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

M.Sc - IT

(2026-27 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

The Master of Science in Information Technology program aims to provide a strong foundation in advanced computing concepts, fostering expertise in Object-Oriented Programming (OOPs), Database Management Systems (DBMS), Operating Systems (OS), Data Structures & Algorithms, Software Engineering, and Computer Networks. Through hands-on lab experiences in DBMS, Computer Networks, and Computer Organization & Architecture (COA), we prepare students for real-world problem-solving and research-driven innovation. Our mission is to nurture technically proficient, industry-ready professionals who can contribute to the evolving IT landscape with ethical responsibility, analytical thinking, and innovative solutions.

Mission

To be a leading academic program that empowers students with cutting-edge knowledge and practical skills in computing, enabling them to excel in IT research and software development. We envision producing graduates who drive technological advancements, contribute to digital transformation, and uphold ethical computing principles while solving complex challenges in academia, industry, and society.

Program Education Objectives

PEO1: Graduates leverage foundational and advanced concepts in information technology can establish themselves as IT professionals, software developers, data analysts, system architects, or researchers in academia and industry.
PEO2: Graduates demonstrate proficiency in emerging technologies and advancements in areas such as Object-Oriented Programming (OOPs), Database Management Systems (DBMS), Operating Systems (OS), and Computer Networks.
PEO3: Graduates exhibit effective leadership roles in IT companies, startups, or research organizations, demonstrating technical expertise and strategic decision-making.

Program Outcomes

PSO1: Graduates apply Object-Oriented Programming (OOPs), Software Engineering principles, and advanced programming techniques to develop secure, efficient, and scalable software applications.
PSO2: Graduates can design, develop, and manage relational and non-relational databases using Database Management Systems (DBMS) and optimize data storage, retrieval, and security.
PSO3: Graduates can understand and apply concepts of Operating Systems (OS) and Computer Networks to enhance system performance, security, and network communication.
PSO4: Graduates can demonstrate hands-on expertise through DBMS Lab, Computer Network Lab, and other practical sessions, ensuring proficiency in designing, testing, and deploying IT solutions.

Program Specific Outcomes

PSO1: Graduates apply Object-Oriented Programming (OOPs), Software Engineering principles, and advanced programming techniques to develop secure, efficient, and scalable software applications.
PSO2: Graduates can design, develop, and manage relational and non-relational databases using Database Management Systems (DBMS) and optimize data storage, retrieval, and security.
PSO3: Graduates can understand and apply concepts of Operating Systems (OS) and Computer Networks to enhance system performance, security, and network communication.
PSO4: Graduates can demonstrate hands-on expertise through DBMS Lab, Computer Network Lab, and other practical sessions, ensuring proficiency in designing, testing, and deploying IT solutions

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Open Elective Courses	OE	8
Audit Courses (without grade or credit)	AU	0
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	36
Professional Elective Courses	PE	16
Program Core Courses	PC	20
Total Credits (Common track)		**80**

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-MSCITC-101	DCSE250116	Programming Methodology	3	1	0	4
PC-MSCITC-102	DCSE250056	Data Structures and Algorithms	3	0	2	4
PC-MSCITC-103	DCSE250102	Operating Systems	3	0	2	4
PE-MSCITC-104	Elective	Program Elective				4
EE-MSCITC-105	DASH250090	Professional Skill Development	3	1	0	4
AU-MSCITC-106	Elective	Audit Elective				0

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-MSCITC-201	DCSE250035	Computer Networks	3	0	2	4
PC-MSCITC-202	DCSE250060	Database Management Systems	2	0	4	4
PE-MSCITC-203	Elective	Program Elective				4
PE-MSCITC-204	Elective	Program Elective				4
OE-MSCITC-205	Elective	Open Elective				4
AU-MSCITC-206	Elective	Audit Elective				0

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-MSCITC-301	EE	Internship				4
PE-MSCITC-302	Elective	Program Elective				4
OE-MSCITC-303	Elective	Open Elective				4
EE-MSCITC-304	EE	Mini Project				8

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-MSCITC-401	EE	Main Project				20

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4	
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4	
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4	
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4	
DCSA250054	Multimedia & Animation Techniques	3	0	2	4	104
DCSA250071	Software Engineering	3	1	0	4	104
DCSA250057	Object Oriented Programming using C++	3	0	2	4	203
DCSA250059	Object-Oriented Programming with Java	2	0	4	4	203
DCSE250079	Fundamentals of Programming in Python	2	0	4	4	204
DOAI250014	Data Analysis using Python	2	0	4	4	204
DCSE250020	Cloud Computing	3	0	2	4	302
DOAI250021	Data Science	3	0	2	4	302

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
DCSE250116	Programming Methodology	3	1	0	4

Course Outcomes:

CO1. Apply structured programming principles to solve computational problems efficiently using modular, readable, and maintainable code.
CO2. Design, implement, and test programs involving decision making, iteration, and function-based decomposition.
CO3. Analyze and debug simple to moderately complex programs to identify logical, syntax, and runtime errors.
CO4. Demonstrate the use of best practices in programming such as code documentation, indentation, and naming conventions.
CO5. Understand and explain fundamental programming concepts such as variables, data types, control structures, functions, and arrays.

