: Learn basic terminologies of linear and nonlinear data structures and algorithms.
CO2: Understand the concept of polynomial addition and sparse matrices using arrays.
CO3: Apply linked lists to solve problems related to stacks, queues, and sparse matrices.
CO4: Understand the operations and traversals of binary trees.
CO5: Apply graph algorithms to solve problems like topological sorting and finding minimum cost spanning trees.

Module 1:

Introduction
Basic terminologies – Linear and Nonlinear data structures -
Algorithm: Definition – Pseudo code – Analysis – Design Techniques.

Module 2:

Arrays, Stacks & Queues
Representation – Polynomial Addition – Sparse Matrices – Multidimensional Arrays - Stacks and Queues - Stack ADT – Operations – Evaluation of Expressions – Queue ADT –Operations – Application – Multiple Stacks and Queues.

Module 3:

Lists
Singly Linked Lists – Linked Stacks and Queues – Operations – Circularly Linked Lists – Equivalence Relations – Sparse Matrices – Doubly Linked Lists.

Module 4:

Trees
Basic Terminologies – Binary trees – Representation, Operations, Traversals, Types – Applications of Trees.

Module 5:

Graphs
Basic Terminologies – Representation, Operations, Traversals – Applications - Shortest path problem, Topological sorting, Minimum Cost Spanning trees.

Labs / Practicals:

1.Searching Algorithms (with the number of key comparisons) - Sequential, Binary and Fibonacci search algorithms.
2.Evaluation of arithmetic expression.
3.Stack, Queue, Circular queue, priority queue.
4.Singly Linked List, Doubly Linked List, Circular Linked List
5.Tree Traversal techniques.
6.Graph Traversal techniques.
7.Dijkstra’s Algorithm to obtain the shortest paths.

References:

1.Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Fundamentals of Data.
Structures in C", India University Press, Second Edition, 2008.
2.Debasis Samanta, "Classic Data Structures", Prentice-Hall of India, Pvt. Ltd., India, Seventeenth Printing, Second Edition, 2009.
3.Dinesh P Mehta & Sartaj Sahni, Handbook of Data Structures and Applications, Second Edition, Chapman and Hall, 2020.

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
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
DASH250056	Financial Accounting & Management	2	0	0	2

Course Outcomes:

CO1: Describe fundamental accounting principles and management concepts.
CO2: Prepare basic accounting statements such as journal entries, ledger, and trial balance.
CO3: Analyze financial statements using ratio analysis and cost analysis techniques.
CO4: Explain budgeting, working capital management, and financial planning in organizations.
CO5: Apply financial decision-making concepts in real-world business or startup contexts.

Module 1:

Introduction to Accounting and Management:
Need and scope of accounting, principles and concepts of accounting, objectives and functions of management, differences between accounting and management, users of accounting information, relevance of financial literacy for computer professionals and entrepreneurs.

Module 2:

Accounting Process:
Double-entry system of accounting, journal entries, posting to ledger; preparation of trial balance, understanding of cash book and petty cash book, simple problems on journal and ledger for service and product-based businesses.

Module 3:

Financial Statements and Analysis:
Preparation of trading account, profit and loss account, and balance sheet (with adjustments), Basics of financial statement interpretation, Ratio analysis: liquidity, profitability, solvency, turnover ratios, Use of spreadsheet tools (Excel) for analysis.

Module 4:

Cost and Budgetary Control:
Types of costs (fixed, variable, direct, indirect), marginal costing and break-even analysis, preparation of budgets: cash, production, flexible, budgetary control techniques, basics of working capital management.

Module 5:

Financial Management and ICT Integration:
Time value of money, capital budgeting basics, sources of finance, digital tools and ERPs (Tally, Zoho, SAP basics), GST basics and e-invoicing, case studies on fintech, budgeting in startups, digital payment ecosystems in India.

Labs / Practicals:

N/A

References:

1.S.N. Maheshwari & S.K. Maheshwari – An Introduction to Accountancy, Vikas Publishing.
2.T.S. Grewal – Double Entry Book Keeping, Sultan Chand & Sons.
3.M.Y. Khan & P.K. Jain – Financial Management: Text, Problems and Cases, McGraw Hill.
4.R.P. Rustagi – Fundamentals of Financial Management, Taxmann Publications.
5.Dr. Jawahar Lal – Accounting for Management, Himalaya Publishing.
6.NCERT Bookkeeping and Accountancy (Class XI & XII) – Foundation Concepts.

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
DCSE250026	Computer Architecture	3	1	0	4

Course Outcomes:

CO1. Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
CO2. Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
CO3. Determine various stages of instruction cycle and describe interrupts and their handling.
CO4. Explain CPU organization, instruction formats, RISC/CISC, pipelining, and apply arithmetic algorithms for data processing.
CO5. Analyze I/O mechanisms, data transfer modes, and evaluate memory hierarchy for system performance.

Module 1:

Data Representation and Basic Computer Arithmetic:
Introduction to Number system, complements, fixed and floating-point representations, alphanumeric representation, and basic binary arithmetic operations including addition, subtraction, multiplication and division.

Module 2:

Digital Logic Circuits:
Logic gates and truth tables, Boolean algebra, combinational circuits (half adder, full adder, decoders, encoder, multiplexers, demultiplexer), circuit simplification using Karnaugh maps and Don't Care conditions, sequential circuits (flip-flops with characteristic tables), and digital components including registers and memory units.

Module 3:

Register Transfer language:
Register Transfer language, Register Transfer, Bus and Memory Transfers (Tree State Bus Buffers, Memory Transfer), Arithmetic MicroOperations, Logic Micro Operations, Shift Micro- Operations, Arithmetic logical shift unit. Instruction codes, Computer registers, computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input output and interrupt, Design of Basic computer, design of Accumulator Unit.

Module 4:

Central Processing Unit:
Instruction format and Types, Addressing modes ,Stack operation , Data transfer and Manipulation, Program control, RISC and CISC processors and their differences, pipelining and Computer Algorithms, Signed and unsigned addition, Booth's Multiplication algorithm, Restoring and non restoring division, Decimal and Floating-point arithmetic operations.

Module 5:

Input –Output Interface:
Input-Output Interface, Asynchronous Data Transfer, Modes Of Transfer, Priority Interrupt, DMA, Input- Output Processor (IOP), CPU IOP Communication, Serial communication. Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory

Labs / Practicals:

NA

References:

1.M. Mano, Computer System Architecture, Pearson Education 1992
2.V. Rajaraman, T. Radhakrishnan, An Introduction to Digital Computer Design, Prentice Hall of India Pvt. Ltd.
3.W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India,2009

Course Code	Course Name	L	T	P	Cr
DCSA250065	Programming in Java	3	1	0	4

Course Outcomes:

CO1. Understand the fundamental concepts of JAVA.
CO2. Understand the fundamental concepts of object-oriented programming (OOP)
CO3. Implement object-oriented programs using classes and methods
CO4. Apply exception handling and file I/O in Java programs.
CO5. Implementation of JDBC.

Module 1:

Introduction to Java: History and evolution of Java, Features of Java (Platform independence, OOP, robustness, etc.),Java Virtual Machine (JVM), JDK, and JREJava program structure, Java development tools (IDE: NetBeans, Eclipse, or IntelliJ IDEA),Data types, variables, Keywords and identifiers, Input/Output, Operators,Decision-making statements, Looping statements, Jump statements, Array

Module 2:

Introduction to Object-Oriented Programming: Principles of OOPs, Classes and Objects in Java, Access Modifiers and Static Members, Constructors, Method Overloading, Static Members, Garbage Collection

Module 3:

Inheritance, Interfaces, and Packages: Inheritance, Polymorphism, Abstract Classes and Methods, Interfaces, Packages

Module 4:

Exception Handling in Java: Introduction to Exceptions, Exception Hierarchy, Handling Exceptions, Throwing Exceptions, User-Defined Exceptions, Best Practices in Exception Handling

Module 5:

Java Database Connectivity: Introduction to JDBC, Establishing a Database Connection, Executing SQL Queries, Handling Exceptions and Transactions

Labs / Practicals:

NA

References:

1. The Complete Reference / Herbert Schildt / Tata Mc-Graw Hill Pub. Co. Ltd. Head First Java / K. Sierra & B. Bates , O'Reilly
2. Internet and Java Programming / R. Krishnamoorthy & S. Prabhu / New Age
3. Java foundations of programming/ Prentice Hall of India Pvt. Ltd.
International (p) Ltd
4. Teach yourself Web Technologies Part – I / Ivan Bayross / BPB Publications

Course Code	Course Name	L	T	P	Cr
DCSA250084	Web Technologies	3	0	2	4

Course Outcomes:

CO1: Explain the fundamentals of web technology, including its history, evolution, client-server architecture, web browsers, web servers, types of web pages, and development environments.
CO2: Develop structured and styled web pages using HTML and CSS, incorporating responsive design principles.
CO3: Apply JavaScript for programming logic, event handling, DOM manipulation, form validation, and AJAX-based interactions.
CO4: Build dynamic web applications using server-side scripting (PHP/Node.js) and integrate them with databases (MySQL).
CO5: Demonstrate skills in website deployment, hosting, web security, content management systems, and version control with Git.

Module 1:

Definition of Web Technology, history, and evolution of the web. Understanding client-server architecture. Web Browsers and Web Servers: Definition, features, and functionalities. Web page types: static, dynamic, responsive, and interactive. Introduction to HTML, CSS, and JavaScript. Overview of web development environments.

Module 2:

Basic structure of an HTML document, HTML tags, attributes, and elements. Creating and formatting text, links, images, lists, tables, and forms. Introduction to Cascading Style Sheets, syntax, selectors, and types (inline, internal, external). Styling text, layouts, and images. Responsive design with media queries.

Module 3:

Introduction, syntax, variables, operators, and control structures. Functions, events, and event handling. DOM manipulation and traversal. Form validation using JavaScript. Introduction to AJAX for asynchronous data loading. Integrating JavaScript with HTML and CSS.

Module 4:

Introduction to server-side scripting languages: PHP, Node.js. Setting up a local server environment. Database concepts: introduction to MySQL, SQL basics, and connecting a web application with a database.

Module 5:

Types of hosting services, domain name registration, and hosting providers. Uploading and managing websites on servers. FTP and file management. Web security measures: authentication, encryption (SSL), and data protection. Introduction to content management systems (CMS) like WordPress. Version control using Git.

Labs / Practicals:

1. Create a simple web page with your name, photo, and contact details.
2. Design a student registration form using different input controls (text, radio, checkbox, etc.).
3. Create an HTML page using tables to display a college timetable.
4. Create a web page with internal and external linking.
5. Write a HTML program to design a form which should allow to enter your personal data (Hint: make use of text field, password field, e-mail, lists, radio buttons, checkboxes, submit button).
6. Write HTML Code to demonstrate the use of Anchor Tag for the Following:-

a. Creating a web link that opens in a new window.
b. Creating a web link that opens in the same window
c. Reference within the same html document.
d. Reference to some image.
e. Making an image a hyperlink to display second image.

7. Create an html page with following specifications. Title should be about my City, Place your City name at the top of the page in large text and in blue color. Add names of landmarks in your city each in a different color, style and typeface. One of the landmark, your college name should be blinking. Add scrolling text with a message of your choice.
8. Create an html page with 7 separate lines in different colors. State color of each line in its text.
9. Design a webpage with a navigation bar using lists.
10. Apply inline, internal, and external CSS to format text and layout.
11. Create a webpage with a fixed header and footer using CSS.
12. Design a responsive layout using Flexbox or Grid.
13. Create a webpage with hover effects on buttons and menus.
14. Build a styled form using CSS (with focus and validation effects).
15. Write a JavaScript program to validate a user login form.
16. Write a JavaScript program to display the current date and time.
17. Create a simple calculator using JavaScript.
18. Write a JavaScript program for form input validation (e.g., email, number).
19. Use JavaScript to change the content and style of an HTML element dynamically.
20. Implement an image slideshow or gallery using JavaScript.
21. Write a PHP script to display your name and current date/time.
22. Create a simple PHP calculator (Add, Subtract, Multiply, Divide).
23. Write a PHP script to process and display form data (POST/GET).
24. Write a PHP program to connect to MySQL database and display data.

