SNDT Women's University, Mumbai

Post-Graduate Diploma in Computer Science and Application

(PGDCSA)

As Per NEP - 2020

Syllabus

w.e.f.

A.Y.: 2025-2026

1, NathibaiThackersey Road,Mumbai- 400020

www.sndt.ac.in

Programme Template:

Programme Degree		Post-Graduate Diploma in Computer Science and Application (PGDCSA)
Parenthesis if any (Specialization) e.g. History, Human Development, English, etc.		NA
Preamble (Brief Introduction to the programme)		The Post-Graduate Diploma in Computer Science and Application is a rigorous academic program designed to provide advanced knowledge and skills in the field of computer science and its applications. This program is crafted to cater to the evolving needs of the industry and academia, equipping students with the expertise required to excel in a dynamic technological landscape. Through a blend of theoretical foundations and practical hands-on experience, this diploma aims to foster critical thinking, problem-solving abilities, and innovative approaches to tackle real-world challenges in computer science and its diverse applications. Recognizing the interdisciplinary nature of modern computing, the curriculum encompasses a wide range of topics including but not limited to computer programming, algorithms and data structures, database systems, software engineering principles, computer networks, artificial intelligence, machine learning, and cybersecurity. Moreover, this diploma emphasizes the importance of ethical considerations, professionalism, and lifelong learning, empowering graduates to contribute responsibly and ethically to the advancement of technology and society. With a faculty comprising experienced academics and industry professionals, state-of-the-art facilities, and a conducive learning environment, this program offers a comprehensive educational experience that prepares graduates for successful careers in various sectors such as software development, IT consulting, research, and academia. By undertaking the Post-Graduate Diploma in Computer Science and Application, students embark on a transformative journey of intellectual growth, skill development, and career advancement, poised to make significant contributions to the ever-expanding field of computer science and its applications.
Programme Outcomes (POs)		**After completing this programme, Learner will**

	1	**Knowledge Application** Apply the knowledge of computer science fundamentals, computing principles, and practical skills to solve real-world problems across various domains such as programming, algorithms, software engineering, and artificial intelligence.
	2	**Problem Analysis** Identify, analyze, and formulate computing problems using logical reasoning, critical thinking, and appropriate methodologies to arrive at effective and efficient solutions.
	3	**Software Solution Design** Design and develop software systems that meet desired specifications and constraints related to functionality, usability, reliability, scalability, and security using modern tools and practices.
	4	**System Evaluation and Optimization** Analyze and evaluate computer-based systems, applications, and processes to identify areas of improvement, and propose solutions that enhance performance and user experience.
	5	**Communication Skills** Effectively communicate technical information, project outcomes, and solutions through oral presentations, written documentation, and visual representations to both technical and non-technical audience
Programme Specific Outcomes (PSOs)		**After completing this programme, Learner will**
	1.	**Demonstrate Advanced Knowledge:** Exhibit a deep understanding of theoretical principles and practical applications across various domains of computer science including programming, algorithms, data structures, software engineering, and artificial intelligence.
	2.	**Apply Problem-Solving Skills:** Utilize analytical and critical thinking skills to identify, formulate, and solve complex problems in computer science and its diverse applications, employing appropriate methodologies and tools.
	3.	**Design and Develop Software Solutions:** Design, develop, and implement software solutions using industry-standard practices and methodologies, considering factors such as scalability, reliability, and security.
	4.	**Analyze and Evaluate Systems:** Analyze, evaluate, and optimize computer systems and applications, identifying areas for improvement and implementing solutions to enhance performance, efficiency, and usability
	5.	**Communicate Effectively:** Communicate technical concepts, ideas, and solutions effectively through oral presentations, written reports, and documentation, catering to diverse audiences including technical and non-technical stakeholders.
	6.	**Collaborate in Interdisciplinary Teams:** Collaborate effectively in interdisciplinary teams, demonstrating leadership, teamwork, and interpersonal skills to achieve common goals and objectives in the development and implementation of computing projects.

	7.	**Adapt to Technological Advancements:** Stay abreast of emerging technologies, trends, and developments in the field of computer science, demonstrating a commitment to lifelong learning and professional development.
	8.	**Adhere to Ethical Standards:** Uphold ethical principles and professional standards in all aspects of computer science practice, demonstrating integrity, accountability, and responsibility in decision-making and conduct.
	9.	**Contribute to Society:** Contribute responsibly and ethically to the advancement of technology and society, addressing societal challenges and promoting inclusivity, diversity, and sustainability in computing initiatives.
	10.	**Pursue Continuous Improvement:** Engage in reflective practice and continuous improvement, seeking opportunities for further education, training, and skill enhancement to remain competitive and adaptable in a dynamic technological landscape.
Eligibility Criteria for Programme		A woman graduate from recognized university with aggregate marks of not less than 50% (Open Category) and 45% (Reserved Category).
Intake (For SNDT WU Departments and Conducted Colleges)		35
Duration		1 year (2 Semsters)

Structure with Course Titles
(Options related to our area of study to be provided with "OR" for baskets of different types)

Postgraduate Programme of 1 year:

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester I					
115761	Mathematical Foundation of Computer Science	Major (Core)	4	100	50	50
115762	Fundamental of Computer Science and Operating System	Major (Core)	4	100	50	50
115763	Data Structures and Analysis of Algorithms using JAVA	Major (Core)	4	100	50	50
115764	Data Communication and Networking	Major (Core)	2	50	0	50
	Elective-I	Major (Elective)	4	100	50	50
135781	Research Methodology	Minor Stream (RM)	4	100	50	50
			22	**550**	**250**	**300**
	Semester II					
215761	Data Science using Python Programming	Major (Core)	4	100	50	50
215762	Artificial Intelligence	Major (Core)	4	100	50	50
215763	Full Stack Development	Major (Core)	4	100	50	50
215764	Database Management Systems	Major (Core)	4	100	50	50
	Elective-II	Major (Elective)	4	100	50	50
245781	OJT	OJT	2	50	0	50
			22	**550**	**250**	**300**

Electives			
Code	**Elective - I**	**Code**	**Elective - II**
125761	A. Cyber Security	225761	A. Ethical Hacking
125762	B. Software Testing	225762	B. Introduction to Data Science
125763	C. UI/UX Design	225763	C. Mobile App Development Technologies

RM: Research Methodology
OJT: On-Job Training
RP: Research Project

Course Syllabus
Semester I

1.1 **Major (Core)**

Course Title	**Mathematics for Computer Science**
Course Credits	**4 Credits**
Course Outcomes	**CO1 :** To Study Set and Operations.
	CO2: To study logic and boolean Algebra
	CO3: To enable the students to use nonlinear and linear methods to analyze and understand problems in the same
	CO4: To demonstrate how the mathematics help to develop thinking ability
	CO5: To demonstrate how the graph and logic helps to develop thinking ability
Module 1 (Credit 1) Set Theory	
Learning Outcomes	**After learning the module, learners will be able to**
	To Understand the set theory and operations To facilitate applications of set
Course Outline	Set definition, set builder form. Operations on set ,Venn diagram, DeMorgan's Law
Module 2 (Credit 1) Logic and Boolean Algebra	
Learning Outcomes	**After learning the module, learners will be able to**
	Logic operations
	Boolean operations
Course Outline	Logic gates , truth tables , Boolean expression , Laws
Module 3 (Credit 1) Solution to nonlinear equation	
Learning Outcomes	**After learning the module, learners will be able to**
	To Understand the methods of nonlinear equation
Course Outline	**Solution to nonlinear equation:** Bisection method, Newton Raphson Method, Regula Falsi method
Module 4 (Credit 1) Graph	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand Basics of Graph
Course Outline	Graph Definition basic terminology ,Matrix representation of graphs , Types of graphs , Hand Shaking Lemma

Reference Books:

(1) S.S. Sastry; Introductory Methods of Numerical Analysis, 3rd edition, Prentice Hall of India,
(2) H.C. Saxena; Finite differences and Numerical Analysis, S. Chand and Company.
(3) Kolman and Ross - Discrete Mathematical Structures
(4) B.S. Vatssa:-Discrete Mathematics
(5) Set Theory: A First Course by Daniel W. Cunningham.

