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

Bachelor of Computer Applications (BCA)

(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 nationally recognized undergraduate program in Computer Applications that imparts quality education, fosters innovation and entrepreneurship, and prepares graduates to meet the challenges of the dynamic digital world with ethical and societal responsibility.

Mission

1. To provide a strong foundation in computer science, software development, and digital technologies through industry-aligned curriculum and practical exposure.
2. To nurture creative thinking, problem-solving abilities, and effective communication among students.
3. To promote professional ethics, environmental awareness, and social responsibility in computing practices.
4. To enable students to pursue higher education, entrepreneurship, or successful careers in the IT industry.
5. To encourage innovation and lifelong learning to stay abreast with emerging trends in computing and information technology.

Program Education Objectives

Graduates of the BCA program will be able to:
1. Apply computing principles and practices to design, develop, and manage software applications across various domains.
2. Pursue higher studies or professional development in the field of computer science or related disciplines.
3. Demonstrate effective communication, teamwork, and leadership in multidisciplinary environments.
4. Uphold professional ethics and contribute positively to society through responsible use of technology.
5. Embrace lifelong learning to remain current with rapidly evolving technologies.

Program Outcomes

1. Apply knowledge of computing fundamentals, programming, and data structures to identify, analyze, and solve real-world problems effectively.
2. Design, develop, test, and maintain software applications using modern tools, frameworks, and programming environments.
3. Demonstrate ethical values, effective communication, teamwork, and leadership skills in professional and social contexts.
4. Understand the impact of computing solutions on society, environment, and culture, and act responsibly as an IT professional.
5. Recognize the need for continuous learning and adapt to emerging technologies for higher education, entrepreneurship, or career advancement.

Program Specific Outcomes

After completing the BCA program, students will be able to:
1. Develop and deploy secure and efficient software applications using modern programming languages and frameworks.
2. Apply data structures, algorithms, and database principles to solve computational problems in real-life scenarios.
3. Utilize cloud computing, artificial intelligence, and web technologies in building innovative and scalable solutions.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Value Added Courses	VA	6
Professional Elective Courses	PE	20
Audit Courses (without grade or credit)	AU	0
Ability Enhancement Courses	AE	8
Open Elective Courses	OE	12
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	16
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-BCAGEN-101	DCSE250077	Fundamentals of Information Technology	3	0	2	4
EE-BCAGEN-102	DCSE250111	Programming in C	2	0	4	4
OE-BCAGEN-103	Elective	Open Elective				4
AE-BCAGEN-104	DASH250016	Communication Skills	1	0	2	2
PC-BCAGEN-105	DASH250059	Basic Mathematics-I	3	1	0	4
AU-BCAGEN-106	Elective	Audit Elective				0

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-201	DCSE250026	Computer Architecture	3	1	0	4
PC-BCAGEN-202	DCSA250084	Web Technologies	3	0	2	4
PC-BCAGEN-203	DCSE250079	Fundamentals of Programming in Python	2	0	4	4
PE-BCAGEN-204	Elective	Program Elective				4
OE-BCAGEN-205	Elective	Open Elective				4
VA-BCAGEN-206	DASH250109	Yoga Practices	0	0	4	2
EE-BCAGEN-207	DCSE250122	Version Control with Git	3	0	2	4

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-301	DCSA250015	Data Structures using Python	3	0	2	4
PC-BCAGEN-302	DCSA250059	Object-Oriented Programming with Java	2	0	4	4
PC-BCAGEN-303	DCSE250041	Fundamentals of Database Systems	3	0	2	4
PC-BCAGEN-304	DASH250017	Graph theory	3	1	0	4
PE-BCAGEN-305	Elective	Program Elective				4
AU-BCAGEN-306	Elective	Audit Elective				0

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-401	DCSE250047	Analysis of Algorithms	3	1	0	4
PC-BCAGEN-402	DASH250011	Probability and Statistics	3	1	0	4
PC-BCAGEN-403	DCSE250042	Introduction to Operating System	3	1	0	4
EE-BCAGEN-404	DCSA250003	Mobile Application Development using Android	2	1	2	4
OE-BCAGEN-405	Elective	Open Elective				4
VA-BCAGEN-406	DASH250052	Environmental Science	2	0	0	2

Semester 5						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-501	DCSA250071	Software Engineering	3	1	0	4
PC-BCAGEN-502	DASH250001	Mathematical Foundations for AI	3	1	0	4
PC-BCAGEN-503	DCSE250034	Data Communication and Computer Networks	3	1	0	4
PE-BCAGEN-504	Elective	Program Elective				4
AE-BCAGEN-505	DASH250056	Financial Accounting & Management	2	0	0	2
EE-BCAGEN-506	EE	Internship				4
AU-BCAGEN-507	Elective	Audit Elective				0

Semester 6						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-601	DOAI250029	Artificial Intelligence with Python	2	0	4	4
PC-BCAGEN-602	DCSE250033	Introduction to Theory of Computation	3	1	0	4
PE-BCAGEN-603	Elective	Program Elective				4
AE-BCAGEN-604	DASH250026	Design Thinking	1	0	2	2
AE-BCAGEN-605	DASH250104	Technical Presentation and Report Writing	0	0	4	2
VA-BCAGEN-606	DASH250091	Professional Values and Ethics	2	0	0	2

Semester 7						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-701	DOAI250014	Data Analysis using Python	2	0	4	4
PC-BCAGEN-702	DOAI250023	Data Visualization	2	0	4	4
PC-BCAGEN-703	DOAI250032	Machine Learning	3	0	2	4
PE-BCAGEN-704	Elective	Program Elective				4
PE-BCAGEN-705 (H)	Elective	Program Elective				4
EE-BCAGEN-705 (HWR)	EE	Minor Project				4
AU-BCAGEN-706	Elective	Audit Elective				0

Semester 8						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-BCAGEN-801	DCSA250031	Full Stack Web Development	3	0	2	4
PC-BCAGEN-802	DOAI250019	Data Mining and Warehousing	3	0	2	4
PE-BCAGEN-803 (H)	Elective	Program Elective				4
EE-BCAGEN-803 (HWR)	EE	Research Project/ Dissertation				11
PE-BCAGEN-804 (H)	Elective	Program Elective				4
EE-BCAGEN-805 (H)	EE	Dissertation				3

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DASH250070	Management Information System	3	1	0	4	
DCSA250009	Blockchain Technology	3	1	0	4	
DCSA250017	Linux Environment	3	0	2	4	
DCSA250029	Digital Image Processing	3	0	2	4	
DCSE250004	Advanced Data Structures and Algorithms	3	0	2	4	
DCSE250024	Compiler Design	3	1	0	4	
DCSE250043	Cryptography and Network Security	3	0	2	4	
DCSE250044	Cyber Security	3	0	2	4	
DCSE250052	Introduction to Cloud Computing	3	0	2	4	
DCSE250074	Programming Algorithms	3	1	0	4	
DOAI250009	Big Data Analytics	3	0	2	4	
DOAI250015	Data Analytics and Visualization	2	0	4	4	
DOAI250021	Data Science	3	0	2	4	
DOAI250027	Generative AI	3	0	2	4	
DOAI250035	Introduction to Computer Vision	3	1	0	4	
DOAI250036	Introduction to Deep Learning	3	0	2	4	
DOAI250038	Introduction to Natural Language Processing	3	1	0	4	
DOEE250127	IOT: Technologies and Applications	3	0	2	4	

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

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

DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0
DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0
DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0
DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0
DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

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

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

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

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

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

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

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

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

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

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DCSE250077	Fundamentals of Information Technology	3	0	2	4

Course Outcomes:

CO1. Understand the fundamentals of computers, including data representation, classification, functional components, and common input-output devices.
CO2. Explain types of software, system and application packages, programming languages, and utility programs used in computer applications.
CO3. Perform number system conversions and apply coding schemes like ASCII and Unicode for data representation in computers.
CO4. Describe microprocessor architecture, processor types, system buses, CPU components, and commonly used processors with their associated technologies.
CO5. Apply basic internet concepts, tools, and applications such as web browsing, email, videoconferencing, and client-server technologies.

Module 1:

What is Computer? Representation of data / information. What is Data Processing? Characteristics of a Computer System, Evolution of Computer, Generation of computers, Block Diagram of Digital Computer, Classification of Digital Computers Classification of Computers,
Functional Components of a computer- Central Processing Unit, Memory-Primary MemoryRAM, ROM, Booting, Secondary Storages Devices : Floppy and Hard Disks, Optical Disks CD-ROM, DVD, Mass Storage Devices : USB thumb drive. Input /Output Devices- Keyboard, Mouse, Trackball, Joystick, Digitizing Tablet, Scanners, Digital Camera, MICR, OCR, OMR, Bar-code Reader, Voice Recognition, Light Pen, Touch Screen, Monitors, Printers & types – Daisy wheel, Dot Matrix, Inkjet, Laser, Line Printer, Plotter, Sound Card and Speakers.

Module 2:

Types of Software, Classification of System and Application Software, System Software - Operating System, Devices Drivers, Application development Packages, Word Processors, Electronic Spreadsheet, Data Base Management System, Presentation packages, Desk Top Publishing Software(DTP), Overview of languages- Machine language, Assembly language, high level languages, Types of high level languages, Generation of languages, Commands , Utility Program-Disk Cleanup Utility, Desk backup Utility, Antivirus Utilities, Customized Application Software.