Module 1:

Introduction to Programming: General Problem solving-Algorithmic & Heuristic, Steps of problem solving,
Problem solving using computer – programming; Basic programming concept – Input, Processing Data,Output Data; Data types - Variables and constants - Integer, Float, Character, Boolean, String etc.- Format of variable names; Type specific operations - arithmetic, logical, relational etc , Built-in functions for operation, Operator Precedence & Associativity, Expressions.

Module 2:

The Program Development: Development Cycle; Program Design - Modular Programming, pseudo code, flowchart; Coding, Documenting & Testing - Programming Languages – Types, Examples, Translators – Assemblers, Interpreters & Compilers, Editors, Debuggers; Coding & Documenting - Testing & Errors; Commercial Programs – Testing & Documenting, Testing phases, External Documentation; Structured Programming, Flow chart - Creating flowchart, Control Structures, Programming Conventions.

Module 3:

Introduction to Selection Structures : Single ,dual, multiple alternative structures, Constructing flow charts with selection structure, ASCII Code, Revisit Logical & Relational Operators, Hierarchy of operators, Using IF structure - Simple IF, IF-Else, Nested IF & IF-ladder, Using Case - like structures, Applications of Selection Structures, Problem solving with selection structures

Module 4:

Modules and Subprograms: Need of Sub Programs - Problems requiring modules, Problem analysis, Design, and Data flow diagrams, Arguments and Parameters, Passing data between modules, Assigning types to parameters, Value & Reference parameters, Scope of Variables - Global, Local; Functions - Revisit built-in functions, User defined functions; Recursion-Recursive Process, Simple recursion examples, Problem solving involving subprograms

Module 5:

Array: Arrays in everyday world, One dimensional Arrays, Array Basics, Working with multiple arrays, Properties and Advantages of arrays, Searching and Sorting, Linear Searching, Bubble / Sink Sort, Strings as Array of characters, String operations, Two Dimensional Arrays, Introduction to two dimensional arrays, Using Nested loops to work with two dimensional arrays, Multi Dimensional Arrays, Problem solving with One, Two, Multi-Dimensional Arrays

Labs / Practicals:

NA

References:

1. Venit, S & Drake E., Prelude to Programming : Concepts & Design, 4th Ed. , AddisonWesley (Pearson)
2. Sprankle, Maureen., Problem Solving and Programming Concepts, 7th Ed., Pearson.
3. Balaguruswamy, E., Programming in ANSI C, 5th Ed., Tata-McGraw Hill
4. Tremblay, J & Bunt, R B., Introduction to Computer Science : An Algorithmic Approach, 2nd Ed., Tata-McGraw Hill

Course Code	Course Name	L	T	P	Cr
DCSE250056	Data Structures and Algorithms	3	0	2	4

Course Outcomes:

1)Classify Data structures
2)Describe Linear Data Structures
3)Explain Non-Linear Data Structures
4)Explain basic algorithmic concepts and recursion
5)Apply different Sorting and Searching Algorithms

Module 1:

Introduction to Data Structures:Basic Terminology, Classification of Data Structures, Operations on Data Structures

Module 2:

Linear Data Structures: Arrays: Introduction to Arrays, Representation in Memory, Operations on an Array, Two Dimensional Arrays
Linked Lists: Singly Linked List, Representation in Memory, Operations on a Single Linked List, Circular Linked Lists, Doubly Linked Lists.
Stacks: Introduction to Stacks, Array Representation of Stacks, Operations on a Stack, Linked List Representation and Operations of Stack, Applications of Stacks-Infix-to-Postfix Transformation, evaluating Postfix Expressions.
Queues: Introduction to Queues, Array Representation of Queues, Operations on a Queue,
Linked List Representation and Operations of Queue, Types of Queues-DeQueue, Circular Queue, Applications of Queues-Round Robin Algorithm

Module 3:

Non Linear Data Structures:
Trees: Basic Terminologies, Definition and Concepts of Binary Trees, Representations of a Binary Tree using Arrays and Linked Lists, Operations on a Binary Tree-Insertion, Deletion, Traversals, Types of Binary Trees.
Graphs: Graph Terminologies, Representation of Graphs- Set, List, Matrix, Graph Traversals

Module 4:

Introduction to Algorithms:
Algorithms and flow charts, Time & Space complexity (definition only)
Recursion: Basic concepts and examples of recursion e.g. factorial problem, Fibonacci sequence.