References:

1. Web Technologies by Uttam K. Roy ,Oxford University Press
2. Web Technology: Theory and Practice by M. Srinivasan, Pearson Education
3. Internet and Web Technologies by Rajkamal, Tata McGraw Hill
4. Web Programming using HTML, CSS, JavaScript & PHP by Dr. Neeraj Kumar, Neeraj Publications

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
DCSA250011	Computer Oriented Numerical Methods	3	1	0	4

Course Outcomes:

CO1: Introduce fundamental numerical techniques used to solve real-world computational problems.
CO2: Provide a solid foundation for applying numerical methods in practical domains.
CO3: Enhance students' ability to understand and analyze mathematical models in applied computing.
CO4: Prepare students to handle numerical aspects such as mobile communications, pattern recognition etc.

Module 1:

Computer Arithmetic:
Number System, Conversion of Numbers, Representation of numbers, Floating point representation, Arithmetic operations with Normalized Floating point Numbers, consequences of normalization, pitfalls in computing.

Module 2:

Roots of Nonlinear Equations:
Introduction, Methods of Solution, Iterative Methods, Bisection method, False position method,
Netwon-Raphson method, Secant method, Rate of convergence of iterative methods.

Module 3:

Solution of simultaneous algebraic Equations:
Gauss elimination method, Pivoting, ill-conditioned systems, Gauss-Seidel iterative method, Convergence of Iteration methods.

Module 4:

Polynomial Interpolation:
Introduction, Polynomial Forms, Linear interpolation, Lagrange interpolation, Newton interpolation, Difference table, Forward and backward difference table.

Module 5:

Numerical Integration:
Trapezoidal Rule, Simpson’s 1/3 rule, Simpson’s 3/8 rule.
Numerical Solution of Ordinary Differential Equations:
Taylor Series Method, Euler’s method, Second and Fourth order Runga-Kutta methods.

Labs / Practicals:

1. Root Finding using Bisection and Newton-Raphson Methods.
2. Solving Linear Systems using Gauss Elimination and Gauss-Jordan Methods.
3. Interpolation using Newton’s Forward and Backward Difference Methods.
4. Numerical Integration using Trapezoidal and Simpson’s Rules.
5. Solving Ordinary Differential Equations using Euler’s and Runge-Kutta Methods using : Initial value problems, Step size sensitivity and Stability and accuracy.

References:

1. Balagurusamy, E., Numerical Methods, Tata McGraw Hill, 1999.
2. Rajaraman V., Computer Oriented Numerical Methods, 3rd Edition, Prentice Hall India, New Delhi, 1998.
3. Stoor, Bullrich, Computer Oriented Numerical Methods, Springer-Verlag, 1998.

Course Code	Course Name	L	T	P	Cr
DCSE250042	Introduction to Operating System	3	1	0	4

Course Outcomes:

CO1: Explain types of operating systems, their structures, services, and system calls in different computing environments.
CO2: Analyze process management concepts including scheduling algorithms, multithreading, synchronization, and deadlock handling.
CO3: Apply memory management techniques such as paging, segmentation, virtual memory, and page replacement algorithms.
CO4: Demonstrate file and I/O management strategies including directory structures, access methods, device drivers, and disk scheduling.
CO5: Explain file management, protection and security mechanisms including authentication, authorization, and policy enforcement in operating systems.

Module 1:

Introduction:
Operating systems (OS) definition, Multiprogramming and Time Sharing operating systems, real time OS, Multiprocessor operating systems, Multicore operating systems, various computing environments.

Module 2:

Operating System Structures:
Operating Systems services, System calls and System programs, operating system architecture (Micro Kernel, client server) operating.

Module 3:

Process Management:
Process concept, Operation on processes, Multi-threaded processes and models, Multicore systems, Process scheduling algorithms, Process synchronization. The Critical-section problem and deadlock characterization, deadlock handling.

Module 4:

Memory Management:
Physical and Logical address space; Memory allocation strategies - Fixed and Variable Partitions, Paging, Segmentation, Demand Paging and virtual memory, Page Replacement algorithm.

Module 5:

File, I/O, Device Management, and Security:
File and directory management including directory structures, file access methods, and file servers; I/O and device management concepts such as device drivers and disk scheduling algorithms; along with protection mechanisms, authentication and authorization techniques, and policy enforcement for system security.

Labs / Practicals:

NA

References:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2008). Operating Systems Concepts. 8th edition.. John Wiley Publications.
2. Dhamdhere, D. M. (2006). Operating Systems: A Concept-based Approach. 2nd edition. Tata McGraw-Hill Education.
3. Stallings, W. (2018). Operating Systems: Internals and Design Principles. 9th edition. Pearson Education.

4. Tanenbaum, A. S. (2007). Modern Operating Systems. 3rd edition. Pearson Education.

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
DOAI250032	Machine Learning	3	0	2	4

Course Outcomes:

CO1: Identify and extract relevant features from raw IoT data suitable for applying appropriate Machine Learning (ML) techniques.
CO2: Evaluate and compare the strengths and limitations of various ML algorithms across diverse application domains, particularly in IoT.
CO3: Select and implement appropriate ML models based on data characteristics and application requirements.
CO4: Mathematically formulate and analyze ML models and algorithms, including their underlying assumptions and performance metrics.
CO5: Design and validate machine learning-based solutions for real-world problems in IoT, demonstrating practical understanding and technical proficiency.

Module 1:

Module 1
Introduction: Introduction to Machine Learning: Introduction. Different types of learning, Hypothesis space and inductive bias, Evaluation. Training and test sets, cross validation, Concept of over fitting, under fitting, Bias and Variance.
Linear Regression: Introduction, Linear regression, Simple and Multiple Linear regression, Polynomial regression, evaluating regression fit.

Module 2:

Module 2
Decision tree learning: Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning, Python exercise on Decision Tree.
Instance based Learning: K nearest neighbor, the Curse of Dimensionality, Feature Selection: forward search, backward search, univariate, multivariate feature selection approach, Feature reduction (Principal Component Analysis), Python exercise on KNN and PCA. Recommender System: Content based system, Collaborative filtering based.

Module 3:

Probability and Bayes Learning: Bayesian Learning, Naïve Bayes, Python exercise on Naïve Bayes, Logistic Regression.
Support Vector Machine: Introduction, the Dual formulation, Maximum margin with noise, nonlinear SVM and Kernel function, solution to dual problem.

Module 4:

Artificial Neural Networks: Introduction, Biological motivation, ANN representation, appropriate problem for ANN learning, Perceptron, multilayer networks and the back propagation algorithm.

Module 5:

Ensembles: Introduction, Bagging and boosting, Random forest, Discussion on some research papers.
Clustering: Introduction, K-mean clustering, agglomerative hierarchical clustering, Python exercise on k-mean clustering.

Labs / Practicals:

1 Write a Python program that imports a dataset of your choice, displays the first five rows, and prints the data type of the DataFrame as a whole. Then, generate and display the statistical summary (mean, median, standard deviation, etc.) for each column and provide complete information about the DataFrame.

2 Write a Python program to import (or create) a dataset with missing values, display the count of missing values per column, and handle them using both imputation and dropping methods. Then, compare the dataset before and after handling missing values, and display the final cleaned dataset.
3 Write a Python program to load (or create) a dataset, split it into training and testing sets, and standardize the features using StandardScaler from sklearn. preprocessing. Finally, display the dataset before and after standardization to observe the changes.
4 Write a Python program to load (or create) a dataset, separate it into predictor variables (X) and target variable (y), and then split the dataset into training and testing sets. Finally, print both the training and testing datasets.
5 Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Python ML library classes in the program.
6 Write a Python program to load (or create) a dataset with missing values, count missing values in each column, and handle them by replacing with NaN, the column mean, and the most frequent value. Finally, display the dataset before and after performing the replacements.
7 Write a Python program to load (or create) a dataset, split it into training and testing sets, and apply feature scaling to the data. Finally, display the dataset before and after scaling to observe the changes.
8 Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
9 Implement the working principle of Artificial Neural Network with feed forward and feed backward principle.
10 Implement and demonstrate the working of SVM algorithm for classification.

References:

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
2. C. M. Bishop. Pattern Recognition and Machine Learning. First Edition. Springer, 2006. (Second Indian Reprint, 2015).
3. Tom M. Mitchell, “Machine Learning”, McGraw-Hill, 2010
4. P. Flach. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. First Edition, Cambridge University Press, 2012.

Course Code	Course Name	L	T	P	Cr
DCSE250029	Computer Graphics	3	1	0	4

Course Outcomes:

CO1: Understand the basics of computer graphics systems and their architecture. Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
CO2: Apply geometric transformations and model graphical objects using OpenGL.
CO3: Develop interactive applications with input and event handling.
CO4: Illustrate concepts of 3D viewing, projections, and shading models.
CO5: Implement algorithms for clipping, rasterization, and rendering pipeline stages.

Module 1:

Introduction to Computer Graphics and OpenGL
Introduction to Computer Graphics: Applications, physical vs. synthetic images,
Graphics Systems: Imaging systems, synthetic camera model, Graphics Architectures: Programmable pipelines, performance metrics, Graphics Programming: Sierpinski gasket, 2D applications, OpenGL Basics: OpenGL API, primitives, attributes, colors Viewing & Control: Viewing, control functions, recursive graphics, 3D Gasket & Implicit Functions

Module 2:

Interaction and Input Mechanisms
Interaction & Input Devices: Event handling, input models, Client-Server Architecture, Display Lists and Modeling: Reusability, efficiency, Event Driven Programming: Menus, picking, Building Interactive Models: CAD systems, Animation and Logic Operations

Module 3:

Geometric Transformations and Modeling
Geometric Objects: Scalars, points, vectors, Three-dimensional Primitives; Coordinate Systems and Frames; Modeling a Colored Cube; Affine Transformations; Rotation, Translation and Scaling; Geometric Objects and Transformations; Transformation in Homogeneous Coordinates; Concatenation of Transformations; OpenGL Transformation Matrices; Interfaces to 3D applications; Quaternion's.

Module 4:

Viewing and Shading
Classical and computer viewing; Viewing with a Computer; Positioning of the camera; Simple projections; Projections in OpenGL; Hidden- surface removal; Interactive Mesh Displays; Parallel-projection matrices; Perspective-projection matrices; Projections and Shadows. Light and Matter; Light Sources; The Phong Lighting model; Computation of vectors; Polygonal Shading; Approximation of a sphere by recursive subdivisions; Light sources in OpenGL; Specification of materials in OpenGL; Shading of the sphere model; Global Illumination.

Module 5:

Implementation Techniques and Rendering Pipeline
Basic Implementation Strategies; Four major tasks; Clipping; Line-segment clipping; Polygon clipping; Clipping of other primitives; Clipping in three dimensions; Rasterization; Bresenham's algorithm; Polygon Rasterization; Hidden-surface removal; Antialiasing; Display considerations.

Labs / Practicals:

NA

References:

1.Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th Edition, Pearson Education, 2008.
2.Donald Hearn and Pauline Baker: Computer Graphics OpenGL Version 3th Edition, Pearson Education, 2004.
3.F.S. Hill Jr.: Computer Graphics Using OpenGL, 3th Edition, PHI, 2909.
4.James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Computer Graphics, Pearson Education 1997.