Module 3:

NUMBER SYSTEM: Binary, Octal, Decimal, Hexadecimal and Conversion between Number Systems, ASCII Codes, Unicode.

Module 4:

MICROPROCESSOR: Basic concepts, Clock speed (MHz, GHz), 16 bit, 32 bit, 64 bit processors: Types: CISC, RISC, Concepts of System Buses, Address Bus, Data Bus, Concepts of Accumulator, Instruction Register, Program counter, Commonly used CPUs and CPU related Terminologies: Intel Pentium Series, Intel Celeron, Cyrix, AMD Series, Xeon, Intel Mobile, Mac Series, CPU Cache, Concept of heat sink and CPU fan, Motherboard; Single, Dual and Multiple Processors.

Module 5:

Internet: Basic Internet terms: Web Page, Website, Home page, Browser, URL, Hypertext, ISP, Web Server, HTML, DHTML, XML, Introduction to client side and server side scripting.
Applications: WWW, e-mail, Instant Messaging, Internet Telephony, Videoconferencing, Web Browser & its environment

Labs / Practicals:

1. Hardware familiarizing with various I/O Peripheral devices, and storage devices.
2. Familiarity with DOS, Implementing various internal and external command sin DOS.
3. MS-Windows: Familiarizing with windows operating system; using built-in accessories; managing files and folders using windows explorer; working with control panel; installing hardware and software.
4. MS-Office (or any other Office Suite), meaning and features, its components.
5. MS-Word (or any other word processor): Creating Document Files, Saving, Closing Files, Page Settings, and Formatting Text. Spell Checking, Thesaurus, Creating Tables, Adding rows, columns. Printing Documents, Setting Print Settings, creating labels and mail merge, taking Printout
6. MS-Excel-Working with worksheets, formulas & functions, Inserting charts, printing in Excel.
7. MS Power Point-Views, Designing, viewing, presenting & Printing of Slides.
8. Internet: Navigating with Internet Explorer; surfing the net, using search engines; using email facility.

References:

1. Basics of Information Technology-Sumita Arora,Dhanpat Rai & Co (Pvt.) Ltd.,New Delhi.
2. Foundation of Information Technology – D.S. Yadav, New Age International Publisher.
3. V. Raja Raman, “Introduction to Computers”, PHI, 1998.
4. Alex Leon & Mathews Leon, “Introduction to Computers”, Vikas Publishing House, 1999. Norton Peter, “Introduction to computers”, 4th Ed., TMH, 2001.

Course Code	Course Name	L	T	P	Cr
DCSE250111	Programming in C	2	0	4	4

Course Outcomes:

CO1. Understand the fundamentals of computation, algorithms, flowcharts, and develop basic C programs using variables, operators, and expressions.
CO2. Apply decision control and looping statements to solve computational problems using conditional, iterative, and nested program structures.
CO3. Design and implement modular programs using library and user-defined functions, recursion, parameter passing, and appropriate variable scope.
CO4. Manipulate arrays, strings, structures, and unions to efficiently store, process, and manage data in C programs.
CO5. Use pointers, dynamic memory allocation, and file operations to develop flexible, memory-efficient, and persistent C applications.

Module 1:

Overview of C:
The Basic model of computation, algorithm, flowchart, programming languages, compilation, linking, loading, testing and debugging, documentation. Programming using C: features of C language, basic structure of C program, comments, header files, data-types, variables, input & output operations, operators, type conversions, operator precedence and associativity, evaluation of expressions.

Module 2:

Decision making & Looping statements:
Introduction to decision control statements, Conditional Statements (if statement, if-else statement, nested if-else statement, else-if ladder, switch-case), goto statement, Understanding syntax and utility of Iterative Statements (while, do-while, for loops, infinite loops), Use of break and continue in Loops, Using Nested Statements.

Module 3:

Module 3:
Functions:
Utility of functions, Library function, User-defined function, Need for user-defined functions, Function definition, function call, function declaration, Return statement, Passing parameters/ arguments to the function: call by value, call by reference, Recursion, Advantages of using function, scope of variables, storage classes.

Module 4:

Arrays, Structures and Unions:
Declaration of Arrays, Accessing elements of array, initialization of one & two dimensional arrays, Operations that can be performed on one and two dimensional arrays, character arrays & strings, string & character functions, String operations, Programming examples. Understanding utility of structures and unions, Declaring, initializing and using simple Structures and Unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

Module 5:

Pointers and File:
Declarations, initialization, accessing a variable through its pointer, chain of pointers, pointer expressions, pointer increments & scale factor, pointers & arrays, null & generic pointers,
dynamic memory allocation : malloc(); calloc(); realloc(); free(), Defining & opening a file, closing a file, Input/ Output operations on files.

Labs / Practicals:

1. Write a C program to display student name using Basic Syntax.
2. Write a C program to convert a temperature from one unit to another say Celsius to Fahrenheit.
3. Write a C program to generate natural numbers up to n terms.
4. Write a C program to find the sum of n natural numbers.
5. Write a C program to generate the first 10 even numbers and also find its sum.
6. Write a C program to generate the first 10 odd numbers and also find its sum.
7. Write a C program to exchange the values of the two variables without using a third Variable
8. Write a C program to generate Prime numbers up to n terms.
9. Write a C program to find Factorial of a number.
10. Write a C program to generate Fibonacci series of number up to n terms.
11. Write a C program to reverse the digits of a given positive number.
12. Write a C program to generate Fibonacci numbers between 1 and N using recursion, N being a natural number.
13. Write a C program to calculate the net salary of an employee based on basic pay, DA, TA and PF.
14. Write a function to return the HCF of two positive integers. Write a main function to read two positive integers and print their HCF and LCM by using the above function.
15.Write a program to convert a decimal number into binary number using function.
16. Write a C program to find the addition of 2 matrices.
17. Write a C program to find multiplication of 2 matrices.
18. Write a C program to find the length of a string and arrange the string in ascending order.
19. Write a C program to count the number of characters, vowels, consonants and digits in a text line.
20. Write a C program to arrange the accepted numbers in ascending order and descending order.
21. Write a C program to find the biggest/ smallest no. in a given list of numbers.
22. Write a C program to find the sum of digits of an integer reducing it to a single digit.
23. Write a C program to check whether a string is a palindrome.
24. Write a C program to reverse a given string.
25. Write a C program to count the no. of vowels in a word/sentence.
26. Write a program using pointers to copy a string to another string variable (without using library function).
27. Declare a structure of a student with details like roll number, student name and total marks. Using this, declare an array with 50 elements.Write a program to read details of n students and print the list of students who have scored 75 marks and above.
28. write a program to copy a text file to another file.

References:

1. E. Balagurusamy, Programming in Ansi C, Tata McGraw Hill.
2. B.W. Kernighan and D M.Ritchie, The C Programming Language, PHI.
3. Gottfried,B.S, Programming with C, TMH Pub. Co. Ltd., New Delhi.
4. Reema Thareja, Computer Fundamentals & Programming in C, Oxford UP.

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
DASH250059	Basic Mathematics-I	3	1	0	4

Course Outcomes:

CO1: Apply the concepts of sets, relations, and functions to model real-world problems and analyze their properties.
CO2: Use matrices and determinants to solve systems of linear equations and apply Cramer's Rule in engineering and computational contexts.
CO3: Demonstrate the ability to differentiate functions, analyze continuity, and apply derivatives for solving problems involving maxima, minima, and monotonicity.
CO4: Apply integration techniques to solve definite and indefinite integrals and compute areas under curves and volumes of solids of revolution.
CO5: Analyze algebraic expressions involving real and complex numbers, polynomials, and progressions to solve linear and quadratic equations in mathematical and applied contexts.

Module 1:

Sets, Relations and Functions (9 hrs)
Sets: Definition, Types of sets, Venn diagrams, operations on sets
Relations: Relations and its Properties, Domain, co-domain, and range, Equivalence Relation, Partial Order Relation
Functions: Functions and its Properties, Domain and Range, Types of functions, Composite and Inverse Functions, Exponential, Logarithmic, Trigonometric functions, Even and Odd functions.

Module 2:

Matrices and Determinants (9 hrs)
Matrix, Types of matrices, matrix operations, Determinants up to 3 orders, properties of determinants, Inverse of matrix, Applications of matrices and determinants in solving linear equations by using Cramer's Rule.

Module 3:

Differential Calculus (7 hrs)
Limits and continuity, types of discontinuities, Differentiation, rules of differentiation, higher-order derivatives, Applications of differentiation: maxima and minima for single variable, Increasing and Decreasing functions.

Module 4:

Integral Calculus (11 hrs)
Integration, techniques of integration (by substitution and by parts), definite and indefinite integrals, Applications of integration: area under curves, volume of solids of revolution.

Module 5:

Algebra (9 hrs)
Number systems: Real and complex numbers, laws of exponents, Polynomials and factorization, Linear and quadratic equations in one and two variables, Arithmetic and geometric progressions.

Labs / Practicals:

N/A

References:

1)B.S. Grewal – Higher Engineering Mathematics, Publisher: Khanna Publishers.
2)N.P. Bali & Manish Goyal – A Textbook of Engineering Mathematics, Publisher: Laxmi Publications.
3)R.D. Sharma – Mathematics for Class XI & XII (Vol I & II), Publisher: Dhanpat Rai Publications.
4)Shanti Narayan & P.K. Mittal – A Textbook of Matrices, Publisher: S. Chand Publishing.