Module 5:

Sorting & Searching Algorithms: Sorting Algorithms: Algorithms and their analysis (time and space) — Bubble sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap sort and Radix Sort
Searching Algorithms: Linear search — Binary search –Concept of Hashing.

Labs / Practicals:

1.Write a program to implement array
2.Write a program to add two matrices using two dimensional arrays
3.Write a program using recursive and non-recursive functions to perform search operation in a given list of integers using linear search technique
4.Write a program to implement search operation in a given list of integers using binary search technique

5.Write a program to implement insertion sorting for a given random data
6.Write a program to implement bubble sorting for a given random data
7.Write a program to implement quick sorting for a given random data
8.Write a program to implement selection sorting for a given random data.

References:

1)Data Structures, R.S. Salaria, Khanna Book Publishing, New Delhi
2)Data Structures Using C, Reema Thareja, Oxford University Press India.
3)Classic Data Structures, SamantaDebasis, Prentice Hall of India.
4)Fundamentals of Data Structure in C, Horowitz, Ellis, Sahni, Sartaj, AndersonFreed, Susan, University Press, India.
5)Data Structures: A Pseudo code approach with C, Richard F. Gilberg, Behrouz A. Forouzan, CENGAGE Learning, India.
6)Data Structures and Algorithms: Concepts, Techniques and Applications, G. A. V. Pai, McGraw- Hill Education, India.
7)Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R. L. Rivest, C. Stein, MIT Press

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

References:

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

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

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

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

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

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

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

Course Code	Course Name	L	T	P	Cr
DASH250090	Professional Skill Development	3	1	0	4

Course Outcomes:

CO1: Understand and apply professional communication skills (written, oral, and digital) in academic and industry settings.
CO2: Analyze personal strengths and behavioral patterns to enhance workplace readiness and interpersonal relationships.
CO3: Demonstrate time management, teamwork, and leadership skills required in data-driven and technical project-based environments.
CO4: Prepare industry-specific professional documents such as résumés, cover letters, and project proposals.
CO5: Evaluate workplace ethics, digital etiquette, and cultural sensitivity in alignment with current Indian and global practices.

Module 1:

Communication for Professionals
Principles of effective communication – verbal, non-verbal, written; communication models; barriers and overcoming techniques; professional writing – email, reports, memos and proposals; oral presentations using digital tools (Zoom, Google Meet, MS Teams).

Module 2:

Self-Development and Emotional Intelligence
SWOT analysis; Self-Confidence and emotional regulation; Goal setting using SMART principles; growth versus fixed mindset; understanding personality types (MBTI, Big Five); self-reflection tools; aligning behavior with academic and workplace goals.

Module 3:

Teamwork, Leadership, and Time Management
Team dynamics and roles (Belbin's framework); Leadership styles – autocratic, democratic, transformational; conflict resolution; time management frameworks – Eisenhower Matrix, Pomodoro Technique; collaborative work in technical and data science environments.

Module 4:

Employability and Career Readiness
Résumé writing, cover letter and Statement of Purpose (SOP) preparation; Mock Interview and group discussion strategies; LinkedIn profile building; networking and job search strategies; understanding Job Descriptions (JDs), KRA/KPIs, and career mapping in core and IT/data roles.

Module 5:

Professional Ethics, Etiquette, and Digital Presence
Workplace values – integrity, accountability, respect; etiquette in email, meetings, and telephonic communication; digital footprint and cybersecurity hygiene; ethics in technology (AI/ML bias, data privacy); Diversity, Equity, and Inclusion (DEI) in modern workplaces.

Labs / Practicals:

N/A

References:

1. Barun K. Mitra – Personality Development and Soft Skills, Oxford University Press
2. Meenakshi Raman & Sangeeta Sharma – Technical Communication: Principles and Practice, Oxford University Press
3. K. Alex – Soft Skills: Know Yourself and Know the World, S. Chand Publishing

4. Ashraf M. Rizvi – Effective Technical Communication, McGraw Hill India
5. R.C. Sharma & Krishna Mohan – Business Correspondence and Report Writing, Tata McGraw Hill

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

Preparing

References:

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

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

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

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

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

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

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

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

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

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

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

References:

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

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

References:

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

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

References:

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

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

Labs / Practicals:

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

References:

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

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

References:

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

References:

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

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

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

Course Outcomes:

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

Module 1:

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

Module 2:

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

Module 3:

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

Module 4:

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

Module 5:

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

Labs / Practicals:

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

References:

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

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

Course Outcomes:

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

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

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

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

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

Module 1:

Introduction to Cloud Computing

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

Module 2:

Introduction to component virtualization

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

Module 3:

Architectural Design of Compute and Storage Clouds

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

Module 4:

Parallel and Distributed Programming Paradigms

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

Module 5:

Security Overview

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

Labs / Practicals:

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

References:

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

Course Code	Course Name	L	T	P	Cr
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