Assessment :

Internal Assessment – 50 Marks

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Interpretation	Accuracy in interpreting the example	15
Representation	Translating problems	15
Calculation	To perform correct numerical calculations	15
Logic Development	To enhance logical thinking	5

Sample Projects

Project 1: Find real world applications of graphs

Project 2: Find real world the applications of logic

Project 3: Prepare the model for Set operations

Project 4: Write program for nonlinear equations

External Assessment: (Marks 50)
End Semester examination of 50 marks for 2 hours duration will be conducted.

1.2 Major (Core)

Course Title	**Fundamental of Computer science and Operating System**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Acquire a comprehensive understanding of computer fundamentals, encompassing both hardware and software aspects.
	• Enhance proficiency in programming and software development, bolstering internet literacy and problem-solving skills.
	• Develop skills in various programming languages, understand software development life cycles, and learn effective debugging techniques.
	• Apply knowledge of computer system architecture and operating systems principles in practical scenarios.

	• Analyze the impact of virtualization techniques on system performance and efficiency.
	• Evaluate processor management and memory allocation strategies to optimize system operations.
	• Design solutions using advanced file systems to ensure reliable data management.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply acquired knowledge of essential computer components, including the CPU, input/output devices, memory, and storage in understanding their roles in computer systems.
	• Analyze data using skills in internal data representation, including understanding number systems, binary arithmetic, Boolean algebra, and logic gates.
	• Evaluate computer organization, focusing on the roles of the CPU, memory, and secondary storage to gain a comprehensive understanding.
	• Design computer systems considering various input and output devices, their associated interfaces, and their functions within computer systems.
Content Outline	• **Knowing computer:** What is Computer, Basic Applications of Computer; Components of Computer System, Central Processing Unit (CPU), Input/output Devices, Computer Memory, Concepts of Hardware and Software; Concept of Computing, Data and Information, classification of computers, various generations of computers; What is an Operating System; Different Popular Operating Systems; The User Interface, System Software: System software Vs. Application Software, Types of System Software, Introduction and Types of Operating Systems • **Internal data Representation in Computers and Digital System Design:** Number Systems Used in Computers, Converting from One Number System to Another, Binary Arithmetic, Boolean Algebra, Boolean Functions, Logic Gates, Logic Circuits • **Computer Architecture:** Basic Functions of a Computer, Basic Computer Organization, CPU Architectures, Memory Architectures, Secondary Storage, Classification of Secondary Storage, Memory Storage Devices, • **Input- Output Device**s: Input Devices, Output Devices, I/O Interfaces
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the evolution and various types of computer languages, using them appropriately in different programming scenarios.
	• Analyze programming challenges using skills in programming constructs, algorithms, and flowcharting for effective problem-solving and program development.
	• Evaluate the roles of language processors in software development, understanding their impact on program execution and performance.
	• Design software solutions with a clear understanding of the relationship between software and hardware in the functioning of computer systems.

Content Outline	• **Computer Languages:** Definition, Generations of computer languages, Types of Languages, Language Processors: Assembler, Interpreter, Compiler, Linker and Loader. Programming constructs, Algorithm & flowchart. • **Software:** Basic Concepts and Terminologies, What is Software? ,Relationship between Hardware and Software, Software Development Life Cycle (SDLC), Advantages of SDLC Model, Limitations of SDLC Model, Software Testing and Debugging • **Introduction to Internet:** WWW and Web Browsers: Basic of Computer networks; LAN, WAN; Concept of Internet; Applications of Internet; connecting to internet; What is ISP; Knowing the Internet; Basics of internet connectivity related troubleshooting, World Wide Web; Web Browsing software, Search Engines; Understanding URL; Domain name; IP Address; Using e-governance website
Module 3(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply knowledge of computer system organization and architecture, with a focus on the role of operating systems, in diverse computing environments.
	• Analyze the structure and operations of operating systems, including process management, memory management, storage management, protection, and security.
	• Evaluate various operating system services provided to users and applications, and their effectiveness in different scenarios.
	• Understand and apply the user-operating system interface, system calls, types of system programs, and process scheduling algorithms in practical use cases.
Content Outline	• **Introduction to Operating Systems (OS):** Computer- System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Distributed Systems, Special-Purpose Systems, Computing Environments. • Operating-System Services, User Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Virtual Machines, Operating-System Generation. • **Processor Management:** Process concept, Process scheduling, Operations on Processes, Interprocess Communication, Multithreading models, threading issues, Process scheduling algorithms, Thread scheduling, Multiple processor Scheduling. **Process Coordination:** Synchronization, Semaphores, Monitors, Deadlocks characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock Avoidance, Deadlock detection, recovery from deadlock.
Module 4(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply and understand various memory management techniques such as swapping, contiguous memory allocation, paging, and segmentation in different computing scenarios.

	• Analyze the structure of the page table and its role in memory addressing for efficient resource management.
	• Evaluate the organization and implementation of file systems, including structure, directory implementation, allocation methods, and free-space management.
	• Explore and understand the efficiency and performance considerations in file management, including recovery mechanisms and log-structured file systems.
Content Outline	• **Memory Management:** Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation • **Virtual memory Management:** Demand Paging, Copy- on-Write, Page replacement, Allocation of Frames, Thrashing. • **File Management:** File Concept, File Access Methods, Directory Structure, File Sharing, File Protection, File-System Structure, File- System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File Systems, NFS.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:
Activity 1: Write an essay on the essential components of a computer system, including the CPU, input/output devices, memory, and storage, explaining their roles and functions.
Activity 2: Create a presentation on internal data representation, explaining number systems, binary arithmetic, Boolean algebra, and logic gates. Include examples and illustrations to enhance understanding.
Activity 3: Design a basic computer system on paper, considering various input and output devices and their associated interfaces. Explain the function of each component within the system.

Module 2:
Activity 1: Write a report on the evolution of computer languages. Discuss different types of languages and their appropriate uses in different programming scenarios. Activity 2: Choose a programming problem and develop a solution using programming constructs, algorithms, and flowcharting. Explain your process and how you arrived at your solution.
Activity 3: Create a presentation on the role of language processors in software development. Discuss their impact on program execution and performance.

Module 3:

Activity 1: Write a report on the organization and architecture of computer systems, focusing specifically on the role of operating systems. Discuss the structure and operations of operating systems, including process management, memory management, storage management, protection, and security.

Activity 2: Write a program that simulates the use of system calls. Document the purpose of each system call used and discuss its importance in the operating system.

Activity 3: Simulate a process scheduling algorithm, document the steps, and evaluate its effectiveness in managing processor resources.

Module 4:

Activity 1: Write an essay discussing various memory management techniques such as swapping, contiguous memory allocation, paging, and segmentation. Include examples of scenarios where each technique would be most effective.

Activity 2: Design a page table for a hypothetical system. Discuss the role of the page table in memory addressing and resource management.

Activity 3: Write a report on the organization and implementation of file systems. Discuss the structure, directory implementation, allocation methods, and free-space management, and evaluate their efficiency and performance.

References:

1. Sinha, P. K., & Sinha, P. (2004). Computer Fundamentals.BPB Publications.
2. Srivastava, C. (2000). Fundamentals of Information Technology. Kalyani Publishers.
3. Rajaraman, V. (2003). Fundamentals of Computers (4th ed.). PHI Publication.
4. Jain, R. K. (2014). Fundamentals of Programming. S. Chand Publication.
5. Silberscatz, A., Galvin, P. B., & Gagne, G. (2005). Operating System Concepts (7th ed.). John Wiley and Sons, Inc.
6. Milenkovic, M. (1992). Operating Systems Concepts And Design (2nd ed.). McGraw-Hill International Editions.
7. Stallings, W. (2005). Operating Systems: Internals and Design Principles (5th ed.). Prentice Hall.
8. Tanenbaum, A. (2007). Modern Operating Systems (3rd ed.). Pearson Education.

1.3 Major (Core)

Course Title	**Data Structures using Java**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Apply understanding of algorithm analysis, including asymptotic performance, to assess and optimize algorithm efficiency.
	• Analyze algorithms by writing rigorous correctness proofs, ensuring their accuracy and validity.
	• Demonstrate familiarity with data structures, understanding their application in various scenarios.
	• Demonstrate and Implement JAVA concept for data structures
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the fundamental concepts of data structures, including arrays, stacks, queues, and linked lists.
	• Analyze scenarios where each data structure is best suited for efficient data organization and manipulation, choosing the appropriate data structure based on the situation.
	• Design efficient solutions to data-related problems by applying asymptotic notations to evaluate and compare the efficiency of algorithms related to data structures.