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
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
DCSE250079	Fundamentals of Programming in Python	2	0	4	4

Course Outcomes:

CO1: Develop algorithmic solutions to solve simple and well-defined computational problems effectively.
CO2: Write and demonstrate Python programs using basic statements, expressions, and syntax constructs.
CO3: Explain and implement control flow and function concepts in Python to solve programming tasks.
CO4: Apply Python data structures, including lists, tuples, and dictionaries, for managing compound data.
CO5: Describe and use file handling, exceptions, modules, and packages in Python-based problem-solving.

Module 1:

Python Introduction, Technical Strength of Python, Introduction to Python Interpreter and program execution, Using Comments, Literals, Constants, Python's Built-in Data types, Numbers (Integers, Floats, Complex Numbers, Real, Sets), Strings (Slicing, Indexing, Concatenation, other operations on Strings), Accepting input from Console, printing statements, Simple 'Python' programs. Flow Chart Symbols, Basic algorithms/flowcharts for sequential processing, decision based processing and iterative processing.

Module 2:

Assignment statement, expressions, Arithmetic, Relational, Logical, Bitwise operators and their precedence, Conditional statements: if, if-else, if-elif-else; simple programs, Notion of iterative computation and control flow –range function, While Statement, For loop, break statement, Continue Statement, Pass statement, else, assert. Lists, tuples, and dictionaries—slicing, indexing, concatenation, and mutability. Operations such as finding max, min, mean, linear search, and frequency count using dictionaries.

Module 3:

Top-down approach of problem solving, Modular programming and functions, Function parameters, Local variables, the Return statement, DocStrings, global statement, Default argument values, keyword arguments, VarArgs parameters. Library function-input(), eval(),print(), Recursion.

Module 4:

String Functions: count(), find(), rfind(), capitalize(), title(), lower(), upper(), swapcase(), islower(), isupper(), istitle(), replace(), strip(), lstrip(), rstrip(), aplit(), partition(), join(), isspace(), isalpha(), isdigit(), isalnum(), startswith(), endswith(), encode(), decode(), String: Slicing, Membership, Pattern Matching, Numeric Functions: eval(), max(), min(), pow(), round(), int(), random(), ceil(), floor(), sqrt(), Date & Time Functions.

Module 5:

File opening in various modes and closing of a file, Reading from a file, Writing onto a file, File functions-open(), close(), read(), readline(), readlines(),write(), writelines(),tell(),seek(), Command Line arguments.

Labs / Practicals:

1. Write a program to take name and age from user and display it.
2. Write a program illustrating the various operators in python (arithmetic, relational, logical, and bitwise operators).
3. Write a program to check the data type of given variables.
4. Write a program to swap two variables.
5. Write a program to check whether a number is even or odd.
6. Write a program to find the largest among three numbers.
7. Write a program to check if a number is positive, negative or zero.
8. Write a program to check whether a year is a leap year or not.
9. Write a program to calculate grade based on marks.
10. Write a program to print the multiplication table of a number.
11. Write a program to find the factorial of a number using a loop.

12. Write a program to display Fibonacci series up to n terms.
13. Write a program to find the sum of digits of a number.
14. Write a program to print all prime numbers in a given range
15. Write a Python function to produce pyramid triangles.
16. Write a program to count vowels and consonants in a string.
17. Write a program to reverse a string.
18. Write a program to check whether a string is a palindrome.
19. Write a program to find the length of a string without using len().
20. Write a program to remove all punctuation from a string.
21. Write a program to find the largest and smallest number in a list.
22. Write a program to add elements to a list and sort them.
23. Write a program to find the sum and average of list elements.
24. Write a program to find duplicates in a list.
25. Write a program to convert a list to a tuple and vice versa.
26. Write a function to calculate factorial of a number.
27. Write a function to check whether a number is prime.
28. Write a function that returns the square of a number.
29. Write a program with a function to count even and odd numbers in a list.
30. Write a recursive function to find the nth Fibonacci number.
31. Write a function that takes the lengths of three sides: side1, side2 and side3 of the triangle as the input from the user using input function and return the area of the triangle as the output. Also, assert that sum of the length of any two sides is greater than the third side.
32. Write a program to create and write to a text file.
33. Write a program to read contents of a file and display it.
34. Write a program to count the number of lines and words in a file.
35. Write a program to append data to an existing file.
36. Write a program to copy contents from one file to anothe

References:

1. Python Network Programming Cookbook by Pradeeban Kathiravelu, Dr. M. O. Faruque Sarkar, PACKT.
2. Head First Python by Paul Berry, O'Reilly
3. Dive into Python by Mark Pilgrim, APress
4. Beginning Programming with Python Dummies by John Paul Meuller.

Course Code	Course Name	L	T	P	Cr
DASH250109	Yoga Practices	0	0	4	2

Course Outcomes:

CO1: Understand the fundamental principles of yoga and its relevance to holistic well-being.
CO2: Demonstrate proficiency in basic yogic postures, breathing techniques, and meditative practices.
CO3: Analyze personal stress patterns and adopt yogic techniques to manage them effectively.
CO4: Integrate yoga into daily routine for enhancing concentration, posture correction, and digital well-being.
CO5: Develop a personalized yoga practice plan aligned with lifestyle, academic, and professional demands.

Module 1:

Introduction to Yoga and Wellness
Concept and meaning of Yoga; goals of Yoga (body–mind–soul alignment); health benefits of yoga in the digital age; importance for computer users – posture, screen fatigue, sleep, mental clarity; yogic lifestyle.

Practices: Sukshma Vyayama (joint movements), Eye exercises, Breathing awareness, Shavasana (relaxation).

Module 2:

Asanas for Posture, Flexibility and Strength
Warm-up routines; standing, sitting, prone and supine asanas; alignment and body awareness; benefits for spinal health, shoulders, and hips – crucial for long hours at the desk.

Practices: Tadasana, Trikonasana, Vajrasana, Bhujangasana, Shalabhasana, Ardha Matsyendrasana, Paschimottanasana, Pawanmuktasana, Marjariasana.

Module 3:

Pranayama and Breath Control
Concept of prana and breath awareness; balancing the nervous system through breath; energy enhancement and mental focus.

Practices: Anulom-Vilom, Bhramari, Ujjayi, Sheetali and Sheetkari; sitting postures (Sukhasana, Padmasana); Nadi Shuddhi techniques.

Module 4:

Meditation and Mental Wellness
Mindfulness, Dharana (concentration), and Dhyana (meditation); reducing digital stress and exam anxiety; improving attention span for coding and analytical work.

Practices: Om chanting, Yoga Nidra (guided relaxation), Trataka (candle gazing), guided meditation for clarity and calm.

Module 5:

Lifestyle Integration and Self-Practice
Designing a weekly yoga plan; yoga for posture correction, eye care, sleep regulation, and productivity; use of yoga apps and wearables; tracking improvement; ethics and discipline in yogic living.

Practices: Integration of all components in a 30-minute daily sequence; peer-guided sessions; journaling and reflection on mental/physical state.

Labs / Practicals:

Assessment Criteria (Suggested)
Attendance and active participation: 30%

Performance in posture, breathing, and alignment: 30%
Practical exam (asana-pranayama flow + meditation): 20%
Yoga journal/self-reflection report: 20%

References:

1. Swami Satyananda Saraswati – Asana Pranayama Mudra Bandha, Bihar School of Yoga
2. Dr. H.R. Nagendra & Dr. R. Nagarathna – Yoga for Students, Vivekananda Yoga Anusandhana Samsthana (S-VYASA)
3. Swami Niranjanananda Saraswati – Yoga Darshan, Bihar Yoga Bharati
4. S.K. Prabhakar – Yoga for Computer Users, Yoga Publications Trust
5. B.K.S. Iyengar – Light on Yoga, HarperCollins India

Course Code	Course Name	L	T	P	Cr
DCSE250122	Version Control with Git	3	0	2	4

Course Outcomes:

CO1. Create, configure, and manage Git repositories both locally and on GitHub.
CO2. Collaborate on projects using branches, pull requests, and peer code review techniques.
CO3. Manage organizational and team-level permissions on GitHub effectively.
CO4. Deploy static websites using GitHub Pages and monitor project activity through analytics.
CO5. Configure user and repository settings to streamline project workflows and enhance security.

Module 1:

Exploring the repository's main page: Creating a new repository, The commits page and a comparison with the git log command, The branches page and a comparison with the git branch command, The Raw, Blame, and History buttons, The Watch, Star, and Fork buttons, Changing the description and URL.
Issue tracker: Creating a new issue, Assigning issues to users, Labels, Creating new label names and setting different colors, Using labels to group issues, Milestones, Creating a new milestone, Adding issues to milestones, Using milestones to see which issues are resolved or are yet to be resolved, README file, Navigating with keyboard shortcuts

Module 2:

Teams: Creating a team, Inviting people, Accepting an invitation, Team member permissions, Requesting to join a team, Step one – as a user, Step two – as an owner or team maintainer, Adding repositories to a team, Team discussions.
The People tab: Managing access levels, Difference between Members and Outside collaborators, Demoting to an outside collaborator.
Organization settings: Profile, Security, Audit log, Third-party access, Teams, Transfer a repository to an organization's namespace, Convert a user account into an organization, Mentioning teams, Organization feed only in dashboard

Module 3:

Pull requests,The connection between branches and pull requests, Creating branches directly in a project – the shared repository model, Creating branches in your fork – the fork and pull model, Create and submit a pull request, Using the Compare & pull request button, Using the compare function directly, Using the GitHub web editor, The shared repository model, The fork and pull model, Submitting a pull request, Peer review and inline comments, The layout of a pull request, The review process, Correcting mistakes, Merging the pull request, Removing/restoring a branch after the pull request is merged, Reverting a pull request.