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
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
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
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
DOAI260041	Natural Language Processing	2	0	4	4

Course Outcomes:

CO1. Develop the ability to collect, preprocess, and analyze textual data using NLP techniques and Python libraries
CO2. Apply text representation techniques and machine learning algorithms to build, train, and evaluate NLP classification models using appropriate performance metrics.
CO3. Develop and apply advanced NLP techniques for information extraction, text summarization, question answering, sentiment analysis, and topic modelling in real-world applications.
CO4. Design, develop, and deploy intelligent chatbot systems using rule-based and neural network-based NLP techniques following the AI project development cycle.
CO5. Evaluate and apply modern NLP concepts including Large Language Models, prompt engineering, ethical AI practices, and model interpretability techniques for responsible deployment of NLP solutions

Module 1:

Introduction to NLP, Data Collection and Processing for NLP – definition and current applications of Natural Language Processing, exploration of text data, text preprocessing pipeline – sentence segmentation, tokenization, stop word removal, stemming, lemmatization, part-of-speech tagging, handling and cleaning raw text data in Python.

Module 2:

Classification for NLP – Bag of Words (BoW) representation of text data, TF-IDF (Term Frequency-Inverse Document Frequency) format, Introduction to word embeddings. Model building for text classification – Naive Bayes classifier, Logistic Regression classifier, training and evaluating the classification model, performance metrics – precision, recall, F1-score, confusion matrix. Roadmap from MLP to LSTM and LSTM to Transformers. Attention Mechanism, Transformer architecture (BERT and GPT). Transformer-based text representation using BERT.

Module 3:

Advanced NLP Tasks and Applications - Named Entity Recognition (NER) and Relation Extraction, Text Summarization: Extractive and Abstractive Approaches, Advanced Question Answering Systems (Open-Domain, Generative QA), Sentiment Analysis and Opinion Mining, Topic Modelling. Applications of BERT in NER, sentiment analysis, and question answering, Coreference resolution, Information extraction.

Module 4:

Chatbot architecture, cosine similarity for rule-based chatbots, neural network-based intent recognition, chatbot training, and building/testing chatbots in Python. Application of the AI Project Cycle for chatbot development, including problem scoping, Data Preprocessing, Classification Model Development, and Evaluation.

Module 5:

Current Topics, Ethics, and Deployment, Large Language Models (LLMs), Capabilities and Prompt Engineering, Bias, Fairness, and Ethical Considerations in NLP Models, Model Debugging, NLP Interpretability and Explainability.

Labs / Practicals:

Explore text data that has been downloaded and processed from a given corpus.
2. Process textual data by sentence segmentation, tokenization, and stop word removal using NLTK/spaCy in Python.
3. Express data in a bag of words format and use the TFIDF format to represent text features.
4. Construct a separate dataset for model training, including train-test split and cross-validation.
5. Construct and train a text classification model using Naive Bayes or Logistic Regression in Python.

6. Evaluate the classification model using precision, recall, F1-score, and confusion matrix.
7. Implement Named Entity Recognition and relation extraction for identifying entities and relationships in text.
8. Perform text summarization using extractive and abstractive techniques, and compare the generated summaries.
9. Develop NLP applications for sentiment analysis, topic modelling, and question answering using sample datasets.
10. Use cosine similarity to build and train a simple rule-based Chatbot.
11. Use a neural network to train a more advanced Chatbot capable of intent recognition.
12. Apply the AI Project Cycle for the Hotel Chatbot problem: data collection, pre-processing, model training, and deployment.
13. Apply AI Project Cycle to develop a complete AI solution for an NLP project of choice, documenting all stages from data collection to evaluation.
14. Explore LLM capabilities using prompt engineering for NLP tasks and analyze ethical issues such as bias and fairness.
15. Apply SHAP and LIME for NLP model interpretability, debugging, and ethical analysis.

References:

Intel AI for Future Workforce – Natural Language Processing Course Content, Intel Digital Readiness Program.
2. Steven Bird, Ewan Klein, and Edward Loper – Natural Language Processing with Python, O'Reilly Media.
3. Delip Rao and Brian McMahan – Natural Language Processing with PyTorch, O'Reilly Media.
4. Yoav Goldberg – Neural Network Methods for Natural Language Processing, Morgan and Claypool Publishers.
5. Jacob Devlin et al. – BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI.
6. Practical Natural Language Processing, Sowmya Vajjala et al. O'Reilly
7. "Attention Is All You Need", 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Course Code	Course Name	L	T	P	Cr
DOAI250014	Data Analysis using Python	2	0	4	4

Course Outcomes:

CO1: Understand and analysis libraries such as Pandas, NumPy, and SciPy to manipulate and analyze structured data.
CO2: Import and export data using Python packages.
CO3: Perform data preprocessing tasks including handling missing values, data cleaning, and applying transformation techniques to prepare datasets for analysis.
CO4. Apply statistical methods for data exploration.
CO5. Create and export data visualizations.

Module 1:

Module 1:
Introduction to Pandas, NumPy, SciPy:Introduction to Pandas DataFrames, Numpy multidimentional arrays, and SciPy libraries to work with different datasets

Module 2:

Module 2:
Import and Export of Data:Installing, loading and using packages for importing and exporting data in Python.

Module 3:

Module 3:
Data Preprocessing and Transformation: Handling of missing data, Data cleaning and Transformation.

Module 4:

Module 4:
Data Exploration: Exploring data using statistical methods: mean, median, mode1, quantiles. Building contingency table 2. Basics of grouping data and Correlation.

Module 5:

Module 5:
Data Visualization: Scatter Plot, line graph, histogram, boxplot, line plots regression, word clouds2, exporting plots as images.

Labs / Practicals:

1.In a meeting of the marketing department, it has been decided that the price of selling an item is fixed at Rs. 50. The team resolved to increase sales and achieve a profit of Rs. 60,000. Use Goal Seek to find out how many items you will need to sell to meet your profit target.
2.Create a worksheet related to crop production of various crops in Indian states for the last five years (wheat, rice, pulses, soybean, and sugarcane).
i)Create a bar chart to visualize the data.
ii)Create a pie chart for crop production distribution.
iii)Create a box plot to analyze the crop production trends.
3.Study and perform various DAX functions to analyze sales and production data. Create a sample data set and perform the following operations:
i)Use SUM, AVERAGE, COUNT, and other DAX functions.
ii)Create a PivotTable to summarize data based on categories.
iii)Use DAX to calculate total revenue, profit, and margin.
4.Create a workbook related to the sales of a business company, showing various products sold over the last four

quarters for 10 salespersons.
i)Create a line graph to show growth/decline in sales over the quarters.
ii)Display the sales figures of each salesperson using a separate graph.
iii)Identify the top two salespersons in the last two quarters.
iv)Find the salespersons who were consistent over the last four quarters.
v)Determine the most popular product and the current top-selling product.
5.Create a PivotTable to show the customer names who placed orders with GSS during 2019-2022.For each customer, also show the total number of orders, total sales, and total profit. Add a slicer or filter that can be used to show information specifically for each customer segment.
Use the PivotTable to answer these questions:
i)Which small business customer had the highest sales?
ii)Which corporate customer placed the greatest number of orders in 2019-2022?
iii)Which consumer customer was the most profitable?
iv)What was the sales figure of the least profitable home office customer?
6.Create a worksheet for employee salaries with the following details:
i)Full Name, Grade (1/2/3), Basic Salary, HRA, PF, Gross Salary, Net Salary, VA, VA > HRA. HRA is calculated as follows:
ii)Grade 1: 40%, Grade 2: 35%, Grade 3: 30%.
iii)PF is 8% for all grades.
iv)VA is 15,000, 10,000, and 7,000 for Grades 1, 2, and 3, respectively.
v)Calculate the maximum, minimum, and average salary for each grade.
vi)Count the number of people where VA > HRA.
vii)Find the most frequently occurring grade

References:

1.Mckinney, W. (2017). Python for Data Analysis. Second edition, O'reilly (SPD).
Additional Resources
1.Grus, J. (2016). Data Science from scratch. First edition, O'reilly (SPD).
2.VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. Second edition, O'reilly (SPD).

Course Code	Course Name	L	T	P	Cr
DOAI250024	Deep Learning	2	0	4	4

Course Outcomes:

CO1: Understand the key concepts and training techniques of neural networks.
CO2: Apply deep learning models like CNNs and RNNs to solve real-world problems.
CO3: Gain expertise in advanced models like transformers and generative models for diverse applications.

Module 1:

Introduction to Neural Networks: Basics of Artificial Intelligence and Machine Learning, Introduction to Neural Networks and Biological Inspiration, Perceptron Model and Multilayer Perceptrons (MLP), Activation Functions (Sigmoid, ReLU, Tanh, Softmax), Gradient Descent and Backpropagation

Module 2:

Deep Feedforward Networks: Deep Learning vs. Traditional Machine Learning, Architecture of Feedforward Neural Networks (FNN), Weight Initialization Techniques, Optimization Techniques (SGD, Adam, RMSprop), Vanishing and Exploding Gradient Problems
Convolutional Neural Networks (CNNs): Introduction to Convolutional Neural Networks, Convolutional Layers, Pooling, and Padding, Popular Architectures (LeNet, AlexNet), Applications of CNNs in Image Processing

Module 3:

Recurrent Neural Networks (RNNs) and LSTMs: Introduction to Sequential Data and RNNs, Backpropagation Through Time (BPTT), Vanishing Gradient Problem in RNNs, Long Short-Term Memory (LSTMs) and Gated Recurrent Units (GRUs).

Module 4:

Generative Models and Autoencoders: Introduction to Generative Models, Autoencoders and Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Applications in Image Generation and Data Augmentation

Module 5:

Generative Models and Autoencoders: Introduction to Attention Mechanisms, Transformer Architecture, Self-Attention and Multi-Head Attention, BERT, GPT, and Large Language Models, Applications in NLP and Computer Vision

Labs / Practicals:

Preparing

References:

1. Charu C. Aggarwal, “ Neural Networks and Deep Learning”, Springer
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press
3. Rowel Atienza , “Advanced Deep Learning with TensorFlow 2 and Keras”, Packt Publishing
4. Lewis Tunstall, Leandro von Werra, Thomas Wolf, "Natural Language Processing with Transformers"
5. "Deep Learning for Natural Language Processing" – Palash Goyal, Sumit Pandey, Karan Jain
6. "Fundamentals of Deep Learning" – Nikhil Buduma, Nicholas Locascio

Course Code	Course Name	L	T	P	Cr
DCSE250044	Cyber Security	3	0	2	4

Course Outcomes:

CO1: Explain fundamentals of cybersecurity, security goals, common threats, attacks, malware, cybercrime, and legal frameworks.
CO2: Apply cryptographic techniques including classical ciphers, modern encryption algorithms, and hashing functions for securing information.
CO3: Analyze network security concepts such as firewalls, IDS, VPNs, and wireless security, and simulate basic attacks and defenses.
CO4: Demonstrate system and application security through OS hardening, malware analysis, and secure coding practices against web-based attacks.
CO5: Use cybersecurity tools (e.g., Wireshark, Nmap, Burp Suite, Metasploit) and prepare incident/case study reports to evaluate real-world cyber threats.

Module 1:

Introduction to Cyber Security:
Basics of Information Security, CIA Triad: Confidentiality, Integrity, Availability, Types of Threats and Attacks, Malware: Virus, Worm, Trojan, Ransomware, Phishing, Social Engineering, Security Goals and Challenges, Cybercrime and Cyber Laws (IT Act 2000).

Module 2:

Cryptography Fundamentals :
History and Evolution of Cryptography, Symmetric vs Asymmetric Encryption, Classical Ciphers: Caesar, Vigenère, Playfair, Modern Cryptography: AES, DES, RSA, Hash Functions: MD5, SHA-1, SHA-256.