Content Outline	• **Introduction to Data Structures** – Fundamental Elements – Asymptotic Notations: Big-Oh, Omega and Theta – Best, Worst and Average case Analysis: • **Definition and an example** Arrays and its representations • **Stacks and Queues** – Linked lists – Singly Linked List – Doubly linked list – Linked list based implementation of Stacks and Queues. Evaluation of Expressions
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the basic concepts of trees and their significance in data representation and manipulation.
	• Analyze differences between trees and other data structures, emphasizing the hierarchical nature of tree structures.
	• Evaluate the use of binary trees and their applications in various domains.
	• Design binary trees using both array and linked list structures, demonstrating proficiency in these representations.
Content Outline	• **Trees:** Introduction to Trees – Basic concepts – Binary Trees – Binary tree representations (Array and list) and Traversals Techniques (Preorder, Inorder, Postorder) • **Succinct Data Structures**: Overview – Level order representation of Binary Trees – Rank and Select – Sub trees.AVL,B-Trees. Binary Search Trees
Module 3(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the basic concepts of graphs and their significance in data representation and manipulation. • Evaluate the use of graphs and their applications in various domains
Content Outline	• **Graphs:** Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search
Module 4(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the basic concepts of sorting and searching and their applications • Evaluate the use various sorting and searching and indexing techniques • Understand concepts of JAVA for data structures programming
	• **Searching and sorting:** Sorting Algorithms: Basic concepts – Bubble Sort – Insertion Sort – Selection Sort – Quick Sort – Shell sort – Heap Sort – Merge Sort – External Sorting. Searching: Linear Search, Binary Search. • **Indexing:** Hashing – Hash Functions – Separate Chaining – Open Addressing: Linear Probing- Quadratic Probing- Double Hashing- Rehashing – Extendible Hashing. • **JAVA programming Concept for Data Structure**

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

1. Test student's understanding of fundamental concepts of data structure and algorithm.
2. Research and present examples illustrating Big-Oh, Omega, and Theta notations.
3. Analyze and compare the time complexities of different algorithms using these notations.

Module 2:

1. Create and visualize binary trees using array and linked list representations.
2. Implement and demonstrate tree traversal techniques (Preorder, Inorder, Postorder).
3. Study and implement level order representation of binary trees.
4. Investigate and implement Rank and Select operations on succinct data structures.
5. Implement Binary Search Trees and AVL Trees.
6. Analyze and compare the performance of these tree structures.
7. Implement hash functions and explore different hashing techniques.

Module 3:

1. Test the understanding on Graph datastructures.
2. Explore and implement Breadth First Search and Depth First Search algorithms for graph traversal.

Module 4:

1. Test the understanding on searching and sorting techniques.
2. Explore with example the concepts of searching and sorting.
3. Implement various searching sorting algorithms.
4. Implement hashing technique.

Practicals

Module 1:

Activity 1: Write a program in JAVA to implement a singly linked list. The program should be able to add elements, delete elements, and display the linked list.

Activity 2: Write a program in JAVA to implement a doubly linked list. Like the previous program, it should be able to add elements, delete elements, and display the doubly linked list. Compare the two programs and discuss the advantages and disadvantages of using singly linked lists versus doubly linked lists.

Activity 3: Write a program in JAVA to implement a Queue. Like the previous program, it should be able to add elements, delete elements.

Activity 4: Write a program in JAVA to implement a Stack. Like the previous program, it should be able to push elements, pop elements.

Module 2:

Activity 1: Write a program in JAVA to implement a binary tree. The program should be able to add elements and perform tree traversal techniques including Preorder, Inorder, and Postorder.

Activity 2: Write a program in JAVA to implement a binary search tree.

Module 3:

Activity 1: Implement Breadth First Search and Depth First Search algorithms for graph traversal. Analyze and compare the performance of these traversal techniques.

Module 4:

Activity 1: Write a program in JAVA to implement hash functions. The program should be able to map data of arbitrary size to fixed-size values. Discuss different hashing techniques and their uses.

References

1. Pai, G. A. V. (2008). Data Structures and Algorithms: Concepts, Techniques, and Applications (1st ed.).
2. Horowitz, E., Sahni, S., & Anderson-Freed, S. (2007). Fundamentals of Data Structures in C (2nd ed.). University Press.
3. Tremblay, J.-P., & Sorenson, P. G. (2007). An Introduction to Data Structures with Applications (2nd ed.). Tata McGraw-Hill.
4. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2003). Introduction to Algorithms (2nd ed.). PHI.
5. Dave, P., & Dave, H. (2008). Design and Analysis of Algorithms. Pearson Education.
6. Knuth, D. E. (1973). Sorting and Searching: The Art of Computer Programming, Vol. 3. Addison-Wesley Publishing.
7. Goodrich MT, Tamassia R, Goldwasser MH. Data structures and algorithms in Java. John wiley& sons; 2014 Jan 28.
8. Sedgewick R, Wayne K, Algorithms (4th edition) Addison-Wesley Professional
9. Pai, G. A. V. (2008). Data Structures and Algorithms: Concepts, Techniques, and Applications (1st ed.)
10. Bailey D. Java structures. McGraw-Hill Companies; 1999 Jan.
11. Downey A. Think data structures: algorithms and information retrieval in Java. " O'Reilly Media, Inc."; 2017 Jul 7.
12. Karumanchi N. Data structures and algorithms made easy in java. CareerMonk Publications; 2011.

1.4 Major (Core)

Course Title	**Data Communications and Networking**
Course Credits	**2**

Course Outcomes	**After going through the course, learners will be able to**
	• Apply understanding of key concepts in data communications, including digital and analog signals, bandwidth, data transmission modes, and network types.
	• Analyze the OSI (Open Systems Interconnection) model and TCP/IP protocol suite, understanding their structure, functions, and importance in data communication and networking.
	• Evaluate different networking protocols such as TCP, UDP, IP, and HTTP, understanding their functions, characteristics, and application scenarios.
	• Design and implement basic computer networks based on specific requirements, including choosing the appropriate networking equipment and setting up connections.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the concept of computer networks and their significance in modern computing, including data sharing, resource sharing, and communication efficiency.
	• Analyze the purposes and advantages of connecting computers in a networked environment such as improved collaboration, resource sharing, and cost-effectiveness.
	• Differentiate between common network topologies like bus, star, ring, mesh, and tree, understanding their structure and working principles.
	• Evaluate the strengths and weaknesses of various topologies in different scenarios, considering factors like scalability, fault tolerance, installation and maintenance costs.
Content Outline	• **Introduction:** Computer Networks and its uses, Network categorization and Hardware: Broadcast and point-to-point networks, Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Networks (WAN), Inter networks, Topologies, Wireless Networks, Network Software: Protocols, Services, network architecture, design issues, OSI Reference model, TCP/IP Reference model, Comparison of OSI and TCP/IP Models. Introduction to Example Networks: Internet, Connection Oriented Networks–X.25, Frame Relay, ATM • **Data Communication Model**: Digital and Analog data and signals, bit rate, baud, bandwidth, Nyquist bit rate, Guided Transmission Media – Twisted Pair, Coaxial cable, Optical fiber; wireless transmission–Radio waves, microwaves, infrared waves; Satellite Communication.
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Differentiate circuit switching from other switching techniques like packet switching and message switching. Circuit switching involves establishing a dedicated communication path between devices for the duration of their conversation.
	• Describe the process of establishing a circuit in a circuit-switched network, which includes call setup, data transfer, and call teardown phases.
	• Analyze the advantages and disadvantages of circuit switching. Advantages include predictable performance and minimal delay, while disadvantages include inefficiency in resource utilization and potential for wasted capacity.
	• Understand common circuit switching protocols such as Integrated

	Services Digital Network (ISDN), which is a set of communication standards for simultaneous digital transmission of voice, video, data, and other network services.
Content Outline	• **Switching:** Circuit Switching, Packet switching; Multiplexing: Frequency Division Multiplexing, Time Division Multiplexing, Synchronous and Asynchronous TDM, Modems, Transmission impairments, Manchester and differential Manchester encoding • **Data Link Layer Design issues:** Framing, error control, Flow Control, Error Detection and correction; Elementary Data Link Protocols, Sliding Windows Protocols; Medium Access Control: Aloha, CSMA protocols, Collision free protocols, Limited Contention Protocols; Wave length division Multiple access protocol, Wireless • **Network Layer**, Design issues Virtual circuit and Datagram Subnet, Routing Algorithms, Optimality principle, shortest path routing, Flooding, Distance Vector Routing, Link State Routing

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Test students' understanding of fundamental concepts in data communications and networking.