Module 4:

GitHub Pages: Creating a user or an organization page, Creating a project page, Choosing a theme to style your page, Using a custom domain, Jekyll, Customizing your page using Jekyll.
Web analytics: Pulse, Contributors – additions/deletions, Community profile, Commits over time, Code frequency, Dependency graph, Network, Forks, Traffic.

Module 5:

User settings: Profile, Setting up multiple emails, Managing your SSH keys, Setting up two-factor authentication, Repository settings, Changing the default branch that appears in a repository's main page, Enabling/disabling the wiki, Enabling/disabling the issue tracker, Adding collaborators, Transferring ownership – user to organization, Deleting a repository, Finding the size of your repositories, Fine-tuning email notifications

Labs / Practicals:

1. Create a local Git repository, add files, make initial commits, and connect the repository to a remote GitHub repository using git remote.

2. Create and switch between branches, make changes in each branch, and merge them back into the main branch using Git commands.
3. Exploring the GitHub interface, including commits, branches, Raw, Blame, and History buttons, and editing the repository description and URL.
4. Create and manage issues in a GitHub repository, assign them to team members, add labels, create milestones, and use keyboard shortcuts for efficient navigation.
5. Create a GitHub organization and teams, invite members, assign roles, add repositories, and configure organization-level settings such as audit log and third-party access.
6. Fork a repository, make changes on a new branch, and submit a pull request to the original repository. Participate in peer reviews and learn how to resolve conflicts.
7. Create a user or project website using GitHub Pages, applying themes, optionally customizing with Jekyll, and publishing the site online.
8. Explore the analytics features of GitHub such as Pulse, Traffic, Contributors, Network, and Dependency Graph to monitor project progress and team contributions.
9. Modify repository settings including the default branch, enabling/disabling the wiki or issue tracker, adding collaborators, and safely transferring or deleting a repository.
10. Configure their GitHub profile, set up multiple email addresses, manage SSH keys, enable two-factor authentication, and customize notification preferences.

References:

1. "GitHub Essentials" by Achilleas Pipinellis.
2. “Introducing GitHub” by Brent Beer.
3. “Introduction to Git and GitHub by Bobby Iliev.

Course Code	Course Name	L	T	P	Cr
DCSA250015	Data Structures using Python	3	0	2	4

Course Outcomes:

CO1: Demonstrate an in-depth understanding of core data structures and their applications.
CO2: Develop and implement advanced data structures using Python.
CO3: Evaluate algorithm performance using time and space complexity analysis.
CO4: Design and implement efficient algorithms for solving computational problems.
CO5: Build end-to-end applications using appropriate data structures.

Module 1:

Introduction to Data Structures: Overview of Data Structures, Types of Data Structures, Abstract Data Types (ADT), Complexity Analysis (Best, Worst, and Average Case), Big O Notation, Recursion in Python, Python Data Structures Overview (Lists, Tuples, Sets, and Dictionaries), Basic Input-Output Operations, Python Libraries for Data Structure Implementation (NumPy, SciPy).

Module 2:

Linear Data Structures: Arrays and Lists in Python, Array Operations (Insertion, Deletion, Searching, Sorting), Stack and Queue Implementations using Arrays and Linked Lists, Circular Queue and Priority Queue, Implementing Deque using Python, Problem Solving with Linear Data Structures.

Module 3:

Non-Linear Data Structures: Introduction to Trees, Binary Trees, Binary Search Trees (BST), AVL Trees, B-Trees, Tree Traversal Techniques (Inorder, Preorder, Postorder), Implementing Trees using Python, Graphs - Representation using Adjacency Matrix and List, Graph Traversal Techniques (BFS, DFS), Graph Algorithms (Dijkstra's, Floyd-Warshall, Prim's, Kruskal's), Pathfinding Algorithms using Python.

Module 4:

Hashing and Advanced Searching Algorithms: Hashing Techniques, Hash Functions, Collision Resolution (Chaining, Open Addressing), Implementing Hash Tables in Python, Searching Algorithms (Linear Search, Binary Search, Interpolation Search), Search Optimization Techniques, Implementing Search Operations on Large Data Sets

Module 5:

Sorting and Real-World Applications: Advanced Sorting Algorithms (Heap Sort, Merge Sort, Quick Sort, Radix Sort, Bucket Sort), Comparative Analysis of Sorting Algorithms, Implementing Sorting Algorithms using Python, Real-World Problem Solving using Data Structures, Capstone Project: Design and Implement an Efficient Data Structure-Based Solution for a Real-Life Problem.

Labs / Practicals:

1. Write a Python program to demonstrate the use of built-in data structures: Lists, Tuples, Sets, and Dictionaries.
2. Implement a Python program to calculate time complexity (operation counting) for linear and binary search.
3. Write a recursive Python program to find the factorial of a number and trace its call stack.
4. Create a Python script using NumPy to perform array operations such as addition and slicing.
5. Develop a Python function that uses recursion to print Fibonacci series up to 'n' terms.
6. Write a Python program to implement an array with operations: insertion, deletion, and search.
7. Implement a Stack using Python list and perform Push, Pop, and Display operations.
8. Develop a Python class to implement a Queue using list and perform Enqueue and Dequeue operations.
9. Write a Python program to implement a Circular Queue using a list or class.
10. Implement a Singly Linked List in Python with methods to insert, delete, and display nodes.

11. Write a Python program to create a Binary Search Tree (BST) and perform Inorder, Preorder, and Postorder traversals.
12. Create a Python program to insert elements into an AVL Tree and show the balanced tree after rotations.
13. Represent a Graph in Python using Adjacency Matrix and Adjacency List.
14. Write a program to perform Breadth-First Search (BFS) and Depth-First Search (DFS) on a graph.
15. Implement Dijkstra's algorithm in Python to find the shortest path between two nodes.
16. Develop a Python program to implement Hashing with Linear Probing for collision resolution.
17. Write a Python program to create a Hash Table using a dictionary and allow Insert, Search, and Delete operations.
18. Implement Linear and Binary Search algorithms in Python and compare their number of comparisons.
19. Write a Python program to implement Quick Sort and Merge Sort. Compare their time complexity using input of different sizes.
20. Design and implement a capstone project in Python that uses appropriate data structures to manage student records (insert, search, update, delete).

References:

1. Harsh Basin (2023). Data Structures with Python. Bpb Publication.
2. Dr. Shriram K. Vasudevan, Abhishek S Nagarajan, Karthick Nanmaran, Data Structures using Python, OUP India, Oxford University Press.

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Inter-thread communication. vectors, lists, maps.

Labs / Practicals:

Preparing

References:

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

Course Code	Course Name	L	T	P	Cr
DCSE250041	Fundamentals of Database Systems	3	0	2	4

Course Outcomes:

CO1: Explain database concepts, architecture, data models, ER/EER modeling, and construct ER/EER diagrams.
CO2: Apply relational model concepts and relational algebra operations to formulate database queries.
CO3: Develop SQL queries for data definition, manipulation, joins, nested queries, and set operations.
CO4: Perform ER/EER to relational mapping and apply normalization techniques up to BCNF using functional dependencies.
CO5: Analyze transaction management, concurrency control mechanisms, and recovery techniques in database systems.

Module 1:

Introduction to databases: Characteristics of the database approach, advantages over file-based systems. Database system architecture, data independence, and data abstraction. Data models, Entity-Relationship (ER) modeling – entity types, attributes, relationships, constraints, and ER diagrams.
Enhanced ER (EER) model concepts – specialization, generalization, and aggregation.

Module 2:

Relational model concepts: Schema, tuple, attribute, domain, keys, and integrity constraints.
Relational Algebra operations: Select, Project, Join, Union, Intersection, Set Difference, Cartesian Product, Rename.
Understanding expressions and queries using relational algebra.

Module 3:

SQL basics: Data Definition Language (DDL), data types, and constraints (Primary Key, Foreign Key, Not Null, Unique, Check).
Data Manipulation Language (DML): Retrieval using SELECT, conditions, ordering, and grouping.
INSERT, UPDATE, DELETE statements.
Nested queries, joins (inner, outer), and set operations.
Creating and managing tables.

Module 4:

ER/EER to relational mapping: Steps and examples.
Functional dependencies and inference rules.
Normalization: First Normal Form (1NF), Second (2NF), Third (3NF), and Boyce-Codd Normal Form (BCNF).

Module 5:

Introduction to transaction processing: Concept of transaction, ACID properties.
Concurrency control: Locking protocols, deadlock handling, and prevention, Database recovery techniques.

Labs / Practicals:

Consider two tables: Student(StudentID, Name, Age, DeptID, Marks)
Departments(DeptID, Department, HOD)

1. Write an SQL query to display all the records from the Students table.
2. Write a query to display the names and marks of students who scored more than 80 marks.
3. Write a query to display the list of students who belong to the 'IT' department and are younger than 22.