Module 3:

Network Security :
Basics of Computer Networks, Network Threats: DoS, DDoS, IP Spoofing, Sniffing, Firewalls and Intrusion Detection Systems (IDS), VPNs and Secure Tunneling, Wireless Security (WEP, WPA2).

Module 4:

System and Application Security:
OS Security: Windows and Linux, Secure Configuration and Hardening, Malware Analysis Basics, Web Application Security: SQL Injection, Cross-site Scripting (XSS), Cross-site Request Forgery (CSRF), Secure Coding Practices.

Module 5:

Cyber Security Tools and Case Studies:
Security Tools: Nmap, Metasploit, Burp Suite, Digital Forensics Basics, Security Audit Process, Ethical Hacking Overview, Recent Cybersecurity Case Studies.

Labs / Practicals:

Module 1: Introduction to Cyber Security
1. Simulate a Phishing Attack Detection:
Identify and report phishing emails from a set of real-looking email samples. Explain the signs you observed.
2. Analyze a Cybercrime Case Study:
Select a real-world cyber attack (e.g., Facebook Data Leak, WannaCry) and create a report explaining the type of attack, vulnerabilities exploited, and impact.
Module 2: Cryptography Fundamentals
3. Implement Caesar Cipher in Python:
Write a program to encrypt and decrypt a given message using Caesar Cipher with a shift key.

4. Implement Vigenère Cipher in Python:
Create a script to encode and decode text using a Vigenère Cipher.
5. Basic RSA Encryption-Decryption Program:
Simulate key generation, encryption, and decryption using basic RSA algorithm (use Python or any language).
6. Generate Hashes of a File:
Use command-line tools or a Python script to generate MD5 and SHA256 hashes of a file and verify integrity.
Module 3: Network Security
7. Packet Analysis using Wireshark:
Capture live network traffic and analyze packets to identify IP addresses, protocols, and suspicious patterns.
8. Scan Network using Nmap:
Use Nmap to scan a local network or localhost and generate a report on open ports and running services.
9. Firewall Rule Configuration (Windows/Linux):
Configure basic firewall rules to block/allow specific IP addresses or ports and verify the effect.
10. Simulate a Ping Flood DoS Attack (in test environment):
Use tools like hping3 or ping with large packet sizes to simulate a ping flood and monitor the system/network behavior.
Module 4: System and Application Security
11. Perform SQL Injection on a Test Web App (DVWA):
Use DVWA (Damn Vulnerable Web App) or bWAPP to perform SQL injection and extract data.
12. Cross-site Scripting (XSS) Demonstration:
Identify and exploit an XSS vulnerability in a test environment and demonstrate how input sanitization can fix it.
13. System Hardening Checklist (Windows/Linux):
Apply at least 5 system hardening techniques to your OS (disable services, create strong passwords, enable firewall) and document them.
Module 5: Tools and Case Studies
14. Use Burp Suite to Intercept HTTP Traffic:
Use Burp Suite to intercept, analyze, and modify HTTP requests/responses between browser and web app.
15. Create a Cybersecurity Incident Report:
Prepare a structured report analyzing a recent cyber attack (use open data or news) including type of attack, tools used, impact, and lessons learned.

References:

1 William Stallings – Cryptography and Network Security
2 Nina Godbole – Cyber Security: Understanding Cyber Crimes
3 Online platforms: Cybrary, OWASP, TryHackMe, Hacker101

Course Code	Course Name	L	T	P	Cr
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4

Course Outcomes:

CO1. Understand and be able to define and explain the terms 'AI for good' and 'responsible AI'. Describe the significance of responsible AI and recognize the potential benefits of AI in society.
CO2. Discuss and examine the impact of ethics on AI decision-making processes. Identify common ethical pitfalls in AI development and deployment. Explore methods for ethical AI practices.
CO3. Understand to apply an ethics-first approach to AI project planning and development. Compare different approaches to ethical AI design which include design of rules based on ethics and virtues. Identify factors influencing ethical decisions in AI development.
CO4. Apply Utilitarian and Kantian ethical principles to AI case studies, scenarios, and dilemmas. Explain the purpose and benefits of AI ethics codes. Develop a code of ethics for AI within an organization.
CO5. Identify the importance of interpretability in AI models. Evaluate the strengths and weaknesses of three different explainable AI tools: SHAP, LIME, and Tree Interpreter.
CO6. Conduct impact assessments for AI solutions. Apply ethical considerations in designing an AI system or solution. Implement measures to mitigate ethical risks and challenges post deployment.

Module 1:

Introduction to AI Ethics – meaning and importance, Role of Ethics in Responsible AI – defining 'AI for good' and 'responsible AI', significance of responsible AI in society, Common Ethical Pitfalls in AI – bias, fairness, accountability, transparency issues in AI development and deployment, Principles of Ethical AI – core principles guiding ethical AI, methods for ethical AI practices in development and deployment.

Module 2:

Ethics-First Approach in the AI Project Cycle – applying ethics at every stage of AI project planning and development, Approaches for Designing Ethical AI – rules-based ethics, values-based design, Introduction to Ethical Frameworks – overview of major ethical frameworks applied to AI, Virtue-Based Ethics – virtue ethics principles and the 4-box method for analysing ethical dilemmas, Applying Bio-Ethics Framework in AI – bioethical principles applied to AI in healthcare and biotechnology.

Module 3:

Utilitarianism for Ethical AI – utilitarian principles applied to AI case studies and dilemmas, Kantianism for Ethical AI – Kantian ethical principles applied to AI scenarios, Code of AI Ethics for Organizations – purpose, benefits and development of an organizational AI code of ethics, Data Protection and AI – risks of data breaches, privacy violations, responsible data handling.

Module 4:

Explainable AI – importance of interpretability in AI models, Intel XAI Toolkit, impact of explainability on decision-making and trust, Explainable AI in Practice – SHAP, LIME and Tree Interpreter tools: strengths, weaknesses and methods for applying them to analyze and interpret AI applications.

Module 5:

Impact Assessment of AI Solutions – conducting impact assessments for AI systems, Mitigating Ethical Issues Post Deployment – strategies to identify and resolve ethical risks after deployment, Project Creation – creating an AI startup with data card, model card, code of ethics, explainability report, impact self-assessment and transparency report.

Labs / Practicals:

1. Develop comprehensive Data Cards for datasets, as outlined in the Data Cards Playbook by Google, to ensure transparency and accountability in AI training data.
2. Design Responsible AI Best Practices.

3. Apply Human-Centered AI Canvas to design AI systems that address user needs and ethical considerations.
4. Learn to use The Box to embed AI principles into organizational workflows and operations.
5. Apply bio-ethics principles to AI case studies, scenarios, and dilemmas; evaluate the impact of virtue-based ethics on AI systems.
6. Develop AI ethics cards for given AI scenarios. Analyse a variety of ethical dilemmas by following the 4-box method from Virtue Ethics.
7. Analyse a variety of ethical dilemmas by walking through the ethical frameworks: Virtue Ethics, Bioethics, Utilitarian, Kantian, and Moral System Agnostic.
8. Develop an AI Code of Ethics for a fictitious company.
9. Use the UXAI Toolkit to brainstorm and plan different explainable interfaces for AI applications. Use the explainable AI library SHAP to create an explainability report for income prediction on the standard adult census income dataset.
10. Conduct impact assessments for AI solutions; create your own AI startup, develop business documents, write documentation (data card, model card, code of ethics, explainability report, impact self-assessment) and create and analyze transparency report.

References:

1. Intel AI for Future Workforce – Applied Ethics for AI Course Content, Intel Digital Readiness Program.
2. Virginia Eubanks – Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor, St. Martin's Press.
3. Cathy O'Neil – Weapons of Math Destruction, Crown Publishers.
4. Kate Crawford – Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence, Yale University Press.
5. Google Data Cards Playbook – https://pair-code.github.io/datacardsplaybook/

Course Code	Course Name	L	T	P	Cr
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4

Course Outcomes:

CO1. Describe what AI is and what it is not. Appreciate the history of AI and its evolution over time.
CO2. Identify and analyze current trends in AI by correlating them with other technologies like IoT, Big Data and 5G.
CO3. Identify 3 common domains of AI – Natural Language Processing, Computer Vision and Statistical Data – based on the type of underlying data.
CO4. Examine and appreciate typical steps involved in an AI Project through the AI Project Cycle.
CO5. Examine the immediate impacts of AI practices on society and appreciate the ethical concerns of AI applications.
CO6. Classify Supervised Learning, Unsupervised Learning and Reinforcement Learning with the help of examples. Discuss the roles played by different key stakeholders in a typical AI team.

Module 1:

Demystification of AI – what AI is and what it is not, History and Evolution of AI, Current trends in AI, AI and Technology convergence, Industry 4.0 and Digitalization, Role of IoT, Big Data and 5G in AI, Domains of AI – Natural Language Processing, Computer Vision and Statistical Data.

Module 2:

AI Project Cycle-I – Problem Scoping and Data Acquisition, AI Project Cycle-II – Modelling and Evaluation, Societal Impact of AI – ethical concerns and responsible practices, Elements of ML – Supervised Learning, Unsupervised Learning and Reinforcement Learning with examples.

Module 3:

Elements of AI – types and characteristics, Data as Fuel for AI – structured and unstructured data, methods of data mining and data storage, AI/Data Science Team – roles and responsibilities of key stakeholders.

Module 4:

Tools to implement AI – No-Code tools (Teachable Machine, Orange, Roboflow, Chatfuel) and Code-based tools, Introduction to Neural Networks – working of simple Neural Networks, difference between common Deep Learning models with examples and applications.

Module 5:

AI Project-I – Natural Language Processing use case using no-code tool Chatfuel, AI Project-II – Statistical Data use case using no-code tool Orange Data Mining, AI Project-III – Computer Vision use case using no-code tool Roboflow, Future Possibilities of AI – emerging software and hardware technology trends and their direct impact on development of AI.

Labs / Practicals:

1. Familiarize with AI tools and techniques, including statistical data analysis using raw graphs, computer vision applications like Vision API, and generative AI tools such as DALL·E, Stable Diffusion, Gemini, and ChatGPT.
2. Apply the AI Project Cycle to real-world scenarios by identifying problems, collecting data, developing models, and evaluating their effectiveness to propose impactful AI-based solutions.
3. Deconstruct the working behind simple Neural Networks and outline the difference between some common DL models with the help of examples and applications.
4. Explore and utilize no-code tools, such as Teachable Machine, to build AI projects.
5. Project Building using No-code tool – Orange Data Mining for Statistical Data Use cases.
6. Project Building using No-code tool – Roboflow for Computer Vision Use cases.
7. Project Building using No-code tool – Chatfuel for Natural Language Processing Use cases.
8. Classify the type of data into structured and unstructured and evaluate data quality using real datasets.

9. Explore various AI domains by running Vision API experiments and comparing outcomes from NLP, CV, and Statistical Data use cases.
10. Demonstrate the AI Project Cycle on a chosen real-world problem from an industrial or social impact perspective and present findings.

References:

1. Intel AI for Future Workforce – Introduction to AI Course Content, Intel Digital Readiness Program.
2. Stuart Russell and Peter Norvig – Artificial Intelligence: A Modern Approach, Pearson Education.
3. Andreas Mueller and Sarah Guido – Introduction to Machine Learning with Python, O'Reilly Media.
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Deep Learning, MIT Press.
5. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.