Module 1:

Activity 1: Write a report discussing the concept of computer networks and their significance in modern computing. Discuss the advantages of connecting computers in a networked environment and differentiate between common network topologies like bus, star, ring, mesh, and tree.

Activity 2: Choose a specific network topology and evaluate its strengths and weaknesses in a given scenario. Consider factors like scalability, fault tolerance, installation, and maintenance costs.

Activity 3: Write a report differentiating between circuit switching and other switching techniques like packet switching and message switching. Discuss the process of establishing a circuit in a circuit-switched network and analyze the advantages and disadvantages of circuit switching.

Activity 4: Design a simple circuit-switched network using a common circuit switching protocol. Document and reflect on your process.

Module 2:

Activity 1: Write a report on the role and functions of the Data Link Layer in the OSI model. Analyze various framing techniques used in the Data Link Layer and evaluate their efficiency and effectiveness.

Activity 2: Implement a simple framing algorithm for efficient data encapsulation and transmission across networks. Document and reflect on your process and any challenges you encountered.

Activity 3: Write a report discussing the responsibilities of the Network Layer in the OSI model and differentiating it from other layers. Discuss efficient addressing schemes for packet routing.

Activity 4: Design a network configuration using an appropriate switching technique to meet a specific networking need. Reflect on the performance, advantages, and disadvantages of your chosen technique

References: -

1. Forouzan, B. A. (2007). Data Communications and Networking (4th ed.). McGraw Hill.
2. Stallings, W. (2013). Data and Computer Communications (10th ed.). Pearson.
3. Tanenbaum, A. S. (2010). Computer Networks (5th ed.). Pearson.

1.5 Major (Elective-I) -A

Course Title	**Cyber Security**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Apply understanding of various cyber threats, including Cyber Warfare, Cyber Crime, Cyber Terrorism, and Cyber Espionage, differentiating their unique characteristics and potential impacts.
	• Analyze the importance of a comprehensive Cyber Security policy, the role of a nodal authority, and the significance of an international convention on Cyberspace for global cybersecurity standards.
	• Evaluate vulnerabilities in software, system administration, network architectures, data access, authentication, broadband communications, and human factors, devising strategies to mitigate these risks.
	• Design basic security measures for HTTP and SOAP services, understanding identity management and authorization patterns, and addressing challenges in securing web applications.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply knowledge of Cyber Security to identify and understand vulnerabilities in software, system administration, network architectures, and other areas, devising strategies to mitigate these risks.
	• Analyze systems and networks by conducting security audits, identifying potential weaknesses, and recommending improvements.
	• Evaluate the role of cryptography in Cyber Security, applying cryptographic techniques to secure data communication and ensure data integrity and confidentiality.
	• Design proactive security measures by understanding the concept of ethical hacking, its role in identifying and addressing system and network vulnerabilities.
Content Outline	• **Introduction to Cyber Security** Overview of Cyber Security, Internet Governance–Challenges and Constraints, Cyber Threats:- Cyber Warfare-Cyber Crime-Cyber, terrorism-Cyber Espionage, Need for a Comprehensive Cyber Security Policy, Need for a Nodal Authority, Need for an International convention on Cyberspace • **Cyber Security Vulnerabilities and Cyber Security Safeguards** Cyber Security Vulnerabilities Overview, vulnerabilities in software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication, Unprotected Broadband communications, Poor Cyber Security Awareness. Cyber Security Safeguards-Overview, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Security

	policy, Threat Management.
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply fundamental security measures for HTTP applications, protecting against common vulnerabilities and ensuring data integrity and safety.
	• Analyze identity management principles and implement secure identity practices within web services, focusing on user verification and access control.
	• Evaluate the effectiveness of applied security measures in preventing and detecting unauthorized access attempts by external entities.
	• Design robust security protocols to safeguard network and system resources, enhancing overall cybersecurity posture.
Content Outline	• **Securing Web Application** Services and Servers Introduction, Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns, Security Considerations, Challenges. • **Intrusion Detection and Prevention** Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques, Anti-Malware software, Network based Intrusion detection Systems, Network based Intrusion Prevention Systems, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis, System Integrity Validation
Module 3(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply knowledge of both symmetric and asymmetric key cryptography, understanding their unique applications in securing data transmission and storage.
	• Analyze data security requirements in various contexts and utilize appropriate cryptographic methods to secure the data.
	• Evaluate the effectiveness of cryptographic methods in different applications, ensuring data security in various contexts.
	• Design strategies for data security by understanding the specificities of the Indian cyber space, including its regulatory framework, challenges, and initiatives.
Content Outline	• **Cryptography and Network Security:** Introduction to Cryptography, Symmetric key Cryptography, Asymmetric key Cryptography, Message Authentication, Digital Signatures, Applications of Cryptography. Overview of Firewalls- Types of Firewalls, User Management, VPN Security, Security Protocols:-security at the Application Layer-PGP and S/MIME, Security at Transport Layer-SSL And TLS, Security at Network Layer-IPSec. • **Cyber space and the Law**: Introduction, Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyber space, Cyber Security Standards. The INDIAN Cyber space, National Cyber Security Policy 2013.
Module 4(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply skills in initiating and conducting preliminary investigations in response to suspected cyber incidents, ensuring the preservation of digital evidence.

	• Analyze digital evidence stored on computer hard drives and other storage media by developing proficiency in conducting disk-based analysis.
	• Evaluate the results from the preliminary investigations and disk-based analysis, ensuring accuracy and completeness for potential forensic analysis.
	• Design effective strategies for preserving and analyzing digital evidence, crucial for investigating and responding to cybersecurity incidents.
Content Outline	• **Cyber Forensics:** Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation, Conducting disk-based analysis, Investigating Information-hiding, Scrutinizing E-mail, Validating E mail header information, Tracing Internet access, Tracing memory in real-time.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

Activity 1: Write a report on a recent cyber security incident in which you identify vulnerabilities that were exploited and devise potential strategies to mitigate these risks.

Activity 2: Conduct a security audit of a hypothetical system or network, identifying potential weaknesses and recommending improvements. Discuss the role of cryptography in securing data communication and data integrity within this system or network.

Module 2:

Activity 1: Design a set of fundamental security measures for a hypothetical HTTP application. Discuss how these measures protect against common vulnerabilities and ensure data integrity and safety.

Activity 2: Write a report on identity management principles and their application within web services. Include a discussion on user verification and access control, and how they can prevent unauthorized access attempts.

Module 3:

Activity 1: Write a short essay on symmetric and asymmetric key cryptography. Discuss their unique applications in securing data transmission and storage.

Activity 2: Choose a data security requirement for a hypothetical scenario and discuss which cryptographic method would be most appropriate to secure the data. Evaluate the effectiveness of this method in the given context.

Module 4:

Activity 1: Write a mock incident report for a suspected cyber incident. Discuss how you would initiate and conduct a preliminary investigation to preserve digital evidence.

Activity 2: Analyze a hypothetical digital evidence stored on a computer hard drive. Discuss your findings and their potential implications for a forensic analysis.