4. Write a query to list all students whose names start with the letter 'A'.
5. Write an SQL query to display students ordered by their marks in descending order.
6. Write a query to find students who scored marks between 70 and 90.
7. Write an SQL query to display student names and corresponding HOD names using a JOIN on the Department column.
8. Write a query to list students from either 'CSE' or 'IT' departments.
9. Write a query to display the average marks scored by students in each department using GROUP BY.
10. Write a query to display departments where the average student marks are greater than 80 using GROUP BY and HAVING clause.

References:

1. A.K. Majumdar, P. Bhattacharyya, Database Management Systems, Tata McGra-Hill, 1996
2. H. Korth, A.Silberschatz, Database System Concepts, McGraw-Hill (Second Edition), 1991
3. R. Elmasri, S. Navathe, Fundamentals of Database System, Benjamin Cummings (Second Edition),. 1994
4. Bipin Desai, An Introduction to Database Systems, Galgotai Publication (West Publishing), 1991
5. F. Mc-Fadden, J. Hoffer, Modern Database Management, Benjamin cummings (Narosa), (Fourth Edition), 1994

Course Code	Course Name	L	T	P	Cr
DASH250017	Graph theory	3	1	0	4

Course Outcomes:

CO1: Explain the fundamental concepts of graphs, types, and their properties.
CO2: Apply graph traversal algorithms and solve connectivity and path problems.
CO3: Analyze trees, spanning trees, and their applications in communication networks.
CO4: Solve problems related to coloring, matching, planarity, and network flows using appropriate algorithms.
CO5: Model real-life problems (routing, scheduling, social networks) using graph theory and find optimal solutions.

Module 1:

Fundamentals of Graphs and Representations (9 hours)
Definition of graphs, directed and undirected graphs, simple graphs, multigraphs, subgraphs, degree of a vertex, special graphs (complete, cycle, wheel, bipartite), graph representations – adjacency matrix, adjacency list, incidence matrix; isomorphism and homeomorphism.

Module 2:

Connectivity and Traversals (9 hours)
Paths and circuits, connected graphs, components, Eulerian and Hamiltonian paths and circuits, graph traversal – BFS and DFS algorithms with applications, applications in social networks and web crawling.

Module 3:

Trees and Spanning Trees (9 hours)
Definition and properties of trees, rooted and binary trees, applications – expression trees, decision trees; spanning trees, BFS/DFS-based spanning trees, Minimum Spanning Tree (Prim's and Kruskal's algorithms); applications in computer networks.

Module 4:

Coloring, Matching, and Covering (9 hours)
Graph coloring – vertex and edge coloring, chromatic number, applications in scheduling and register allocation; matchings – perfect, maximum matching, Hall's theorem; vertex and edge covers.

Module 5:

Advanced Topics and Applications (9 hours)
Planar graphs and Kuratowski's Theorem, dual graphs, graph algorithms for shortest paths (Dijkstra, Bellman-Ford), network flows (Ford-Fulkerson), applications in transport, power grid, bioinformatics, machine learning (graph embeddings, GNNs basics).

Labs / Practicals:

N/A

References:

1. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, PHI

2. S. Arumugam and S. Ramachandran, Invitation to Graph Theory, Scitech Publications
3. R. Balakrishnan and K. Ranganathan, A Textbook of Graph Theory, Springer/ Universities Press

Course Code	Course Name	L	T	P	Cr
DCSE250047	Analysis of Algorithms	3	1	0	4

Course Outcomes:

CO1: Introduce fundamental algorithm design techniques such as iterative, divide and conquer, dynamic programming, and greedy approaches.
CO2: Develop skills in applying sorting and searching algorithms with correctness proofs and computational complexity analysis.
CO3: Equip learners with advanced analysis methods, including amortized analysis, accounting and potential methods.
CO4: Provide understanding of classical graph algorithms, their applications, and performance evaluation.
CO5: Strengthen problem-solving abilities through greedy methods applied to optimization problems like scheduling, spanning trees, and shortest paths.

Module 1:

Algorithm Design Techniques: Iterative technique: Applications to Sorting and Searching (review), their correctness and analysis. Divide and Conquer: Application to Sorting and Searching (review of binary search), mergesort, quick sort, their correctness and analysis. Dynamic Programming: Application to various problems (for reference; Weighted Interval Scheduling, Sequence Alignment, Knapsack), their correctness and analysis. Greedy Algorithms: Application to various problems, their correctness and analysis.

Module 2:

More on Sorting and Searching: Heapsort, Lower Bounds using decision trees, sorting in Linear Time -Bucket Sort, Radix Sort and Count Sort, Medians &Order Statistics, complexity analysis and their correctness.

Module 3:

Advanced Analysis Technique: Amortized analysis, Accounting method or Taxation Method, potential Method.

Module 4:

Graphs: Graph Algorithms-Breadth First Search, Depth First Search and its Applications.

Module 5:

Greedy method: General method, application-job sequencing with deadlines, Knapsack problem, Minimum cost spanning trees, single source shortest path problem.

Labs / Practicals:

NA

References:

1. Kleinberg, J., &Tardos, E. (2013).Algorithm Design. 1st edition. Pearson Education India.
2. Cormen, T.H.,Leiserson,C.E. Rivest, R.L.,&Stein, C.(2015).Introduction to Algorithms.3rd edition. PHI.
3. Sarabasse&GlederA.V. (1999).Computer Algorithm– Introduction to Design and Analysis.3rd edition. Pearson Education

Course Code	Course Name	L	T	P	Cr
DASH250011	Probability and Statistics	3	1	0	4

Course Outcomes:

CO1: Apply the fundamental concepts of probability, including conditional probability and Bayes' theorem to solve real-world problems in data science.
CO2: Analyze discrete and continuous random variables, evaluate their probability distributions and compute associated statistical measures.
CO3: Apply descriptive statistical measures, correlation, and regression techniques to interpret datasets in analytics contexts.
CO4: Perform statistical hypothesis testing and construct confidence intervals to assess the reliability of conclusions.
CO5: Apply the Law of Large Numbers, Central Limit Theorem, and basics of Markov chains to model and predict outcomes in
computational and machine learning applications.

Module 1:

Introduction to Probability (9 Hours)
Sample space and events, classical and axiomatic definitions, conditional probability, independent events, Bayes' theorem, applications in data science.

Module 2:

Random Variables and Distributions (9 Hours)
Discrete and continuous random variables, probability mass function (PMF) and probability density function (PDF), cumulative distribution function (CDF), expectation, variance, common distributions: Binomial, Poisson, Normal, Exponential.

Module 3:

Descriptive Statistics (9 Hours)
Measures of central tendency: mean, median, mode, dispersion: range, variance, standard deviation, moments and skewness, correlation and linear regression, applications in business analytics and performance modeling.

Module 4:

Statistical Inference and Hypothesis Testing (9 Hours)
Sampling techniques, estimation: point and interval, hypothesis testing: null and alternative hypotheses, Z-test, t-test, chi-square test, errors: Type I and II, confidence intervals, case studies from computing applications.

Module 5:

Applied Probability and Statistical Tools (9 Hours)
Law of Large Numbers; Central Limit Theorem (CLT), Markov chains: basics and applications, introduction to statistical machine learning concepts.

Labs / Practicals:

N/A

References:

1. N.G. Das- Statistical Methods (Combined Volume I & II), McGraw Hill Education (India) Pvt. Ltd.

2. S.C. Gupta & V.K. Kapoor – Fundamentals of Mathematical Statistics, Sultan Chand & Sons.
3. P. R. Vittal – Probability and Statistics, Margham Publications.
4. R.S. Narsingh Deo – Probability and Statistics, New Age International.
5. P. G. Hoel, S. Port, and C. Stone – Introduction to Probability Theory, Universal Book Stall (Indian edition).
6. V.K. Rohatgi and A.K. Md. Ehsanes Saleh – An Introduction to Probability and Statistics, Wiley India.

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
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
DASH250052	Environmental Science	2	0	0	2

Course Outcomes:

CO1: Understand the importance and dimension of a healthy environment, become environmentally conscious, skilled and responsible in all their actions with a concern for sustainable development.
CO2: Comprehend the significance and issues related to ecosystems, natural resources and bio-diversity and become aware of the need and ways to protect/ preserve them.
CO3: Grasp the issues related to environmental pollution, solid waste management and climate change, and become conscious and proactive in the discharge of their responsibilities towards the environment.
CO4: Become aware and appreciate the values and concerns of environmental movements and policies and the role of communities, and act responsibly on environment-related issues.
CO5: Understand environmental policies, ethics, and community roles in sustainable development.

Module 1:

Introduction to Environmental Studies and Ecosystems:
Environmental Studies: Importance and scope, Multidisciplinary nature, Concept of sustainability and sustainable development, Ecosystems: Concept, Structure and Function, Pond and forest ecosystems, Food chains and food webs, Concept of ecological succession, Bio-geographical zones of India, Levels of biological diversity: Genetic, Species, and Ecosystem.

Module 2:

Biodiversity and Conservation:
Biodiversity Hotspots with special reference to India, Threats to biodiversity, Conservation of biodiversity: In-situ and Ex-situ, Endangered and endemic species: Concept Afforestation: Social forestry, Agroforestry, Green belt.