Course Code	Course Name	L	T	P	Cr
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4

Course Outcomes:

CO1. Define Gen AI and distinguish its role in the broader context of AI. Describe how the basic model of Gen AI works. Discuss the wider societal impact of Gen AI.
CO2. Understand how NLP has evolved over time and the shift from Traditional AI to Gen AI. Understand Prompt Engineering and its impact on the output of NLP tasks.
CO3. Learn the limitations of current AI generation tools for video and audio. Recognize the challenges in ethical considerations and responsibilities when using generative AI tools.
CO4. Describe the stages of a generative AI project and learn to select and use appropriate AI tools for development.
CO5. Understand the basic structure of neural networks, architecture of transformer models, workings of diffusion models, and the pre-training process for generative AI models.
CO6. Learn to fine-tune generative AI models, understand RLHF, prototype Gen AI applications, and apply principles of responsible and ethical use of LLMs.

Module 1:

Introduction to Gen AI – definition and scope, role of Gen AI in the broader context of AI, how the basic model of Gen AI works, societal impact of Gen AI, Demystifying Gen AI Models – types of generative models (GANs, VAEs, Diffusion Models, LLMs), evolution of NLP to Gen AI, shift from Traditional AI to Gen AI.

Module 2:

Prompt Engineering-I – introduction to prompt engineering, basic applications and impact on NLP task outputs, text generation and conversation initiation, Prompt Engineering-II – advanced prompt engineering techniques for diverse NLP applications, real-world examples such as Mint Mobile Commercial, Basics of Video and Audio for Generation using Gen AI for Productivity-I – introduction to video generation tools (Runway ML, Genmo), text-to-video and image-to-video generation, limitations of current AI generation tools for video and audio.

Module 3:

Basics of Video and Audio for Generation using Gen AI for Productivity-II – audio and music generation using AIVA, code generation using Codeium, text generation using ChatGPT, Gemini and Perplexity, Generative AI-Ethics – ethical considerations, responsibilities and challenges when using generative AI tools, Generative AI Project Lifecycle – stages of a Gen AI project, selection of appropriate AI tools, Gen AI Tools for Development – open-source tools such as Stable Diffusion XL, proprietary tools such as Bing Image Creator powered by DALLE3.

Module 4:

Neural Network Fundamentals – basic structure and functioning of neural networks, how neural networks process data to make predictions, Transformers and the Evaluation of LLMs – architecture of transformer models, methods to evaluate large language models for performance and reliability, walkthrough of OpenAI Playground and parameter exploration, Diffusion Models and the Evaluation of Text to Image Generation – workings of diffusion models, evaluation of quality of text-to-image generation, Pre-Training Gen AI Models-I – pre-training process for generative AI models, importance of pre-training in building effective AI systems, modifying classification labels of Cohere datasets and its impact on fine-tuning and model confidence.

Module 5:

Fine-Tuning Gen AI Models – techniques for fine-tuning generative AI models to adapt them for specific tasks and applications, RLHF – Evaluating and Optimizing Gen AI Models: reinforcement learning with human feedback, evaluation and improvement of performance of generative AI models, Prototyping Gen AI Applications – skills to design, develop and test prototypes of generative AI applications, LangChain components – LLM, Agents, Prompts, Chains, Document loaders and utils with code-based implementation, LLM Application and Responsible use of AI – practical applications of large language models, ethical and responsible use of LLMs, building an AI

Tutor using open-source models such as Dolly 3B and Intel Neural Chat 7B.

Labs / Practicals:

1. Apply prompt engineering skills in basic NLP tasks, such as text generation, conversation initiation, and simple problem-solving.
2. Develop advanced prompt engineering skills for diverse NLP applications. Examine real-world prompt engineering examples, such as the Mint Mobile Commercial, to discern its role in marketing.
3. Explore cutting-edge Gen AI tools, their features, and capabilities for content generation including video and audio.
4. Explore open-source models such as Stable Diffusion XL and proprietary tools such as Bing Image Creator (powered by DALLE3) for image generation.
5. Text-to-video generation and image-to-video generation using tools like Runway ML and Genmo. Generate audio and music files using the tool AIVA.
6. Generate code by prompts using the tool Codeium. Generate text using tools like ChatGPT, Gemini, and Perplexity.
7. Walkthrough of OpenAI Playground. Explore how changing parameters can affect responses generated by different models.
8. Explore how modifying classification labels of Cohere datasets impacts fine-tuning and model confidence.
9. Explore basic components of LangChain like LLM, Agents, Prompts, Chains, and Document loaders and utils, along with a code-based implementation.
10. Utilize open-source text generation models such as Dolly 3B and Intel Neural Chat 7B for building an AI Tutor application.

References:

1. Intel AI for Future Workforce – Introduction to Generative AI Course Content, Intel Digital Readiness Program.
2. Antony Hagiwara – Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media.
3. Ashish Vaswani et al. – Attention is All You Need, Advances in Neural Information Processing Systems.
4. OpenAI – GPT-4 Technical Report, OpenAI.
5. Harrison Chase – LangChain Documentation – https://docs.langchain.com/

Course Code	Course Name	L	T	P	Cr
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4

Course Outcomes:

CO1. Describe Machine Learning and how Deep Learning is a subset of the wider field of Machine Learning.
CO2. Understand the importance of dashboards and discuss various applications of dashboards across different industries.
CO3. Explain the mathematical working behind different ML models. Implement different classification and regression ML models using Python.
CO4. Outline the difference between supervised, unsupervised and reinforcement learning algorithms. Examine the working of different reinforcement learning models.
CO5. Describe the working of Neural Networks and contrast it with biological Neurons. Describe various applications of neural networks. Outline different methods to overcome variance and bias in DL models.
CO6. Discuss and interpret the future of ML based on current and upcoming trends. Develop Python-based AI Projects incorporating Supervised Learning, Unsupervised Learning and Deep Neural Networks.

Module 1:

Introduction to Machine Learning – definition, types and applications, Relationship between AI, ML and Deep Learning, Tools to Implement AI (Python & Mathematics) – I: Python basics for ML, NumPy, Pandas, data manipulation.

Module 2:

Tools to Implement AI (Python & Mathematics) – II: mathematical foundations – linear algebra, statistics and probability, No-Code tool for Data Exploration – Tableau dashboard creation, data visualization techniques, statistical concepts implementation using no-code visualization tools.

Module 3:

AI (Modeling) Supervised Learning-I – Linear Regression, Logistic Regression, mathematical working behind models, classification and regression using Python, AI (Modeling) Supervised Learning-II – SVM, Decision Tree, industrial applications of ML models, AI (Modeling) Unsupervised Learning-I – K-Means Clustering, mathematics of clustering algorithms, AI (Modeling) Unsupervised Learning-II – Recommendation Systems, mathematics behind recommendation system, applications of recommendation systems.

Module 4:

Reinforcement Learning – definition, difference from supervised and unsupervised learning, working of reinforcement learning models with applications, Neural Network and Deep Learning-I – biological vs artificial Neurons, working of Neural Networks, applications of neural networks, Deep Learning-II – Convolutional Neural Networks, Recurrent Neural Networks, methods to overcome variance and bias in DL models, implementation using Python and deep learning frameworks, exploring OpenAI's Gym for reinforcement learning.

Module 5:

ML Project – Supervised Learning AI models: end-to-end Python project using classification/regression, ML Project – Unsupervised Learning AI models: end-to-end Python project using clustering, ML Project – Deep Neural Networks: building and training DNN in Python, Future of Machine Learning and Deep Learning – current and upcoming trends in ML and DL.

Labs / Practicals:

1. Interpret and understand the basic tools required for building ML Projects through a selected list of topics from Mathematics and Python Programming.
2. Develop and create a simple dashboard for visualizing data through Tableau.
3. Implement statistical concepts using the no-code visualization tool and create your own dashboard with the help of a no-code visualization tool.

4. Implement supervised machine learning algorithms, such as SVM and Decision Tree, using Python to develop a deeper understanding of their practical applications.
5. Implement unsupervised machine learning algorithms, such as K-Means, using Python to develop a deeper understanding of their practical applications.
6. Implement neural networks and create simple generative AI models using Python and deep learning frameworks, including exploring OpenAI's Gym for hands-on experience with AI and reinforcement learning.
7. Develop Python-based use cases and AI Projects incorporating Supervised Learning methods.
8. Develop Python-based use cases and AI Projects incorporating Unsupervised Learning methods.
9. Develop Python-based use cases and AI Projects incorporating Deep Neural Networks.
10. Implement a recommendation system using Python and evaluate its performance on a real-world dataset.

References:

1. Intel AI for Future Workforce – Introduction to Machine Learning Course Content, Intel Digital Readiness Program.
2. Andreas Mueller and Sarah Guido – Introduction to Machine Learning with Python, O'Reilly Media.
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Deep Learning, MIT Press.
4. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
5. Christopher Bishop – Pattern Recognition and Machine Learning, Springer.

Course Code	Course Name	L	T	P	Cr
DCSA250017	Linux Environment	3	0	2	4

Course Outcomes:

CO1: Effectively operate and navigate a Linux-based operating system, manage files and directories, perform user and permission management, and utilize basic shell scripting to automate tasks.
CO2: Demonstrate the ability to install software, manage system processes, and configure system settings relevant to development and deployment in AI/ML environments.

Module 1:

Introduction to Linux OS:
History and evolution of Linux, Linux distributions (Ubuntu, CentOS, Debian, etc.), Linux vs Windows/Unix., Open-source philosophy and licensing (GPL).

Module 2:

Linux File system and File Operations:
Linux directory structure and navigation, File types, file permissions, and ownership,Basic file operations (ls, cp, mv, rm, cat, touch, nano, vim), Hard links vs soft links
Understanding file descriptors and redirection.

Module 3:

User and Permission Management:
User and group management (adduser, passwd, usermod), File and directory permissions (chmod, chown, chgrp), Access control and file security sudo and root privileges.

Module 4:

Process and Package Management:
Understanding processes (ps, top, kill, nice), Background and foreground jobs (&, jobs, fg, bg), Package management (APT, YUM, DPKG) ,Installing and removing software System monitoring tools.

Module 5:

Shell Scripting and Automation:
Introduction to Bash shell, Writing and executing shell scripts, Variables, loops, conditionals ,File I/O in scripts
Crontab for task scheduling, Automation of basic system admin tasks.

Labs / Practicals:

- Linux Installation (VirtualBox/WSL/Live USB)
- Navigating the File System and Directory Structure
- Creating and Managing Files and Directories
- Changing File Permissions and Ownership
- User and Group Creation and Management
- Writing Basic Shell Scripts

References:

1. Linux Command Line and Shell Scripting Bible By Richard Blum & Christine Bresnahan Great for scripting, automation, and command-line usage.

2. UNIX and Linux System Administration Handbook By Evi Nemeth et al. Industry-level Linux operations and admin tasks.

Course Code	Course Name	L	T	P	Cr
DOAI250019	Data Mining and Warehousing	3	0	2	4

Course Outcomes:

CO1: Understand and apply data warehousing concepts and architectures.

CO2: Perform clustering, classification, and prediction for pattern discovery.

CO3: Address issues related to data streams and implement stream mining techniques.

CO4: Analyze and apply web mining methodologies for real-world applications.

CO5: Explore current trends and advancements in distributed warehousing and pattern mining.

Module 1:

Data warehousing components, Building a data warehouse, Data warehouse architecture, DBMS schemas for decision support, Data extraction, cleanup, and transformation tools, Metadata, Reporting, Query tools and applications, Online Analytical Processing (OLAP), OLAP and multidimensional data analysis.

Module 2:

Data mining functionalities, Data preprocessing: cleaning, integration, transformation, reduction, discretization, Concept hierarchy generation, Architecture of typical data mining systems, Classification of data mining systems, Efficient and scalable frequent itemset mining methods, Mining various kinds of association rules, From association mining to correlation analysis, Constraint-based association mining.

Module 3:

Issues in classification and prediction, Classification methods: Decision Trees, Bayesian classification, Rule-based classification, Neural networks: backpropagation, Support Vector Machines (SVM), Associative classification, Lazy learners, Other classification methods, Prediction techniques, Accuracy and error measures, Evaluating classifiers/predictors, Ensemble methods, Model selection.