References:

1. Hassan, N., & Hijazi, R. (n.d.). Digital Privacy and Security Using Windows: A Practical Guide. Apress.
2. DSCI-Nasscom. (2013). Cyber Crime Investigation.
3. Gobole, N. (2008). Information Systems Security: Security Management, Metrics, Frameworks and Best Practices (With CD).
4. Weber, R. (1999). Information Systems Control and Audit. Pearson Pub.
5. Pettier, T. (2004). Information Security Policies, Procedures and Standards.
6. Tipton, H. F. (Ed.). (2003). Information Security Management Handbook (5th Edition).
7. Basta, A., & Halton, W. (2008). Computer Security.Cengage Learning
8. Peltier, T. R. (2016). Information Security Policies. Auerbach Publications

1.5 Major (Elective-I) -B

Course Title	**Software Testing**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Analyse fundamental principles and concepts of software testing, its purpose, objectives, and role in software development.
	• Proficient in test design techniques, including equivalence partitioning, boundary value analysis, decision tables, and state transition testing, to create effective test cases.
	• Apply testing methods such as functional, non-functional, unit, integration, system, regression, and acceptance testing to diverse software systems
	• Evaluate quality assurance principles and best practices, emphasizing testing's importance in the software development lifecycle, along with ethical and professional responsibilities associated with software testing.
Module 1(Credit 1) Overview of Software Testing	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply software testing terminologies, methodologies, and life cycles proficiently.
	• Analyze the economic aspects of testing and its influence on organizational structures.
	• Develop skills in policy creation, test strategies, and risk management to meet customer needs effectively.
	• Evaluate the advantages of structured testing processes and their associated cost implications, demonstrating proficiency in the seven-step software testing process..
Content Outline	• Software Testing Terminology and Methodology Software Testing Terminology, Software Testing Life Cycle, Writing a Policy for Software Testing, Economics of Testing, Testing – An organizational Issue, Management Support for Software Testing, Fig. of Software Testing Methodology, Risk associated with not meeting customer needs, Developing Test Strategy • Overview of Software Testing Process Advantages of Following a Process, The Cost of Computer Testing, The Seven-Step Software Testing Process • Verification and Validation Verification and Validation (V&V)

	Activities, Verification, Verification of Requirements, Verification of High –level Design, Verification of Low –level Design, How to Verify Code? ,Validation • Static Testing Inspections,λ Structured Walkthroughs, Technical Reviews.
Module 2(Credit 1) Validation and Regression Testing	
Learning Outcomes	**After learning the module, learners will be able to**
	• Execute validation activities, including unit, integration, function, system, and acceptance testing, to assess software functionality and ensure compliance with requirements.
	• Differentiate between progressive and regressive testing, understanding the significance of regression testing in maintaining software quality.
	• Apply regression testing techniques to identify potential issues from software changes, ensuring stability and reliability
	• Identify regression testing objectives, determine suitable instances for conducting tests, define regression test problems, and select appropriate types and strategies to minimize risks.
Content Outline	• Validation Activities Unit Validation Testing, Integration Testing, Function Testing, System Testing, Acceptance Testing • Regression Testing Progressive vs. Regressive Testing, Regression Testing Produces Quality Software, Regression Testability, Objectives of Regression Testing, When is Regression Testing Done? , Regression Testing Types, Defining Regression Test Problem, Regression Testing Techniques.
Module 3(Credit 1) Testing Management and Metrics	
Learning Outcomes	**After learning the module, learners will be able to**
	• Establish a comprehensive understanding of test management structures, facilitating effective test planning and detailed design
	• Recognize the necessity of software metrics and demonstrate the capability to define, classify, and apply them within the software development life cycle.
	• Evaluate entities to be measured in software projects, with a focus on size metrics and their implications for software management.
	• Formulate testing-specific measurement objectives, identifying attributes and relevant metrics for monitoring and controlling the testing process..
Content Outline	• Test Management Test Organization, Structure of Testing Group, Test Planning, Detailed Test Design and Test Specifications • Software Metrics Need for Software Management, Definition of Software Metrics, Classification of Software Metrics, Entities to be Measured, Size Metrics • Testing Metrics for Monitoring and Controlling the Testing Process Measurement Objectives for Testing, Attributes and Corresponding Metrics in Software Testing, Attributes, Estimation Models for Estimating Testing Efforts (include only topic Halstead Metrics), Test Point Analysis (TPA) – introduction only.
Module 4(Credit 1) Automation Testing Tool	
Learning Outcomes	**After learning the module, learners will be able to**
	• Evaluate the necessity and significance of test process maturity, measuring, assessing, and improving processes using established models

	• Identify the rationale for automation in testing, categorize testing tools, and apply selection criteria considering associated costs.
	• Analyze guidelines for automated testing and gain an overview of commercial tools to implement automation effectively.
	• Apply agile methodologies to enhance testing, recognizing agility's importance, overcoming inhibitors, and implementing solutions within an agile framework.
Content Outline	• Testing Process Maturity Models Need for Test Process Maturity, Measurement and Improvement of a Test Process, Test Process Maturity Models • Automation and Testing Tools Need for Automation, Categorization of Testing Tools, Selection of Testing Tools, Cost Incurred in Testing Tools, Guidelines for Automated Testing, Overview of Some Commercial Testing Tools Testing Object Oriented SSoftware ObjectOriented Testing • Using Agile Methods to Improve Software Testing The importance of Agility, Building an Agile Testing Process, Agility Inhibitors, Is Improvement Necessary, Compressing Time, Challenges, Solutions , Measuring Readiness , The Seven-Step Process 4.5 Test Plan.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1: To develop a comprehensive software testing policy to ensure efficient testing practices within an organization.
Module 2: Begin by researching and understanding the software testing terminology, methodologies, and the software testing life cycle. Evaluate the economics of testing and its organizational impact,including management support for software testing. Develop an understanding of the seven-step software testing process and its advantages.
Module 3: Identify the verification and validation activities, including verification of requirements, high-level design, low-level design, and code validation. Explore static testing techniques such as inspections, structured walkthroughs, and technical reviews.
Module 4: Based on your understanding of the above concepts, draft a policy document outlining the software testing process. Your policy document should include sections on software testing terminology, the software testing life cycle, test strategy development, verification and validation activities

References:
1. Chauhan, N. (2016). Software testing principles and practices. Oxford University Press.
2. Perry, W. E. (2006). Effective methods of software testing (3rd ed.). Wiley, India.
3. Desikan, S., & Ramesh, G. (2005). Software testing principles and practices. Pearson Education.
4. Patton, R. (2005). Software testing (2nd ed.). Pearson Education.
5. Dustin, E. (2003). Effective software testing: 50 specific ways to improve your testing. Pearson Education.

1.5 Major (Elective-I) -C

Course Title	**UI/UX Design**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**

	• Understand and apply core principles of User Interface (UI) and User Experience (UX) design. • Conduct user research and translate insights into design decisions. • Create wireframes, prototypes, and visual designs using industry tools. • Evaluate and iterate on design solutions through usability testing. • Apply accessibility, responsive design, and design system principles to real-world projects.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Understand the difference between UI and UX. • Apply principles of human-centered and interaction design. • Conduct user research, build personas, and define user journeys.
Content Outline	• Introduction to UI/UX: Definitions, Differences, Importance. • Design Thinking & Human-Centered Design Principles. • UX Process: Empathize, Define, Ideate, Prototype, Test. • User Research: Interviews, Surveys, Observation. • Personas, Empathy Maps, User Journey Maps.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply visual hierarchy, layout principles, color theory, and typography. • Design intuitive interfaces using layout and component best practices. • Utilize UI patterns and design consistency.
Content Outline	• UI Design Principles: Alignment, Consistency, Contrast, Proximity. • Color Theory and Accessibility (WCAG standards). • Typography and Visual Hierarchy. • Layout and Grids, Spacing and Balance. • UI Components and Design Systems. • Mobile-First and Responsive Design Concepts.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Build interactive prototypes using design tools. • Test designs through usability testing and interpret feedback. • Understand Figma/Adobe XD/Sketch for prototyping.
Content Outline	• Introduction to UI Design Tools: Figma, Adobe XD, Sketch. • Wireframing vs Mockups vs Prototypes. • Creating Interactive Prototypes. • Usability Testing: Goals, Setup, Metrics. • Iterative Design and Feedback Integration.
Module 4 (Credit 1)	

Learning Outcomes	**After learning the module, learners will be able to**
	• Apply principles of accessibility, inclusivity, and responsive design. • Integrate UI/UX deliverables into development workflows. • Prepare a professional UI/UX design portfolio.
Content Outline	• Apply principles of accessibility, inclusivity, and responsive design. • Integrate UI/UX deliverables into development workflows. • Prepare a professional UI/UX design portfolio.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)
Module 1:

- Conduct a user interview and summarize findings.
- Develop a persona and journey map for a product.
- Present a group discussion on good vs poor UX design.

Module 2:

- Create a UI mood board and color palette for an app.
- Design wireframes and mockups for a mobile or web interface.
- Conduct a heuristic evaluation of an existing UI.

Module 3:

- Create a clickable prototype of a selected app.
- Conduct usability testing with at least 3 users and analyze results.
- Present design iterations and refinements.

Module 4:

- Audit an interface for accessibility compliance.
- Build a responsive UI prototype for both mobile and web.
- Submit a personal UI/UX case study as part of a portfolio.