Module 3:

Pollution and Waste Management:
Air, water, and noise pollution: Causes, effects, and control measures, Climate change, global warming, ozone depletion, acid rain: Impact on humans and agriculture, Solid waste management: biodegradable and non-biodegradable waste.
- Segregation of domestic waste at source
- Impact of plastic on human and animal health

Module 4:

Natural Resources and Disaster Management:
Land resources and land-use changes, Land degradation, soil erosion, and desertification, Water: Use and over-exploitation, Water conservation : Rainwater harvesting, Watershed management : Meaning and importance, Energy resources: Renewable and non-renewable, Alternate energy sources, Disaster management – Types and self-protection during disasters Environmental Policy, 2006 : Provisions and importance; Environmental Impact Assessment – Concept; Swachh Bharat Mission Objectives; International agreements , Montreal and Kyoto protocols.

Module 5:

Environmental Policies and Human Role
Population growth: Impact on environment, health, and welfare, Environmental ethics : Religion and cultural role, Environmental movements: Chipko, Narmada Bachao, Silent Valley, Bishnoi's, Community initiatives:
Salu Marada Thimmakka, Sacred Groves (Devarakadu), Environment Protection Act, Biodiversity Act (2002), National Environmental Policy (2006), Environmental Impact Assessment, Swachh Bharat Mission, International agreements: Montreal and Kyoto Protocols.

Labs / Practicals:

NA

References:

1. Asthana, D. K. .Text Book of Environmental Studies. S. Chand Publishing.
2. Basu, M., Xavier, S. . Fundamentals of Environmental Studies, Cambridge University Press, India
3. Basu, R. N. Environment. University of Calcutta, Kolkata
4. Bharucha, E. Textbook of Environmental Studies for Undergraduate Courses. Universities Press, India.
5. De, A.K. Environmental Chemistry, New Age International, .
6. Mahapatra, R., Jeevan, S.S., Das, S. (Eds) . Environment Reader for Universities, Centre for Science and Environment.
7. Masters, G. M., &Ela, W. P. .Introduction to environmental engineering and science. Englewood Cliffs, NJ: Prentice Hall.
8. Odum, E. P., Odum, H. T., & Andrews, J. Fundamentals of ecology. Philadelphia: Saunders.
9. Sharma, P. D., & Sharma, P. D. .Ecology and environment. Rastogi Publications.

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
DASH250001	Mathematical Foundations for AI	3	1	0	4

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

N/A

References:

1.S. Kumaresan – Linear Algebra: A Geometric Approach, PHI Learning.
2.B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers.
3.Pankaj Jalote – Mathematical Foundations of Computer Science, Wiley India.
4.K. D. Joshi – Foundations of Discrete Mathematics, New Age International.
5.Marc Deisenroth, A. Faisal, C. Ong – Mathematics for Machine Learning, Cambridge University Press.
6.Shai Shalev-Shwartz & Ben-David – Understanding Machine Learning: From Theory to Algorithms.

Course Code	Course Name	L	T	P	Cr
DCSE250034	Data Communication and Computer Networks	3	1	0	4

Course Outcomes:

CO1. Describe the hardware, software components of a network and their interrelations.
CO2. Compare OSI and TCP/IP network models.
CO3. Describe,analyze and compare different protocols at the data link, network, and transport layer.
CO4. Design/implement data link and network layer protocols in a simulated networking environment.
CO5. Explain and apply network security concepts, tools, safe practices, emerging technologies, and cyber hygiene measures.

Module 1:

Module 1:
Basics of Networking and Communication:
Introduction to Computer Networks: Definition, Importance, and Real-world Applications, Types of Computer Networks, Network Topologies, Network Models: OSI Model, TCP/IP Model, Switching Techniques: Circuit Switching, Packet Switching, Message Switching.

Module 2:

Module 2:
Physical and Data Link Layer:
Analog Signal, digital signal, encoding techniques, digital modulation techniques, Transmission Media, and multiplexing (frequency division, time divion and wavelength division multiplexing).
Framing Methods, Error Detection Techniques, Flow Control Protocols, Medium Access Control: ALOHA, CSMA/CD, Ethernet Basics and IEEE 802.3, Wireless LAN (Wi-Fi) : Basic operation and use cases.

Module 3:

Module 3:
Network Layer and IP Addressing:
Functions of Network Layer: Packet Forwarding, Logical Addressing, IP Addressing: IPv4, IPv6, Subnetting, Basic Routing Concepts: Static vs Dynamic Routing, Networking Devices: Hub, Switch, Router, Bridge, Gateway

Module 4:

Module 4:
Transport and Application Layers:
Transport Layer Basics: TCP, UDP, Ports and Sockets, Common Application Layer Protocols: HTTP, FTP, SMTP, POP3, DNS, Overview of the Internet and Web Browsing: URL and Domain Name, How Browsers Communicate with Servers, Introduction to IP and DNS Resolution.

Module 5:

Module 5:
Introduction to Network Security and Emerging Trends:
Basic Network Security Concepts: Importance of Security in Networks, Common Threats: Virus, Malware, Phishing, Security Tools: Introduction to Firewalls and Antivirus Software, Basic Safety Practices in Networking: Strong Passwords, Use of Secure Websites, Network Hygiene, Emerging Trends:5G Networks, IoT (Internet of Things), Cloud Computing, Cyber Hygiene: Safe browsing, protecting digital identity, Online Threats Awareness: Social Engineering, Fake Wi-Fi Networks.

Labs / Practicals:

NA

References:

1. "Data Communications and Networking" – Behrouz A. Forouzan McGraw Hill Education.
2. "Computer Networks" – Andrew S. Tanenbaum, David Wetherall Pearson Education.
3. "Computer and Communication Networks" – Nader F. Mir
Pearson Education.
4. "Introduction to Computer Networks" – Tata McGraw Hill or Technical Publications (local authors).

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
DOAI250029	Artificial Intelligence with Python	2	0	4	4

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

List of 10 Practical Exercises

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

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

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

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

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

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

7.Load an image and perform

a. Convert to Gray Scale.

b. Apply a Gaussian blur filter.

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

d. Display the original and processed image.

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

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

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

References:

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

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

3."Deep Learning with Python" by François Chollet

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

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

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

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

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

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

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

Course Code	Course Name	L	T	P	Cr
DCSE250033	Introduction to Theory of Computation	3	1	0	4

Course Outcomes:

CO1: Explain fundamentals of automata theory, formal languages, language operations, and Chomsky language hierarchy.
CO2: Apply finite automata concepts to recognize regular languages and perform NFA-DFA conversions.
CO3: Design Mealy/Moore machines, use regular expressions, and apply pumping lemma for regular languages.
CO4: Construct and simplify context-free grammars, apply CNF/GNF, and analyze pushdown automata properties.
CO5: Explain Turing machine concepts, language decidability, halting problem, and models of computation.

Module 1:

Module 1: Languages:
Fundamentals of Automata Theory, Introduction to Formal Laguages, Alphabets, string, language, Basic Operations on language,Union, Concatenation, Kleene Closure, Chomsky Hierarchy of Languages.

Module 2:

Module 2: Finite Automata and Regular Languages:
Finite Automata :Definition and it Application, Regular Languages and their properties, Regular languages and their relationship with finite automata, Deterministic and non-deterministic finite automata, Epsilon NFA, NFA to DFA Conversion.

Module 3:

Module 3: Finite Automata with Output and Regular Expression:
Finite Automata with Output: Mealy and Moore Machines, Conversion between Mealy and Moore, Regular Expressions, Transition Graph, Kleene's Theorem, NFA or DFA to RE (State Elimination Method),Relation between FA and Regular Expression,Pumping lemma and closure properties of regular languages.

Module 4:

Module 4: Context free languages:
Context free grammars, Derivations (leftmost, rightmost),parse trees, ambiguities in grammars and languages, Simplification of CFG,Chomsky Normal Form: Chomsky Normal Form(CNF),Greiback Normal Form(GNF), Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma, Properties and Closure Properties of CFL.

Module 5:

Module 5:Turing Machines and Models of Computations:
Turing Machine as a model of computation, Universal Turing Machine, Language acceptability, decidability, halting problem, Recursively enumerable and recursive languages, insolvability problems.

Labs / Practicals:

NA

References:

1. K.L.P.Mishra,N.Chandrasekaran, Theory of Computer Science Automata,language and computation, Third Edition, PHI.
2. Michael Sipser, Introduction to the Theory of Computation, Third Edition, Cengage Learning.
3. Hoperoft, Aho, Ullman, Introduction to Automata theory, Language & Computation –3rd Edition, Pearson Education.

4. P. Linz, An Introduction to Formal Language and Automata 4th edition Publication Jones Bartlett.
5. Daniel I.A.Cohen, Introduction to computer theory, John Wiley.
6. Lewis & Papadimitriou, Elements of the theory of computation, PHI.

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
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
DASH250091	Professional Values and Ethics	2	0	0	2

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

N/A

References:

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

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

preparing

References:

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

Course Code	Course Name	L	T	P	Cr
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
DASH250070	Management Information System	3	1	0	4

Course Outcomes:

CO1: Understand the structure and components of management information systems.
CO2: Apply information systems in functional areas such as manufacturing, telecom, and decision-making.
CO3: Analyze data flow, system design, and ERP solutions for real-time industrial scenarios.
CO4: Evaluate security, privacy, and ethical challenges in MIS deployment.
CO5: Use MIS concepts in project planning, automation, and digital transformation initiatives.

Module 1:

Introduction to MIS:
Definition, characteristics, and components of MIS; types of information systems – TPS, DSS, ESS, and MIS; role of MIS in technical organizations; impact on planning, controlling, and decision-making; Indian examples in telecom and smart cities.