Module 4:

Types of data in clustering, Categorization of major clustering methods, Partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based clustering methods, Clustering high-dimensional data, Constraint-based clustering, Outlier analysis.

Module 5:

Mining objects, Spatial data mining, Multimedia data mining, Text mining, Mining the World Wide Web, Multidimensional analysis of complex data, Descriptive mining of complex data objects.

Labs / Practicals:

1. Design a Data Warehouse Schema

Create Star Schema and Snowflake Schema for business data (sales, e-commerce, etc.).

2. ETL Process Implementation

Perform Extract, Transform, Load (ETL) operations on structured data using SQL-based tools (MySQL / PostgreSQL).

3. Association Rule Mining

Apply the Apriori Algorithm for mining frequent itemsets and generating association rules from transactional datasets.

4. Classification Using Decision Trees

Build, visualize, and evaluate Decision Tree Classifiers on datasets like Iris or Titanic.

5. Clustering Using K-Means

Apply K-Means Clustering on multi-dimensional datasets and visualize the results.

6. Data Preprocessing & Dimensionality Reduction

Handle missing values, normalize data, and apply Principal Component Analysis (PCA).

7. OLAP Operations and Cube Construction

Perform OLAP operations (Roll-up, Drill-down, Slice, Dice) using sample datasets.

8. Web Mining Basics

Analyze web log files for pattern discovery (frequent user paths, sessions).

References:

1. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, Third Edition, Elsevier, 2011.

2. Alex Berson, Stephen J. Smith, Data Warehousing, Data Mining & OLAP, Tata McGraw-Hill, 10th Reprint, 2007.

3. K.P. Soman, Shyam Diwakar, V. Ajay, Insight into Data Mining: Theory and Practice, PHI, 2006.

4. G.K. Gupta, Introduction to Data Mining with Case Studies, PHI, 2006.

5. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education, 2007.

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
DCSE250043	Cryptography and Network Security	3	0	2	4

Course Outcomes:

CO1: Explain security fundamentals, including the CIA Triad and various cyber threats.
CO2: Describe security services and mechanisms, such as authentication and encryption.
CO3: Understand the history and evolution of cryptography, including classical techniques.
CO4: Recognize the role of cryptanalysis in breaking cryptographic systems.
CO5: Apply mathematical foundations, including number theory, modular arithmetic, and random number generation, in cryptographic algorithms.

Module 1:

Introduction to Cryptography & Security Concepts
Basics of Security, Security Goals: Confidentiality, Integrity, Availability (CIA), Threats and Attacks: Passive vs. Active Attacks, Security Services & Mechanisms, Cybersecurity Principles, Introduction to Cryptography, History & Evolution of Cryptography, Classical Cryptographic Techniques: Caesar Cipher, Vigenère Cipher, Playfair Cipher, Modern Cryptography & Cryptanalysis
Mathematical Foundations of Cryptography: Number Theory Basics: Prime Numbers, Modular Arithmetic, Euler's Theorem, Greatest Common Divisor (GCD), Fermat's Theorem, Random Number Generation

Module 2:

Symmetric & Asymmetric Cryptography
Symmetric Key Cryptography, Block Ciphers vs. Stream Ciphers, Data Encryption Standard (DES) & Triple DES, Advanced Encryption Standard (AES), Modes of Operation: ECB, CBC, CFB, OFB, CTR, Asymmetric Key Cryptography, Public Key Cryptosystems: RSA Algorithm, Key Exchange Mechanisms: Diffie-Hellman Key Exchange, Elliptic Curve Cryptography (ECC), Comparison of Symmetric & Asymmetric Cryptography

Module 3:

Hash Functions, Digital Signatures & Authentication Cryptographic Hash Functions, Properties of Hash Functions, Common Hash Algorithms: MD5, SHA-1, SHA-256, SHA-3, Message Authentication, Message Authentication Codes (MAC), HMAC (Hash-based Message Authentication Code), Digital Signatures & Certificates, RSA Digital Signature Algorithm, Digital Certificates & Public Key Infrastructure (PKI), SSL/TLS Encryption and HTTPS Authentication Mechanisms, Password-Based Authentication, Multi-Factor Authentication (MFA), Biometric Authentication

Module 4:

Network Security & Secure Protocols
Network Security Fundamentals, Security in OSI & TCP/IP Models, Firewalls and Intrusion Detection Systems (IDS/IPS), Virtual Private Networks (VPNs) & IPsec, Secure Network Protocols, Secure Email: PGP & S/MIME, Secure Web Transactions: HTTPS, SSL/TLS
Wireless Security: WPA, WPA2, WPA3, Network Attacks & Defense Mechanisms, Denial of Service (DoS) & Distributed DoS (DDoS) Attacks, Man-in-the-Middle (MITM) & Eavesdropping, SQL Injection & Cross-Site Scripting (XSS)

Module 5:

Emerging Trends & Cybersecurity Applications
Blockchain & Cryptography, Role of Cryptography in Blockchain, Bitcoin & Ethereum Security, Cloud Security & Zero Trust Security Model, Cloud Encryption Mechanisms, Zero Trust Architecture & Access Control
Cybersecurity & Ethical Hacking, Introduction to Ethical Hacking, Penetration Testing Basics, Legal & Ethical

Issues in Cybersecurity, Future Trends in Cryptography & Security

Labs / Practicals:

Preparing

References:

1. "The Joy of Cryptography" by Michael Rosulek
https://joyofcryptography.com/pdf/book.pdf

2. "Cryptography and Computer Security" by Chris Bourke
https://cse.unl.edu/~cbourke/cryptoNotes.pdf

3. "Applied Cryptography (20th Anniversary Edition)" by Bruce Schneier
https://www.schneier.com/books/applied-cryptography

4. "Cryptography and Network Security (Eighth Edition)" by William Stallings
http://williamstallings.com/Cryptography/Crypto8e-Student

5. "Cracking Codes With Python: An Introduction To Building And Breaking Ciphers" by Al Sweigart
https://inventwithpython.com/cracking

6. "Yet Another Introductory Number Theory Textbook (Cryptology Emphasis Version)" by Jonathan Adam Poritz
https://www.poritz.net/jonathan/share/yaintt.pdf

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
DCSE250011	Augmented Reality and Virtual Reality	3	0	2	4

Course Outcomes:

CO1: Describe the concept of virtual reality and Communication Media.
CO2: Understand current virtual reality hardware and software.
CO3: Understand various modeling approaches.
CO4: Illustrate the concepts of Human Factors and Applications of VR.
CO5: Build a Virtual Reality Application.

Module 1:

INTRODUCTION TO VIRTUAL REALITY (VR):
Defining Virtual Reality, Key elements of virtual reality experience, Virtual Reality, Tele-presence, Augmented Reality and Cyberspace.
Bird's-Eye View: Hardware, Software, Human Physiology and Perception.

Module 2:

Input Devices:(Trackers, Navigation, and Gesture Interfaces): Three-dimensional position trackers, navigation and manipulation, interfaces and gesture interfaces.
Output Devices: Graphics displays, sound displays & haptic feedback.

Module 3:

Modeling:
Geometric modeling, Kinematics modeling, Physical modeling, Behaviour modeling, Model management.

Module 4:

Augmented Reality (AR): Taxonomy, Technology and Features of Augmented Reality, AR Vs VR, Challenges with AR, AR systems and functionality, Augmented Reality Methods, Visualization Techniques for Augmented Reality, Enhancing interactivity in AR Environments, Evaluating AR systems.
AR software development : AR software, Camera parameters and camera calibration, Marker-based augmented reality, AR Toolkit.

Module 5:

Interaction & Audio:
Interaction - Motor Programs and Remapping, Locomotion, Manipulation, Social Interaction. Audio -The Physics of Sound, The Physiology of Human Hearing, Auditory Perception, Auditory
Rendering. Interaction - Motor Programs and Remapping, Locomotion, Manipulation, Social Interaction.
Audio -The Physics of Sound, The Physiology of Human Hearing, Auditory Perception, Auditory Rendering.

Labs / Practicals:

1.Introduction to Unity and Scene Setup.
2.Creating a Basic Virtual Reality Environment.
3. Marker-Based Augmented Reality using Vuforia.
4.Gesture and Object Interaction in AR/VR.
5.Deploying AR Application on Mobile Device.

References:

1. Virtual Reality Technology, Second Edition, Gregory C. Burdea & Philippe Coiffet, John Wiley & Sons, Inc, 2017.
2. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016.
3. Rajesh K. Maurya, Computer Graphics with Virtual Reality System, 3rd Edition, Wiley Publication, 2018.

4. William R. Sherman and Alan B. Craig, Understanding Virtual Reality Interface, Application,and Design, 2nd Edition, Morgan Kaufmann Publishers, Elsevier, 2019.
5. Grigore C. Burdea, Philippe Coiffet, Virtual Reality Technology, 2nd Edition, Wiley,2017.
6. K.S. Hale and K. M. Stanney, Handbook on Virtual Environments, 2nd Edition, CRC Press,2015.

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
DCSE250081	Fuzzy Logic and Fuzzy Sets	3	1	0	4

Course Outcomes:

CO1: Understand and explain the fundamentals of fuzzy sets, including operations, properties, and extensions.
CO2: Apply fuzzy logic and approximate reasoning to model uncertainty and imprecision in real-world scenarios.
CO3: Implement fuzzy inference systems for decision-making and control applications.
CO4: Develop and apply fuzzy clustering and pattern recognition algorithms to analyze complex data.
CO5: Integrate fuzzy logic concepts with conventional systems to enhance performance in intelligent applications.

Module 1:

Fundamentals of Fuzzy Sets: Basic concepts on fuzzy sets, Basic Types, Fuzzy sets versus crisp sets, Properties of alpha-cuts, Representation of fuzzy sets, Extension principle.

Module 2:

Operations on Fuzzy Sets and Relations: Operation on fuzzy sets, Fuzzy union, intersection and complement, combinations of operations, Fuzzy arithmetic – Fuzzy numbers, Arithmetic operations on fuzzy numbers, Fuzzy relations, Binary fuzzy relations, Fuzzy equivalence and compatibility relations, Fuzzy ordering relations, Fuzzy morphism.

Module 3:

Fuzzy Measures and Fuzzy Logic: Fuzzy measures, Belief and possibility measures, Evidence theory, Possibility theory versus Probability theory, Fuzzy logic, Multivalued logic, Fuzzy propositions, Fuzzy qualifiers.

Module 4:

Approximate Reasoning and Fuzzy Expert Systems: Approximate reasoning – Fuzzy expert system (an overview), Fuzzy implications, Selection of fuzzy implication, Multiconditional approximate reasoning.

Module 5:

Fuzzy systems: Fuzzy system & neural network, Fuzzy automata, Fuzzy clustering, Fuzzy pattern recognition.

Labs / Practicals:

NA

References:

1.S. N. Sivanandam, S. Sumathi and S. N. Deepa, Introduction to Fuzzy Logic using MATLAB, Springer.
2.S. N. Sivanandam, S. Sumathi and S. N. Deepa, Introduction to Fuzzy Logic using MATLAB, Springer.
3.Timothy J. Ross, Fuzzy Logic With Engineering Applications, Wiley, 3rd edition.

Course Code	Course Name	L	T	P	Cr
DOAI250051	Text Analysis	3	1	0	4

Course Outcomes:

CO1: Understand the foundations of text processing and representation.
CO2: Explore various methods of extracting information from unstructured text.
CO3: Apply NLP and machine learning techniques to analyze and interpret textual data.
CO4: Use real-world datasets to build and evaluate text-based models.

Module 1:

Introduction to Text Analysis:
Nature and structure of textual data,Text vs. Structured Data,
Applications: Sentiment analysis, document classification, chatbots, information extraction, Overview of NLP pipeline,
Text preprocessing: tokenization, stop-word removal, stemming, lemmatization.