References

1. **Don Norman** – *The Design of Everyday Things*, Basic Books.
2. **Steve Krug** – *Don't Make Me Think*, New Riders.
3. **Alan Cooper** – *About Face: The Essentials of Interaction Design*, Wiley.
4. **Interaction Design Foundation** – www.interaction-design.org (articles and readings).
5. **Google Material Design Guidelines** – material.io
6. **Lidwell, Holden, and Butler** – *Universal Principles of Design*, Rockport.
7. **Jakob Nielsen & Rolf Molich** – Usability Heuristics for User Interface Design (nngroup.com).
8. **Figma Learn** – learn.figma.com (for hands-on tutorials and best practices).
9. **Adobe XD Ideas** – ideas.adobe.com (industry articles and case studies).

1.6 Minor Stream (RM)

Course Title	**Research Methodology**
Course Credits	**4 Credits**
Course Outcomes	**After going through the course, learners will be able to**
	• Apply understanding of the key steps involved in the research process, including problem formulation, literature review, data collection, analysis, and interpretation.
	• Analyze research challenges by formulating clear and focused research questions and hypotheses based on a thorough review of existing literature and identification of research gaps.
	• Evaluate and select suitable research designs based on the nature of the research questions, including experimental, quasi-experimental, and non-experimental designs.
	• Design comprehensive literature reviews to identify relevant studies, theories, and methodologies within a specific research domain.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the fundamental meaning of research and its role in acquiring knowledge and solving problems, identifying the primary objectives of research, including the pursuit of knowledge, problem solving, and contributing to existing knowledge.
	• Analyze research challenges by developing the skills necessary to define a clear and focused research problem, including the identification of gaps in existing literature and formulating relevant research questions.
	• Evaluate hypotheses as they serve as a tentative explanation or prediction that guides the research, understanding their role in the research process.
	• Design research strategies based on a clear understanding of research fundamentals, problem identification, and hypothesis formulation.
Content Outline	• **Foundations of Research:** Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable Research Process. • **Problem Identification & Formulation** – Research Question – Investigation Question. • **Measurement Issues** – Hypothesis – Qualities of a good Hypothesis – Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of research design and its crucial role in shaping the structure and execution of research studies.
	• Analyze research challenges using the concept of exploratory research, its types, and its applications in uncovering new insights and generating hypotheses.
	• Evaluate the use of qualitative research, with its emphasis on exploring the depth and complexity of phenomena. Understand the concepts of measurement, causality, generalization, and replication in the context of qualitative research.

	• Design research strategies considering the concept of scaling in research, recognizing its importance in measuring attitudes, opinions, and other abstract concepts.
Content Outline	• **Research Design:** Concept and Importance in Research – Features of a good research design – Exploratory Research Design – concept, types and uses, Descriptive Research Designs – concept, types and uses. Experimental Design: Concept of Independent & Dependent variables. • **Qualitative and Quantitative Research:** Qualitative research – Quantitative research – Concept of measurement, causality, generalization, replication. Merging the two approaches. • **Measurement and Scale:** Concept of measurement– what is measured? Problems in measurement in research – Validity and Reliability. Measurement Scale – Nominal, Ordinal, Interval, Ratio.
Module 3(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the sampling frame and its role as a list or framework from which the actual sample is drawn.
	• Analyze the factors influencing sample size determination and its significance in achieving reliable results.
	• Evaluate the key characteristics that contribute to the representativeness, reliability, and validity of a good sample in research.
	• Design research strategies considering the process of selecting a simple random sample, understanding its application in ensuring each element has an equal chance of being included.
Content Outline	• **Sampling:** Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non Response. Characteristics of a good sample. • **Probability Sample** – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample, Practical considerations in sampling and sample size.
Module 4(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the importance of data preparation in ensuring the quality and reliability of data for analysis.
	• Analyze the essential components and structure of a research paper, including title, abstract, introduction, methodology, results, discussion, and conclusion.
	• Evaluate the effective use of reference materials such as encyclopedias, research guides, handbooks, and other resources to enhance research quality.
	• Design data science findings presentations using effective communication skills, leveraging knowledge gained through case study analysis.
Content Outline	• **Analysis:** Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis –Cross tabulations and Chi-square test including testing hypothesis of association. • **Interpretation of Data and Paper Writing** – Layout of a Research Paper, Journals in Computer Science, Impact factor of Journals, When and where to publish ? Ethical issues related to publishing, Plagiarism and Self Plagiarism. • **Use of Encyclopedias**, Research Guides, Handbook etc., Academic

	Databases for Computer Science • **Business Communication** for Data Science (Case Study- Presentation)

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

Activity 1: Write a short essay on the fundamental meaning of research, its role in acquiring knowledge and solving problems. Discuss the primary objectives of research.

Activity 2: Identify a research problem in your area of interest. Define the problem clearly, identify gaps in existing literature, and formulate relevant research questions and hypotheses.

Module 2:

Activity 1: Design a simple research study. Choose whether to use an exploratory, descriptive, or experimental design and explain your choice. Discuss how you would approach qualitative or quantitative research in your study.

Activity 2: Create a measurement scale for a variable in your research study. Discuss the type of scale you chose (nominal, ordinal, interval, ratio) and why.

Module 3:

Activity 1: Explain the concept of a sampling frame and its importance in research. Discuss the factors that influence sample size determination.

Activity 2: Select a sampling method for your research study and explain why you chose it. Discuss how you would ensure your sample is representative.

Module 4:

Activity 1: Discuss the importance of data preparation in research. Describe the steps you would take to ensure your data is reliable for analysis.

Activity 2: Write a mock research paper based on your study design from the previous modules. Be sure to include all essential components: title, abstract, introduction, methodology, results, discussion, and conclusion.

Activity 3: Create a presentation of your research findings. Practice presenting your findings in a clear and concise manner.

References

1. Kothari, C. R. (2004). Research Methodology. Wiley Eastern.
2. Wilkinson, K. P., & Bhandarkar, L. (2010). Formulation of Hypothesis. Himalaya Publication.
3. Best, J. W., & Kahn, V. (2016). Research in Education. PHI Publication.
4. Kumar, R. (2023). Research Methodology: A Step by Step Guide for Beginners. Pearson.

5. Krishna Swami, K. N., & Others. (2006). Management Research Methodology: Integration of Principles, Methods, and Techniques. Pearson Education.

Semester II

2.1. Major Core

Course Title	**Data Science using Python Programming**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Recognizing different types and sources of data, along with the ability to collect and analyze it. • Understanding various statistical concepts like distributions, hypothesis testing, confidence intervals, and correlation. • Familiarity with various machine learning algorithms such as Linear Regression, Logistic Regression, Decision Trees, Clustering, etc. • Application of text mining techniques to analyze unstructured data, possibly including hands-on lab sessions for practical understanding
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply decision-making constructs, loops, and control statements in Python. • Analyse and implement OOP concepts, including inheritance, polymorphism, and encapsulation, along with anonymous functions. • Evaluate and implement exception handling using try, except, and finally blocks. • Design Python programs utilizing key features of the language.
Content Outline	• **Introduction to Python Programming**–Why Python? – Essential Python libraries – Python Introduction- Features, Identifiers, Reserved words, Indentation, Comments, Built-in • **Data types and their Methods:** Strings, List, Tuples, Dictionary, Set – Type Conversion- Operators • **Decision Making- Looping**- Loop Control statement- Math and Random number functions • **User defined functions** – function arguments & its types User defined Modules and Packages in Python- • **Files**: File manipulations, File and Directory related methods Python Exception Handling. • **OOPs Concepts** – Class and Objects, Constructors – Data hiding- Data Abstraction- Inheritance, Types of Inheritance, Polymorphism, Encapsulation, Anonymous Function.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply arithmetic operations, indexing, slicing, and Boolean indexing on NumPy arrays. • Analyse and manipulate NumPy ndarrays, including specifying data types. • Evaluate and implement essential functionality such as dropping entries, indexing, selection, and filtering in pandas. • Design data manipulation workflows using unique values, value counts,

	and membership in pandas DataFrames.
Content Outline	• **NumPy Basics:** Arrays and Vectorized Computation- The NumPy ndarray- Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy • **Arrays**- Basic Indexing and Slicing – Boolean Indexing Transposing Arrays Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical Methods-Sorting Unique and Other Set Logic. • **Introduction to pandas Data Structures**: Series, Data Frame, Essential Functionality: Dropping Entries Indexing, Selection, and Filtering- Function Application and Mapping- Sorting and Ranking. • **Summarizing and Computing Descriptive Statistics**- Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Identify and differentiate between various types of data • Illustrate the stages within the Data Science lifecycle and their significance in problem-solving. • Construct models for predictive or descriptive purposes based on analyzed data. • Implement feature extraction methodologies to enhance model performance and interpretability. • Perform various mathematical operations efficiently on NumPy arrays. • Comprehend the data structures provided by Pandas (Series, DataFrame) and their applications in data analysis. • Execute merging operations to combine datasets efficiently for comprehensive analysis. • Apply various types of joins effectively to merge datasets based on common columns or indices.
Content Outline	• What is Data? Different kinds of data, Data Sources, Different types of data sources, • Exploratory Data Analysis (EDA), Data Science lifecycle, Data Collection • Data Extraction, Data Analysis & Modeling, • Data transformations: Dimension reduction, Feature extraction, Smoothing and aggregating. • The World of arrays with Numpy: creating an array, Mathematical operations, Indexing and slicing, Shape manipulation. • Empowering Data analysis with pandas: the data structure of pandas, Inserting and exporting data, • Data Cleansing: checking missing data, filling missing data, merging operations • Data Operations: Aggregation operations, Joins
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**