Module 2:

Information System Components and Design:
System architecture—hardware, Software, Database, Network, User Interface; System Development life cycle (SDLC); Input-Output design, Data flow diagrams (DFD); Introduction to Relational Databases; Basics of SQL and data models.

Module 3:

MIS in Functional Areas:
MIS in production, HR, Finance, and Telecom service management; case study — MIS in mobile network monitoring, Inventory control, Customer Relationship Management (CRM); Supply Chain Management (SCM); Real-time data handling in electronics.

Module 4:

Enterprise Systems and Decision Support:
Enterprise Resource Planning (ERP); integration with MIS; DSS and AI-enabled decision-making; overview of cloud-based MIS; business analytics and dashboards; mobile MIS and IoT-based systems in Indian telecom and electronics sectors.

Module 5:

Security, Ethics, and Future Trends:
Cybersecurity, authentication and access control; information system auditing; ethical use of MIS; data privacy, IT Act (India), and regulatory guidelines; trends – MIS with AI/ML, blockchain, and automation; MIS roles in Digital India and Industry 4.0.

Labs / Practicals:

N/A

References:

1. Waman S. Jawadekar – Management Information System, McGraw Hill Education
2. D.P. Goyal – Management Information Systems: Managerial Perspectives, Macmillan India
3. A.K. Gupta – Management Information System, S. Chand Publishing
4. S.A. Kelkar – MIS Concepts and Applications, PHI Learning
5. Mahadeo Jaiswal & Monika Mital – Management Information Systems, Oxford University Press India

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
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
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
DCSE250004	Advanced Data Structures and Algorithms	3	0	2	4

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

Preparing

References:

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

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

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

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

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

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

Course Code	Course Name	L	T	P	Cr
DCSE250024	Compiler Design	3	1	0	4

Course Outcomes:

CO1: Understand the structure and various phases of a compiler.
CO2: Apply formal grammar and parsing techniques to perform syntax analysis.
CO3: Design lexical and syntax analysers using tools like Lex and YACC.
CO4: Implement semantic analysis and intermediate code generation.
CO5: Apply optimization techniques and generate efficient machine code.

Module 1:

Introduction to Compilers & Lexical Analysis
Overview of Compiler Structure: Phases of a compiler – Lexical, Syntax, Semantic, Intermediate Code Generation, Optimization, Code Generation, Code Linking
Compiler vs Interpreter, Bootstrapping
Lexical Analysis: Role and tasks of the Lexical Analyzer, Input buffering and pre-processing, Token recognition, Regular Expressions and Finite Automata (DFA, NFA), Conversion of Regular Expression to NFA and NFA to DFA, Minimization of DFA
Lex Tool: Introduction to Lex for generating lexical analysers with practical examples

Module 2:

Syntax Analysis (Parsing Techniques)
Role of the Parser in compilation
Context-Free Grammar (CFG): Notation, Derivations, Ambiguity
Top-Down Parsing: Recursive Descent Parser, LL(1) Grammar and Predictive Parsing
Bottom-Up Parsing: Shift-Reduce Parsing, Construction of LR Parsing Tables, LR(0), SLR(1), LALR(1), Canonical LR Parsers
Error Handling in Syntax Analysis
YACC Tool: Introduction and usage for generating parsers

Module 3:

Semantic Analysis and Intermediate Code Generation
Syntax-Directed Definitions (SDD): Inherited and Synthesized Attributes
Syntax Directed Translation Schemes (SDTS)
Evaluation Order of Attributes (Dependency Graphs, Topological Sort)
Intermediate Code Forms: Abstract Syntax Trees (AST), Three-Address Code (TAC), Quadruples and Triples
Type Checking and Type Systems: Static vs Dynamic typing, Type Expressions, Type Conversion, Symbol Table Management

Module 4:

Run-Time Environment and Code Optimization
Storage Organization: Activation Records, Stack Allocation, Heap Allocation, Static and Dynamic Scoping
Run-Time Environment: Parameter Passing Mechanisms, Access Links, Control Links
Code Optimization: Local vs Global Optimization, Peephole Optimization, Common Subexpression Elimination, Constant Folding and Propagation, Loop Optimization Techniques

Module 5:

Code Generation and Compiler Tools

Target Code Generation: Issues in Code Generation, Basic Blocks and Flow Graphs, DAG Representation of Basic Blocks, Instruction Selection and Scheduling, Register Allocation using Graph Coloring
Introduction to Modern Compiler Tools: Overview of Lex & YACC Integration, Introduction to LLVM or GCC backend, Brief overview of Just-In-Time (JIT) Compilation

Labs / Practicals:

NA

References:

1. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2007). Compilers: Principles, techniques, and tools (2nd ed.). Pearson Education.
2. Muchnick, S. S. (1997). Advanced compiler design and implementation. Morgan Kaufmann.
3. Dhamdhere, D. M. (2012). Compiler construction: Principles and practice (2nd ed.). McGraw-Hill Education India.

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
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
DCSE250052	Introduction to Cloud Computing	3	0	2	4

Course Outcomes:

CO1. Identify and describe various computing paradigms including Grid, Cluster, Distributed, Utility, and Cloud Computing.
CO2. Explain the foundational concepts of Cloud Computing, its history, major service providers.
CO3. Analyze Cloud Computing architecture and compare it with traditional client-server models; distinguish between service models (IaaS, PaaS, SaaS) and deployment models (Public, Private, Hybrid, Community).
CO4. Demonstrate understanding through case studies of real-world cloud platforms such as Google App Engine, Microsoft Azure, Amazon EC2, and Eucalyptus.
CO5. Understand cloud security mechanisms, including network, host, and application-level security; assess challenges in data security, jurisdictional issues, and authentication in the cloud.

Module 1:

Overview of Computing Paradigm :
Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing.

Module 2:

Introduction to Cloud Computing:
Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing.

Module 3:

Cloud Computing Architecture:
Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

Module 4:

Case Studies:
Case study of Service model using Google App Engine, Microsoft Azure, Amazon EC2 Eucalyptus.

Module 5:

Cloud Security:
Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

Labs / Practicals:

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms.
3. Working on tools used in cloud computing online a) Storage b) Sharing of data c) manage your calendar, to-do lists, d) a document editing tool
4. Exploring Google cloud
5. Exploring Microsoft Azure
6. Exploring Amazon Web Services.

References:

1. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010.
2. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011.
3. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz,Russell Dean Vines, Wiley-India, 2010.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, Cloud Computing, A Practical Approach, McGraw Hills, 2010.

Course Code	Course Name	L	T	P	Cr
DCSE250074	Programming Algorithms	3	1	0	4

Course Outcomes:

After the completion of this course, the student will:

- Understand the fundamentals of algorithms, their structure, and performance metrics including time and space complexity
- Analyze problems and apply suitable algorithmic techniques such as Divide and Conquer, Greedy, Dynamic Programming, and Backtracking.
- Demonstrate proficiency in using data structures like arrays, linked lists, trees, graphs, heaps, and hash tables for problem-solving.
- Design and implement searching and sorting algorithms efficiently with an understanding of their time complexities.
- Apply graph-based algorithms (BFS, DFS, Dijkstra, Kruskal, etc.) for solving real-world problems involving networks and connectivity.

Module 1:

1. Introduction to Algorithms:

Algorithm Definition:Understanding what an algorithm is and its role in problem-solving.
Algorithm Analysis: Learning how to analyze the efficiency of algorithms using time and space complexity.
Asymptotic Notation: Mastering Big O, Omega, and Theta notations to describe algorithm performance.
Recurrences: Understanding and solving recurrence relations, which often arise in the analysis of recursive algorithms.

Module 2:

Common Algorithm Design Techniques:
Divide and Conquer: Breaking down problems into smaller subproblems, solving them recursively, and combining the solutions. Examples include Merge Sort and Quick Sort.
Greedy Algorithms: Making locally optimal choices at each step with the hope of finding a global optimum. Examples include Kruskal's and Prim's algorithms for Minimum Spanning Trees.
Dynamic Programming: Solving problems by breaking them into overlapping subproblems and storing the solutions to avoid redundant calculations. Examples include the Knapsack problem and the Longest Common Subsequence problem.
Backtracking: Exploring all possible solutions by systematically trying and undoing choices. Examples include the N-Queens problem and the Subset Sum problem.

Module 3:

Data Structures:
Arrays: Understanding how to store and access elements in a contiguous block of memory.
Linked Lists: Understanding how to store and access elements in a non-contiguous block of memory, linked together by pointers.
Trees: Understanding hierarchical data structures and their applications, such as Binary Search Trees and AVL Trees.
Graphs: Understanding how to represent relationships between data using nodes and edges.
Heaps: Understanding priority queue data structures and their applications, such as Heap Sort.
Hashing: Understanding how to use hash tables for efficient data storage and retrieval.

Module 4:

Searching and Sorting:

Linear Search: Searching for an element by sequentially checking each element in a list.
Binary Search: Searching for an element in a sorted list by repeatedly dividing the search interval in half.
Sorting Algorithms: Understanding various sorting algorithms like Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort.

Module 5:

Graph Algorithms:
Graph Representation: Understanding how to represent graphs using adjacency matrices or adjacency lists.
Graph Traversal: Exploring graphs using Depth-First Search (DFS) and Breadth-First Search (BFS).
Shortest Path Algorithms: Understanding algorithms like Dijkstra's and Bellman-Ford for finding shortest paths in graphs.
Minimum Spanning Tree Algorithms: Understanding algorithms like Kruskal's and Prim's for finding minimum spanning trees.