Module 2:

Text Representation Techniques:
Bag of Words (BoW),TF-IDF (Term Frequency–Inverse Document Frequency),Word embeddings: Word2Vec, GloVe, FastText, Sentence embeddings and vector similarity, Dimensionality reduction (PCA, t-SNE for visualization).

Module 3:

Text Classification and Clustering:
Supervised learning for text (Naïve Bayes, Logistic Regression, SVM), Multi-class and multi-label classification, Evaluation metrics (Precision, Recall, F1-score, ROC-AUC),
Unsupervised learning: k-means clustering, topic modeling (LDA).

Dealing with imbalanced datasets

Module 4:

Deep Learning for Text:
Introduction to Recurrent Neural Networks (RNN), LSTM, GRU, Attention mechanisms and Transformers, Pre-trained language models (BERT, GPT, etc.),Transfer learning in NLP.

Module 5:

Advanced Text Analysis Applications:
Named Entity Recognition (NER), Information extraction and relation mining,Text summarization (extractive and abstractive), Question Answering Systems, Ethics and bias in NLP systems.

Labs / Practicals:

1. Using NLTK/Spacy for basic preprocessing, Tokenization and frequency analysis.
2. Implementing BoW and TF-IDF in Python, Visualizing word vectors.
3. Building a text classifier using Scikit-learn, Implementing topic models using Gensim.
4. Sentiment classification using LSTM,Using Hugging Face Transformers for text classification.
5. NER with SpaCy or Hugging Face,Summarization using pre-trained models.

References:

1. Bernard, H.R. and G. Ryan. 2010. Analyzing Qualitative Data: Systematic Approaches. Sage.
2. Yoav Goldberg, Neural Network Methods for NLP, Morgan & Claypool.

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
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
DCSE250045	Cyber Threat Intelligence	3	0	2	4

Course Outcomes:

CO1: What is cyber threat intelligence and why is it useful
CO2: What are threat actors and their motivations
CO3: What are the common threat vectors and vulnerability types
CO4: What is the intelligence lifecycle and why is it so important.

Module 1:

Objectives of Cyber threat Intelligence:
Concept of Cyber Threat Intelligence, Organizational Benefits of Cyber Threat Intelligence, The Role of a Cyber Threat Intelligence Analyst.

Module 2:

Introduction to threat actors:
Grey, White and Black Hat Hackers, Black Hat Hackers.

Module 3:

Threat Vectors:
Threat Vectors and Vulnerability Types, vectors encountered by organizations and gain a foundational understanding around attack surface

Module 4:

Intelligence Cycle:
The Intelligence Lifecycle,foundational pillar of cyber threat intelligence,uses and how the differing elements of the intelligence lifecycle come together.

Module 5:

Introduction to Law and Ethics:
Law and Ethics, how it applies to cyber threat intelligence and why it is so important, commonality between these laws and those in your own country.

Labs / Practicals:

N/A

References:

1. operationalizing threat intelligence: a guide to developing and operationalizing cyber threat intelligence programs by by Joseph Opacki and Kyle Wilhoit.
2. Mastering Cyber Intelligence by Jean Nestor M. Dahj.

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
DCSA250029	Digital Image Processing	3	0	2	4

Course Outcomes:

CO1: Understand the fundamental concepts of digital image formation, representation, and pixel relationships in various image types.
CO2: Explore and apply spatial domain techniques such as intensity transformations, filtering, and enhancement methods.
CO3: Analyze frequency domain techniques using Fourier and Cosine transforms for image enhancement and filtering.
CO4: Investigate image restoration methods and compression techniques to improve image quality and storage efficiency.
CO5: Apply morphological processing and segmentation techniques to extract meaningful features and objects from images.

Module 1:

Introduction to Digital Image Processing
Fundamentals of Digital Images: Definition, Applications, Components
Brightness, Adaptation, and Discrimination
Light and the Electromagnetic Spectrum
Image Sampling and Quantization
Basic Relationships Between Pixels (4-, 8-connectivity, adjacency, distance)
Types of Images: Binary, Grayscale, Color, Multispectral

Module 2:

Spatial Domain Filtering
Intensity Transformation Functions: Log, Power-law, Piecewise Linear
Histogram Processing: Histogram Equalization and Specification
Spatial Filtering Basics: Correlation and Convolution
Smoothing Filters:
Averaging Filters (Box, Gaussian)
Order-Statistics Filters (Median, Max, Min)
Sharpening Filters:
Laplacian Operator
Gradient Operators (Sobel, Prewitt)

Module 3:

Frequency Domain Filtering
Discrete Fourier Transform (DFT) and its Properties
Filtering in Frequency Domain:
Ideal Low-Pass and High-Pass Filters
Butterworth Filters
Discrete Cosine Transform (DCT): 1D and 2D Applications

Module 4:

Image Restoration and Compression
Image Degradation and Restoration Model
Noise Models: Gaussian, Salt-and-Pepper, Speckle
Restoration Techniques:

Mean, Median, Wiener Filters
Fundamentals of Image Compression
Compression Techniques:
Huffman Coding
Run-Length Encoding (RLE)
JPEG Compression Standard

Module 5:

Morphological Processing and Segmentation
Morphological Operations:
Erosion, Dilation, Opening, Closing
Hit-or-Miss Transformation
Basic Morphological Algorithms: Boundary Extraction, Region Filling, Convex Hull
Image Segmentation:
Point, Line, and Edge Detection (Sobel, Canny)
Thresholding (Global and Adaptive)
Region-Based Segmentation (Region Growing, Splitting and Merging)

Labs / Practicals:

1. Implement histogram equalization and spatial filtering using OpenCV/Matlab
2. Apply DFT and DCT transforms on sample images
3. Noise addition and restoration using filters
4. Encode and decode images using Huffman and RLE techniques
5. Morphological operations and object extraction
6. Perform image segmentation using thresholding and edge detection techniques

References:

1. Rafael C. Gonzalez & Richard E. Woods, Digital Image Processing, Pearson Education, 4th Edition
2. Anil K. Jain, Fundamentals of Digital Image Processing, PHI
3. S. Jayaraman et al., Digital Image Processing, Tata McGraw Hill
4. Bernd Jähne, Digital Image Processing, Springer
5.OpenCV Documentation and Python Libraries (https://docs.opencv.org)

Course Code	Course Name	L	T	P	Cr
DOAI250009	Big Data Analytics	3	0	2	4

Course Outcomes:

CO1: Describe big data characteristics, challenges, and applications across industries and domains.

CO2: Evaluate and apply NoSQL databases and distributed storage architectures for big data management.

CO3: Install, configure, and operate Hadoop and HDFS for batch-oriented big data processing.

CO4: Develop and implement data analysis workflows using Hadoop ecosystem tools including Hive, Pig, and HBase.

CO5: Design and implement real-time data processing and analytics using Spark Streaming and PySpark.

CO6: Apply visualization and regression techniques for analytical insights from large-scale data.

Module 1:

Introduction to Big Data Ecosystem

Big Data Characteristics: Volume, Variety, Velocity, Veracity, and Value, Business applications and case studies of big data analytics, Challenges in conventional systems, Big Data Analytics Lifecycle, Tools and Technologies in Big Data, Hadoop vs. Traditional Systems, NoSQL Overview (Key-Value, Document, Columnar, Graph databases), Basics of MongoDB and Cassandra for big data storage.

Module 2:

Hadoop Framework for Big Data

History and Evolution of Hadoop, Hadoop Distributed File System (HDFS): Architecture, Internals, and Operations, Block-level Storage, Fault Tolerance, MapReduce Paradigm: Concepts, Anatomy of a MapReduce Job, Failures, Scheduling, Shuffle, Sort, and Task Execution, Input / Output Formats in MapReduce, Hadoop Streaming and Integrations, Developing MapReduce Applications with Java and Python APIs.

Module 3:

Data Analysis with Hadoop Ecosystem

Hive: Data Modeling, Data Types, File Formats, HiveQL (DDL, DML, DQL), Querying Structured Data, Hive Services and Optimization.
Pig: Data Flow Language, Pig Latin Programming, Grunt Shell, Data Models, Operators, Joins, Developing and Testing Pig Scripts.
HBase: Architecture, Data Model, Integration with Hadoop, Use Cases.
Introduction to Data Lakes using Hadoop.
Hands-on with Hadoop Ecosystem for ETL pipelines.

Module 4:

Real-Time Big Data Analytics and Streaming

Stream Processing Fundamentals, Stream Data Models, Stream Architecture, Concepts of Sampling, Filtering, Counting in Streams, Real-time Sentiment Analysis, Stock Market Analysis using Streams, Introduction to Apache Kafka for Streaming Data Ingestion, Spark Basics: RDD, DataFrames, Spark SQL, In-Memory Computing, PySpark APIs, Spark Streaming for Real-Time Processing, Real-Time Analytics Applications, Case

Studies.

Module 5:

Analytical Models and Visualization for Big Data

Regression Techniques: Simple Linear Regression, Multiple Linear Regression, Model Interpretation for Large-Scale Data, Feature Engineering for Big Data, Visualization Techniques: Interactive Dashboards, Graphical Tools for Big Data (Tableau / PowerBI concepts), Visual Analytics: Trends, Patterns, Correlations, Interaction Models, Real-World Applications of Big Data Visualizations.

Labs / Practicals:

1. Setting up Hadoop, HDFS, and YARN on local and cloud-based environments (AWS / Azure / GCP).

2. Developing basic MapReduce jobs using Java and Python.

3. Implementing ETL pipelines using Hive and Pig.

4. Performing analytics with HBase and Hive.

5. Real-time data streaming with Apache Kafka and Spark Streaming (using PySpark).

6. Implementing basic ML pipelines over Spark (MLlib).

7. Visualizing Big Data using interactive tools (Matplotlib, Plotly, Tableau / PowerBI).

8. Group Project: Building and deploying an end-to-end Big Data Analytics pipeline on a real-world dataset.

References:

1. Tom White, Hadoop: The Definitive Guide, 4th Edition, O'Reilly Media, 2015.

2. Anand Rajaraman, Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 3rd Edition, 2020.

3. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & Sons, 2012.

4. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann / Elsevier, 2011.

5. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, Learning Spark: Lightning-Fast Big Data Analysis, 2nd Edition, O'Reilly Media, 2020.

6. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Harness the Power of Big Data – The IBM Big Data Platform, Tata McGraw Hill, 2012.

7. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

8. Da Ruan, Guoqing Chen, Etienne E. Kerre, Geert Wets, Intelligent Data Mining, Springer, 2007.

Official Documentation (Highly Recommended for Labs / Practical Work):

9. Official documentation for Hadoop: https://hadoop.apache.org/

10. Official documentation for Spark: https://spark.apache.org/docs/latest/

11. Official documentation for Hive: https://cwiki.apache.org/confluence/display/Hive/Home

12. Official documentation for Pig: https://pig.apache.org/

13. Official documentation for HBase: https://hbase.apache.org/

14. Official documentation for Kafka: https://kafka.apache.org/documentation/

15. Official documentation for PySpark: https://spark.apache.org/docs/latest/api/python/

Course Code	Course Name	L	T	P	Cr
DOAI250021	Data Science	3	0	2	4

Course Outcomes:

CO1: Describe the data science life cycle, key concepts, and the role of Python in data acquisition, preprocessing, and visualization.
CO2: Apply core Python programming constructs and data structures (lists, dictionaries, tuples, stacks, queues, data frames) for effective data manipulation and analysis.
CO3: Develop and utilize user-defined functions and classes to implement modular and object-oriented programming techniques in data science workflows.
CO4: Analyze and handle data using file operations, pandas data frames, and essential machine learning libraries such as NumPy, Scikit-learn, and Matplotlib.
CO5: Implement and evaluate machine learning models using Scikit-learn for classification, regression, clustering, and present insights through case studies like spam detection and customer segmentation.