	• Identify and differentiate between various probability distributions (normal, binomial, Poisson, etc.) and their realworld applications. • Control line properties in various chart types for better visualization. • Differentiate between various types of Machine Learning (supervised, unsupervised, reinforcement learning). • Implement User-Based Collaborative Filtering techniques for generating recommendations based on user similarities. • Explain the theory behind text mining and its applications in analyzing unstructured data.
Content Outline	• Inferential Statistics: Various forms of distribution, z-score, pvalue, Type 1 and Type 2 errors, Confidence Interval, Correlation, Chi-square distribution, ANOVA • Making Sense of Data Through Visualization: Controlling the line properties of a chart, creating multiple plots, styling your plots, Boxplots, Heatmaps, Scatter plots with histogram, Bubble charts • Uncovering Machine Learning: Different types of Machine Learning, Linear Regression, Logistic Regression, Decision Tree, K-means Clustering, Hierarchical Clustering • Generating Recommendations Systems: User Based collaborative filtering, Item Based collaborative filtering, Context Based filtering • Case Study Theory: Analyzing Unstructured Data using Text mining techniques. (Case Study Practical Implementation to be performed in lab as part of Practical's)

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

These assignments aim to encouraging practical application and critical thinking.

• Students research and present on various types of data sources, their advantages, and limitations.

• Given a dataset, apply dimension reduction, feature extraction, and smoothing/aggregation methods. Explain the impact of each transformation on the dataset.

• Solve coding exercises using NumPy and Pandas to manipulate arrays, perform data operations, and handle missing data.

• Analyze a dataset statistically, calculate z-scores, p-values, and confidence intervals. Interpret the findings and make conclusions.

• Describe different ML algorithms (Linear Regression, Logistic Regression, Decision Trees, Clustering) and their applications with examples.

Module 1:

1. Web Scraper: Write a Python program using built-in data types, functions, loops, and decision-making constructs that scrapes data from a website (such as a news site or weather forecast site) and stores the data in a text file.
2. Contact Book Application: Design a Python program that implements a simple contact book. Use OOP concepts to create classes for 'Contact' and 'AddressBook'. The 'Contact' class should have data members for storing name, phone number, email, etc., and 'AddressBook' class should manage multiple 'Contact' objects.
3. File Organizer: Write a Python program that organizes files in a specified directory on your computer. The program should categorize files based on file type (e.g., .txt, .docx, .pdf) and move them to corresponding sub-folders.

Module 2:

1. Stock Market Analyzer: Use NumPy and pandas to analyze stock market data. Fetch data from a free API like Alpha Vantage, perform operations like calculating moving averages, daily return, and visualize the data using matplotlib or seaborn.
2. Survey Data Analysis: Use a dataset from an online survey tool (like Google Forms) exported as a CSV. Load the data into a pandas DataFrame, clean the data, filter it, and perform statistical analysis.
3. Health Tracker: Design a program that tracks personal health information. Users can input their daily calories, exercise, water intake, etc. Use pandas to store and manipulate this data, and NumPy to calculate weekly averages or other statistics.

Module 3

1. Write a Python program to perform Exploratory Data Analysis
2. Write a Python program to perform Data Operations: Aggregation operations, Joins
3. Write a Python program to perform Data Cleansing: checking missing data, filling missing data, merging operations

Module 4

1. Write a Python program to performLinear Regression, Logistic Regression, Decision Trees, Clustering
2. Write a Python program to perform User Based collaborative filtering, Item Based collaborative filtering, Context Based filtering

References:

1. Provost, Foster, and Fawcett, Tom. Data Science for Business.
2. McKinney, Wes. Python for Data Analysis.
3. Newbold, Paul, and Carlson, William L. Statistics for Business and Economics.
4. Müller, Andreas C., and Guido, Sarah. Introduction to Machine Learning with Python.
5. Madhavan, Samir. Mastering Python for Data Science: Explore the World of Data Science Through Python and Learn How to Make Sense of Data. Packt Publishing.
6. VanderPlas, Jake. Python Data Science Handbook: Essential Tools for Working with Data. O.
7. Liang, Y. D. (2012). Introduction to Programming using Python. Pearson.
8. McKinney, W. (2018). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Ipython (2nd ed.). O'Reilly.
9. Savaliya, M. T., Maurya, R. K., & Magar, G. M. (2021). Programming with Python. SYBGEN Learning India Pvt. Ltd.
10. VanderPlas, J. (2017). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly.

2.2 Major Core

Course Title	**Artificial Intelligence**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**

	• Demonstrate an overview of Artificial Intelligence, recognizing its importance in various fields. • Trace the historical development of AI and identify related fields. • Explain different methods of representing knowledge in AI. • Understand and apply knowledge base systems. • Analyze state space search problems using examples like the 8- Queens, Traveling Salesman, and others. • Understand adversarial search in game scenarios. • Implement the minimax algorithm and comprehend Alpha-Beta Pruning for optimizing game strategies. • Represent simple facts using logic. • Understand computable functions in predicates. • Apply resolution and unification techniques. • Differentiate between forward and backward reasoning
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Demonstrate a comprehensive understanding of Artificial Intelligence (AI), including its definition, objectives, and significance in various domains.
	• Trace the historical development of AI, identifying key milestones, breakthroughs, and influential figures in the field.
	• Explain various methods of representing knowledge in AI, including symbolic, semantic, and sub-symbolic approaches.
	• Understand and discuss the limitations and challenges of AI, Implement and interpret Knowledge Base Systems for organizing and managing information.
Content Outline	• **Introduction:** • Overview of AI, Importance of AI, History, related fields, Representation of Knowledge, Knowledge Base Systems, State Space Search Problem Characteristics of 8- Queens, Traveling Salesman, Missionary & Cannibals, Crypt, Arithmetic, Monkey Banana Problem, Tower of Hanoi and Block World.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Implement DFID to combine the advantages of DFS and BFS. • Assess the efficiency of DFID in terms of time and space complexity. • Apply Greedy Best-First Search to solve optimization problems. • Analyse the role of heuristic functions in guiding the search process.
	• Implement Hill Climbing Search for local optimization. • Recognize the limitations andchallenges associated with hill climbing. • Apply genetic algorithms for optimization and problem-solving.
Content Outline	**Searching Methods:** • Uninformed Search Methods: Breadth First Search (BFS), Depth First Search (DFS), Depth Limited Search, Depth First Iterative Deepening (DFID), • Informed Search Methods: Greedy best first Search, A* Search, Memory bounded heuristic Search. • Local Search Algorithms and Optimization Problems: Hillclimbing search Simulated annealing, Local beam search, Genetic algorithms. • Adversarial Search: Games, Optimal strategies, The minimax algorithm,

	Alpha-Beta Pruning.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Represent and manipulate simple facts using propositional and firstorder logic. • Understand the syntax and semantics of logic representations
Content Outline	Predicate & Logic: • Representing simple facts in Logic -Computable functions in predicates, resolution – unification • forward vs. backward reasoning., Probabilistic reasoning – Bayes's Theorem – Certainty Factors– Demphster– Shafer Theory – Fuzzy, Sets, Reasoning with Fuzzy Logic, Natural Language Computation with Fuzzy Logic.
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Apply understanding of the concept of soft computing, appreciating its significance in solving complex problems where traditional methods may not be effective. • Analyze the distinctions between soft computing and hard computing approaches, explaining their differences, advantages, and disadvantages. • Evaluate the concepts of supervised and unsupervised learning, understanding their roles in machine learning and AI. Supervised learning involves learning from labeled data, while unsupervised learning involves finding patterns in unlabeled data.
Content Outline	• Importance of soft computing Soft computing versus hard computing; Supervised and unsupervised learning; • Introduction to main components of soft computing: Fuzzy logic, Neural networks

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

Activity: Write a report on the role of AI in a specific sector of engineering of your choice, and how it impacts daily activities in that sector. Analyze the difference between natural intelligence and artificial intelligence within the context of this sector. Discuss the significant task domains of AI in this sector and the limitations and challenges faced, including ethical considerations, data privacy concerns, and technical hurdles.