Labs / Practicals:

NA

References:

1. Design and Analysis of Algorithms, S.Sridhar, Oxford University Press
2. Design and Analysis of Algorithms, R. Panneerselvam, PHI Learning
3. Data Structures and Algorithms Made Easy, Narasimha Karumanchi, CareerMonk Publications
4. Data Structures Through C in Depth, S.K. Srivastava & Deepali Srivastava, BPB Publications

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
DOAI250015	Data Analytics and Visualization	2	0	4	4

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

References:

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

Course Code	Course Name	L	T	P	Cr
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
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
DOAI250035	Introduction to Computer Vision	3	1	0	4

Course Outcomes:

CO1: Understand the fundamentals of computer vision, image formation, and basic image representations.
CO2: Apply feature detection and matching techniques for image analysis and object matching.
CO3: Analyze depth and motion using stereo disparity, optical flow, and feature tracking methods.
CO4: Implement object and face detection/recognition techniques using classical computer vision methods.
CO5: Apply computer vision techniques to real-world applications such as OCR, segmentation, gesture recognition, and object counting.

Module 1:

Module 1:
Foundations of Computer Vision:
What is Computer Vision? Applications and scope, Basic Image Concepts:Pixels, Resolution, Color Spaces (RGB, Grayscale, HSV), Introduction to Image Formation: Pinhole Camera Model, Perspective Projection, Image Representation and File Formats.

Module 2:

Module 2:
Feature Detection and Matching:
What is a Feature in an Image? Detecting Features:Point features (Harris Corner Detector), Edges (Sobel, Canny), Lines (Hough Transform), Feature Matching: Template Matching, Basic Keypoint Matching (using ORB in OpenCV), Applications: Panoramas, Simple Object Matching.

Module 3:

Module 3:
Shape and Depth Perception :
Introduction to 3D Perception from 2D Images, Stereo, Depth using Stereo Disparity, Optical Flow, Feature Tracking:Lucas-Kanade Tracker, Applications: Object tracking, motion detection.

Module 4:

Module 4:
Object and Face Recognition:
Image Classification vs. Object Detection, Object Detection Techniques: Haar Cascade Classifier (OpenCV), Contour detection and bounding boxes, Face Detection and Recognition, Introduction to Pre-trained Classifiers, Scene Understanding.

Module 5:

Module 5:
Applications and Case Studies:
OCR (Optical Character Recognition) using Tesseract, Barcode/QR Code detection, Image Segmentation: Thresholding, Contours, Simple Object Counter using contours, Introduction to Gesture Recognition.

Labs / Practicals:

NA

References:

1. “Computer Vision: A Modern Approach” by David A. Forsyth and Jean Ponce
2. “Computer Vision: Algorithms and Applications” by Richard Szeliski
3. “Digital Image Processing” by Rafael C. Gonzalez & Richard E. Woods
4. “Practical Python and OpenCV” by Adrian Rosebrock

5. “Programming Computer Vision with Python” by Jan Erik Solem
6. “Practical Python and OpenCV” by Adrian Rosebrock

Course Code	Course Name	L	T	P	Cr
DOAI250036	Introduction to Deep Learning	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of neural networks, perceptrons, and deep network basics.
CO2: Apply feedforward deep neural networks with various activation and loss functions.
CO3: Implement optimization techniques including gradient descent and its variants for training deep models.
CO4: Analyze overfitting, bias-variance tradeoff, and apply regularization methods to improve model performance.
CO5: Apply deep learning algorithms such as Autoencoders, CNNs, and RNNs for practical applications.

Module 1:

Module 1:
Fundamentals of Neural Network :
Biological neuron, Mc-Culloch Pitts Neuron, Perceptron, Perceptron Learning, Multilayer Perceptron: Linearly separable, linearly non-separable classes. Deep Networks Fundamentals: Brief History, Three Classes of Deep Learning, Basic Terminologies of Deep Learning.

Module 2:

Module 2:
Fundamentals of Deep Neural Network :
Training Feedforward DNN : Feed Forward Neural Network, Activation functions: Tanh, Logistic, Linear, Softmax, ReLU, Leaky ReLU, Loss functions: Squared Error loss, Cross Entropy.

Module 3:

Module 3:
Optimization:
Learning with back propagation, Learning Parameters: Gradient Descent (GD), Stochastic and Mini Batch GD, Momentum Based GD, Nesterov Accelerated GD, AdaGrad, Adam, RMSProp.

Module 4:

Module 4:
Regularization:
Overview of Overfitting, Types of biases, Bias Variance Tradeoff Regularization Methods: L1, L2 regularization, Parameter sharing, Dropout, Weight Decay, Batch normalization, Data Augmentation, Adding noise to input and output.

Module 5:

Module 5:
Deep Learning Algorithms:
Autoencoder: Introduction, Linear Autoencoder, Undercomplete Autoencoder, Overcomplete Autoencoders,Regularization in Autoencoders.. Application of Autoencoders: Image Compression Convolutional Neural Networks (CNN): Introduction, Convolution operation, Padding, Stride, Relation between input, output and filter size, CNN architecture: Convolution layer, Pooling Layer, Weight Sharing in CNN Recurrent Neural Networks (RNN): Sequence Learning Problem, Unfolding Computational graphs, Recurrent Neural Network,Bidirectional RNN, Backpropagation Through Time (BTT), Vanishing and Exploding Gradients, Truncated BTT.

Labs / Practicals:

1. Implement Perceptron algorithm to simulate any logic gate (NOT, OR, AND, NOR, and NAND).
2. Implement Multilayer Perceptron algorithm to simulate XOR gate.
3. Apply any of the following learning algorithms to learn the parameters of the supervised single layer feed forward neural network.
a. Stochastic Gradient Descent
b. Mini Batch Gradient Descent
c. Momentum GD
d. Nestorev GD
e. Adagrad GD
f. Adam Learning GD
4. Implement linear regression to predict house price using Boston Housing data.
5. Implement a backpropagation algorithm to train a DNN with at least 2 hidden layers.
6. Design and implement a fully connected deep neural network with at least 2 hidden layers for a classification application. Use appropriate Learning Algorithm, output function and loss function.
7. Design the architecture and implement the autoencoder model for Image Compression.
8. Design the architecture and implement the autoencoder model for Image denoising.
9. Design and implement a CNN model for digit recognition application.
10. Design and implement a CNN model for image classification.
11. Design and implement Vanilla RNN for classification on text data.

References:

1. Li Deng and Dong Yu, —Deep Learning Methods and Applications‖, Publishers Inc.
2. Heaton, J.(2015). Deep Learning and Neural Networks, Heaton Research Inc. Additional
3. Satish Kumar "Neural Networks A Classroom Approach" Tata McGraw-Hill.
4. M. J. Kochenderfer, Tim A. Wheeler. —Algorithms for Optimization‖, MIt Press.
5. Ian Goodfellow, Yoshua Bengio, Aaron Courville. —Deep Learning‖, MIT Press Ltd, 2016

Course Code	Course Name	L	T	P	Cr
DOAI250038	Introduction to Natural Language Processing	3	1	0	4

Course Outcomes:

CO1: Understand the origins, challenges, grammar, and applications of NLP.
CO2: Apply text processing techniques such as tokenization, stemming, lemmatization, and edit distance.
CO3: Build and evaluate n-gram language models with smoothing techniques.
CO4: Implement supervised text classification using Naïve Bayes and Logistic Regression with evaluation metrics.
CO5: Apply POS tagging, named entity recognition, and semantic analysis using statistical and vector-based methods.

Module 1:

Module 1:
Introduction to NLP:
Origins of NLP. Language and Knowledge. The Challenges of NLP. Language and Grammar. NLP Applications. Some Successful Early NLP Systems, Ambiguity.

Module 2:

Module 2:
Text Processing :
Regular Expressions, Text Normalization: Tokenization – Stemming – Lemmatization, Sentence Segmentation, Edit Distance.

Module 3:

Module 3:
N-gram Language Models:
N-grams - Evaluating Language Models - Sampling sentences from a language model - Generalization and Zeros – Smoothing.

Module 4:

Module 4:
Text Classification:
Supervised Text Classification - Naive Bayes, Evaluation: Precision, Recall, F-measure. Avoiding Harms in Classification. Logistic Regression – The sigmoid function - Classification with Logistic Regression. Gradient Descent.

Module 5:

Module 5:
Parts of Speech and Named Entities and Semantic Analysis:
Part-of-Speech Tagging. Named Entities and Named Entity Tagging. Markov Models. Hidden Markov Models. HMM Part-of-Speech Tagging Lexical Semantics- Word Similarity- Word Relatedness- Semantic Frames and Roles- Connotation. Vector Semantics. Words and Vectors- Document Dimensions- Word Dimensions. Cosine for Measuring Similarity. TF IDF

Labs / Practicals:

NA

References:

1. Daniel Jurafsky, James H. and Martin, “Speech and Language Processing”,2023, Third Edition,Pearson.
2. Siddiqui and Tiwary U.S.,”Natural Language Processing and Information Retrieval”,2008, Oxford University.
3. Manning, Christopher, and Hinrich Schutze. “Foundations of statistical natural language processing”. MIT

press, 1999.

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

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

References:

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