Module 1:

Data Science Overview: Data Science life cycle; data acquisition, public data repositories, pre-processing, data visualization, Machine Learning algorithms, pattern evaluation, knowledge representation, analytical strategies; Python fundamentals: data types, mutable & immutable types, operators & expressions, standard library modules, the flow of control, etc.

Module 2:

Python data structures: String manipulation: string operators, slices, methods; List manipulation: creating & accessing the list, list operations, true copy of lists, nested lists; Tuples: creation, accessing, modifying tuples, nested tuples, methods & functions, tuple slicing; Dictionaries: key-value pairs, creation, accessing, modifying dictionaries, Stack, Queue, data frame, etc.

Module 3:

User-defined functions & Classes: modules in Python, importing python modules, name resolution, module aliasing, package/library, locating modules, using Python's standard library, user-defined functions: parameters in functions, passing arrays/tuples/dictionary, and lists to functions, function call, the scope of variables, dynamic typing and dynamic binding, Creating and instantiating classes with attributes and methods, public v/s non-public members, Inheritance, Polymorphism, etc.

Module 4:

Data Handling: data file operations: opening & closing a file, reading/writing/appending a file, relative and absolute paths, standard file streams, binary file operations, CSV file handling & pickling, concatenating multiple CSV files/text files, panada's DataFrame, convert pandas data frames to numpy arrays/CSV, etc; introduction to various ML libraries: NumPy, Pandas, Scikit-learn, SciPy, Statsmodels, MatplotLib, Seaborn, Sympy, etc. and their use cases.

Module 5:

Scikit-learn library: using sci-kit-learn modules for classification e.g. Support Vector Classifier & Regressor, CART, clustering (k-means), data reduction (PCA), data visualization (box plot), etc. Web Scraping with Beautiful Soup, use cases and case studies: spam detection, drug response, stock price prediction, customer segmentation, etc.

Labs / Practicals:

1. Drawing different patterns e.g. Pascal triangle, Pyramid of '*', etc. using loop and branching statements
2. Creating a Python class, instantiate 10 objects, save them in csv file, re-open and load them in Python workspace
3. Sorting a student database based on student identification numbers by reading it from csv file

4. Problem related to classification using CART, SVC, K-NN, etc
5. Problem e.g. Market-Basket analysis, related to association mining using Apriori, FPGrowth, etc.
6. Problem related to clustering e.g. image segmentation using K-means,
7. Problem related to regression analysis e.g. electricity load prediction during peak hours using SVR, CART, etc.
8. Problems related to anomaly detection e.g.
9. Training a Machine learning model for a real-life dataset with good practices such as Train/Test split, 10-fold cross validation, reporting performance with RoC plots etc. over a separate test dataset
10. Using all plot libraries to do data pre-processing & visualization e.g. Box plot, Bar chart, probability density function plot, etc.
11. Understanding data processing, reading, pre-processing, cleaning data set, generating Train/Test/Validation datasets, label encoding, one-hot encoding, etc. using Sklearn
12. Importing & using various ML libraries: NumPy, Pandas, Scikit-learn, SciPy, Statsmodels, MatplotLib, Seaborn, Sympy, etc.
13. Creating and using NumPy Array and converting it into other types
14. Creation and usage of panda's data frame and converting it into other types
15. Using various data visualization techniques such as probability density function, box plot, etc.
16. CSV file handling & pickling, concatenating multiple CSV files/text files
17. Implementing a regression algorithm for rain-fall prediction, stock market prediction, wine quality, etc.
18. Implementing a classification algorithm for image classification, breast cancer, etc.
19. Implementing an unsupervised algorithm for grouping data points into meaningful clusters.
20. Implementing a clustering algorithm for data reduction

References:

1. Cormen, Leiserson, Rivest and Stein, Introduction to algorithms (Main textbook)
2. Kleinberg and Tardos , Algorithm Design
3. Mark Weiss, Data structures and algorithm analysis in C++ (Java)
4. Aho, Hopcroft and Ullman, Data structures and algorithms
5. S. Sahni, Data Structures, Algorithms, and Applications in C++, Silicon Press

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
DCSA250009	Blockchain Technology	3	1	0	4

Course Outcomes:

Upon completion of the course, the learners will be able to:
CO1: Understand the concept of blockchain and its need.
CO2: Analyze methods of cryptography for application with blockchain.
CO3: Evaluate the working of blockchain.
CO4: Understand the underlying technology of transactions, blocks, proof-of-work, and consensus building.
CO5: Identify real world problems that blockchain can solve and analyze a use case.

Module 1:

Fundamentals of Blockchains Technology, The Structure of Blockchains, Blockchain Applications, The Blockchain Life Cycle, Distributed Ledger vs Centralized System, Merkel Root, Nonce Value, Hash Algorithm, and its applications.

Module 2:

Block Ciphers; Encryptions; Secret Keys; Elliptic Curve Cryptography; Hash cryptography; Encryption vs hashing; Digital Signature; Memory Hard Algorithm, Zero Knowledge Proof.

Module 3:

Introduction: Transactions, blocks, hashes, consensus, verify and confirm blocks, peer to peer networks, blocks of data in a chain, decentralization of networks, processes & workflows, cryptocurrencies, nodes, assets, consensus; types of blockchain; chain policy; working of blockchain; life of blockchain application; privacy, anonymity and security of blockchain.

Module 4:

Hyperledger: Introduction, where can Hyperledger be used, Hyperledger architecture, Hyperledger Fabric, features of Hyperledger; Open source blockchain platform technology; Tools & services; Cloud options in blockchains AWS; Azure workbench; Hyperledger console; Consensus: Proof of work, proof of stake, delegated proof of stake, proof of burn, BlocBox Protocol.

Module 5:

History; Distributed ledger; Smart contracts; Cryptocurrency; Bitcoin protocols; Mining strategy and rewards; Ethereum – construction; DAO; GHOST; Vulnerability; Attacks; Sidechain; Namecoin

Labs / Practicals:

NA

References:

1. D. Tapscott, A. Tapscott, Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World, Portfolio Publishers.
2. A. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, O'Reilly.
3. P. Champagne, The Book of Satoshi: The Collected Writings of Bitcoin Creator Satoshi Nakamoto, e53 Publishing, LLC.
4. M. Swan, Blockchain: Blueprint for a New Economy, O'Reilly.

5. R. Wattenhofer, The Science of the Blockchain, Inverted Forest Publishing.
6. R. Modi, Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain, Pack

Course Code	Course Name	L	T	P	Cr
DCSA250054	Multimedia & Animation Techniques	3	0	2	4

Course Outcomes:

CO1. Explain the concepts and components of multimedia systems.
CO2. Apply principles of image, audio, and video processing.
CO3. Create 2D and 3D animations using appropriate tools.
CO4. Design and develop multimedia applications using industry-standard software.
CO5. Evaluate multimedia systems based on performance, format, and usability.

Module 1:

Introduction to Multimedia:
Multimedia: Definition, Components, Applications
Multimedia Systems Architecture
Multimedia Authoring Tools & Types
File Formats: Text, Image, Audio, Video (JPEG, PNG, MP3, MP4, AVI, etc.)

Module 2:

Graphics and Image Editing:
Raster vs. Vector Graphics
Color Models: RGB, CMYK, HSV
Image Compression Techniques (Lossy & Lossless)
Tools: Photoshop, GIMP – Layers, Filters, Image Manipulation

Module 3:

Audio and Video Processing:
Digital Audio: Sampling, Bitrate, Formats
Sound Editing Tools: Audacity, Adobe Audition
Video Compression and Editing: Codecs, Frame Rate, Key frames Tools: Adobe Premiere, Final Cut Pro

Module 4:

Animation Techniques:
Principles of Animation (Squash & Stretch, Timing, etc.)
Types of Animation: 2D, 3D, Stop Motion
2D Animation Tools: Adobe Animate, Pencil2D
3D Animation Tools: Blender, Maya – Keyframing, Rigging, Rendering

Module 5:

Interactive Multimedia and Applications:
Multimedia in Web and Mobile
Story boarding and Scripting
Multimedia Project Life Cycle
Applications: E-learning, Gaming, Virtual Tours

Labs / Practicals:

1.Image Editing using Photoshop/GIMP
2.Audio Editing using Audacity
3.Video Clipping & Effects using Premiere Pro
4.Creating 2D animation with Adobe Animate
5. Modelling and Animating a basic object in Blender

References:

Textbooks:
1.Tay Vaughan, "Multimedia: Making It Work", McGraw Hill.
2.Ze-Nian Li & Mark S. Drew, "Fundamentals of Multimedia", Pearson.

Reference Books:
1.John Villamil-Casanova, "Multimedia: An Introduction", Prentice Hall.
2.Isaac Victor Kerlow, "The Art of 3D Computer Animation and Effects", Wiley.

Course Code	Course Name	L	T	P	Cr
DASH250015	Cognitive Systems	3	1	0	4

Course Outcomes:

CO1 : Demonstrate a thorough understanding of cognitive systems principles.
CO2 : Apply knowledge of cognitive science concepts to analyze and solve complex problems.
CO3 : Design and implement computational models of cognitive processes.
CO4 : Develop cognitive robotics systems capable of sensorimotor integration, spatial cognition, and human-robot interaction.
CO5 : Critically evaluate the ethical, societal, and practical implications of cognitive systems and their applications.

Module 1:

Introduction
Introduction to Cognitive Systems Overview of cognitive systems, Historical development, Interdisciplinary nature, Basic concepts in cognitive science, Cognitive architectures, Applications of cognitive systems, Cognitive system modeling approaches, Cognitive development across the lifespan, Neural correlates of cognition, Ethical considerations in cognitive systems, Future trends in cognitive science.

Module 2:

Foundations of Cognitive Psychology
Cognitive Psychology Fundamentals Human cognition and perception, Memory systems, Attention and consciousness, Language processing, Problem-solving and decision-making, Learning and adaptation, Emotion and cognition, Social cognition, Cognitive biases and heuristics, Cognitive development theories, Cognitive neuroscience methods.

Module 3:

Foundations of Cognitive Robotics
Cognitive Robotics Cognitive robotics fundamentals, Embodied cognition, Sensorimotor integration, Spatial cognition, Navigation and mapping, Human-robot interaction, Cognitive architectures for robots, Developmental robotics, Robot learning and adaptation, Multi-robot systems, Ethical considerations in cognitive robotics.

Module 4:

Professional Communication and Career Development
The role of Business Presentations, Planning and organizing presentations, Team Presentations, online Presentations. Understanding yourself, Career, Goal Setting, Preparing Resume, Resume Formats, Writing Covering Letters, and Enquiry mails, Preparing for the job interview.

Module 5:

Advanced Topics in Cognitive Computing and Applications
Applications and Advanced Topics Cognitive computing applications, Autonomous systems, Cognitive modeling and simulation, Brain-computer interfaces, Ethical considerations in cognitive systems, Future trends and challenges in cognitive science, Cognitive augmentation technologies, Cognitive rehabilitation techniques, Computational psychiatry, Neuromorphic computing, Human-centered AI design principles.

References:

1)Boden, M. A. (Ed.). "Handbook of Cognitive Science: An Embodied Approach." Elsevier.
2)Russel, S., & Norvig, P. "Artificial Intelligence: A Modern Approach." Pearson.
3)Thrun, S., Burgard, W., & Fox, D. "Probabilistic Robotics." MIT Press.
4)Luger, G. F., & Stubblefield, W. A. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving." Pearson.
5)Hutto, D. D., & Myin, E."Evolutionary Robotics and the Reach of Embodiment." Frontiers in Psychology.

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.