Module 2:

Activity: Choose a complex problem that traditional computing methods struggle to solve effectively. Apply your understanding of soft computing to propose a solution to this problem. Compare and contrast the soft computing approach with a hard computing approach to this problem. Analyze the advantages and disadvantages of each. Finally, design a simple supervised

or unsupervised learning task that could be part of the solution, explaining why you chose that approach.

References:

1. Russel, S., & Norvig, P. (2015). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson.
2. Khemani, D. (2013). A First Course in Artificial Intelligence (1st ed.). Tata McGraw-Hill.
3. Rich, E., Knight, K., & Nair, S. (2009). Artificial Intelligence (3rd ed.). Tata McGraw-Hill.
4. Deva, R. (2014). Artificial Intelligence: A Rational Approach (1st ed.). Shroff Publishers.
5. Das, A. B. (2013). Artificial Intelligence & Soft Computing for Beginners (1st ed.). SPD.

2.3 Major Core

Course Title	**Full Stack Development**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Design, develop, and deploy complete web applications using both frontend and backend technologies. • Apply knowledge of HTML, CSS, JavaScript, and modern frontend frameworks for effective UI/UX. • Implement server-side logic, RESTful APIs, and database integration using backend frameworks. • Utilize version control, development tools, and deployment platforms for full-stack application lifecycle management. • Demonstrate project planning, teamwork, and documentation practices in full stack web development.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Develop static and responsive web pages using HTML5 and CSS3.
	• Apply JavaScript for DOM manipulation, form validation, and interactivity.
	• Use Bootstrap or Tailwind CSS for responsive design and UI components
Content Outline	• **Introduction:** • Introduction to Web Development: Frontend vs Backend, Architecture of Web Applications. • HTML5: Semantic tags, forms, multimedia. • CSS3: Box model, flexbox, grid, media queries. • JavaScript (ES6+): Variables, functions, control structures, DOM manipulation, events. • Introduction to Bootstrap/Tailwind CSS for UI design.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**

	• Use frontend frameworks (like React) for creating dynamic SPAs. • Manage state, props, and component lifecycles effectively. • Apply routing and basic hooks in a React application.
Content Outline	• Introduction to React: JSX, components, props, state. • Functional components and React hooks. • Routing using React Router. • State management basics (Context API or Redux introduction).
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Build server-side logic using Node.js and Express.js. • Create RESTful APIs and connect them to frontend apps. • Handle JSON data, HTTP requests, and middleware.
Content Outline	• Node.js basics: Modules, npm, event-driven architecture. • Express.js framework: Routing, middleware, REST APIs. • HTTP Methods: GET, POST, PUT, DELETE. • API testing tools (Postman). • Introduction to Authentication and Authorization (JWT, sessions).
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Design and integrate databases using MongoDB or MySQL. • Deploy full-stack applications using GitHub and cloud platforms. • Implement continuous integration and basic DevOps practices.
Content Outline	• Introduction to Databases: MongoDB (NoSQL) and MySQL (RDBMS) overview. • CRUD operations with MongoDB using Mongoose ORM. • Project structure and best practices. • Deployment: Netlify, Vercel, Render, or Heroku. • Version Control using Git & GitHub. • CI/CD fundamentals.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Create a responsive portfolio website using HTML/CSS.
- Implement a dynamic form with JavaScript validations.
- Build a single-page responsive UI using Bootstrap/Tailwind.

Module 2:

- Develop a small React app (e.g., Todo list or Weather app).
- Implement routing between pages (home/about/contact).
- Use hooks for state management and effect handling.

Module 3:

- Build a REST API for a blog or e-commerce product catalog.
- Integrate frontend app with backend API.
- Implement login and signup routes using JWT/session-based auth.

Module 4:

- Integrate a MongoDB database into the project.
- Deploy a full-stack app on a cloud platform.
- Maintain source control via GitHub with commit history and branches.

References:

1. **Jon Duckett**, *HTML and CSS: Design and Build Websites*, Wiley.
2. **Ethan Brown**, *Web Development with Node and Express*, O'Reilly.
3. **Robin Wieruch**, *The Road to React*, Self-published.
4. **Brad Traversy**, *Modern Full Stack Development*, Udemy Course (can be referenced as additional learning).
5. **Kyle Simpson**, *You Don't Know JS (book series)*, O'Reilly.
6. **MongoDB University**, Free online materials (https://university.mongodb.com).
7. **MDN Web Docs** (https://developer.mozilla.org/) – For HTML, CSS, JS references.

2.4 Major (Core)

Course Title	**Database Management System**
Course Credits	**4**
Course Outcomes	**After completion of this Course, the students will be able to**
	• Apply and develop proficiency in database languages, particularly SQL, in different scenarios.
	• Understand and analyze SQL commands, functions, data constraints, grouping data, subqueries, joins, and performance tuning.
	• Evaluate the effectiveness and efficiency of these commands and functions in various use cases.
	• Design complex SQL queries and optimize them for better performance.
	• Apply and develop proficiency in NoSQL
Module1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Understand the core concepts of a database system and apply them in practical scenarios. • Analyze and synthesize database designs using E-R and Relational Models • Evaluate the effectiveness and efficiency of these models in different use cases. Design complex databases using E-R and Relational Models for efficient data management
Content Outline	• **Introduction**: Database System application, Database System versus File systems, View of Data, Data Models, Database Languages, Database Users and administrator. DBA Roles and activity, Database system structure. • **Entity-Relational Model**: Basic Concepts,Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features, Design of E-R Database Schema, Reduction of an E-R Schema to Tables •
Module2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Learn and understand Relational model • Apply SQL commands in various scenarios and understand the use of constraints. • Apply relation model to a case study application
Content Outline	• **Relational model**: Structure of Relational Databases • Creation of Schemas, Tables, • **SQL:** SQL commands, Joins, Functions, Data Constraints, Grouping Data,

	Sub-queries, Joins, Performance Tuning, Security Management
Module3 (Credit 1)	
	• Apply SQL commands in various scenarios and understand the use of constraints. • Evaluate and demonstrate the process of decomposing databases and normalizing data using normalization techniques. • Design and analyze database systems with a focus on efficient data retrieval using concepts of hashing and indexing
Content Outline	• **Integrity & Security**: Domain Constraints, Referential Integrity, Privileges in SQL. • **Relational Database Design**: Functional Dependencies, Decomposition, Normalization 1NF-5NF,BCNF
Module4 (Credit 1)	
Learning Outcomes	**After learning this module, learners will be able to**
	• Understand the fundamentals of NoSQL databases and differentiate them from traditional RDBMS. • Apply data modeling techniques suitable for various NoSQL database types (Document, Key-Value, Column-Family, Graph). • Design and execute queries using popular NoSQL systems like MongoDB and Cassandra. • Evaluate use cases for choosing the right NoSQL database based on consistency, availability, and scalability needs.
Content Outline	• Introduction to NoSQL: Need, Characteristics, and CAP Theorem • Types of NoSQL Databases: Document, Key-Value, Column-Family, Graph • Basic CRUD Operations in MongoDB (Insert, Find, Update, Delete) • Indexing and Aggregation in MongoDB • Data Modeling in NoSQL: Embedding vs. Referencing • Understanding Partitioning, Replication, and Consistency in NoSQL

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

Activity 1: Design and implement a database schema for a hypothetical business scenario. The schema should include appropriate constraints to maintain data integrity.

Activity 2: Normalize the database schema created in Activity 1. Write a report detailing the normalization process and its effects on database performance and integrity.

Activity 3: Using the same business scenario, write SQL queries to demonstrate the use of different types of joins and clauses. Analyze the results and discuss their implications for data management.

Module 2:

Activity 1: Write a PL/SQL script that includes the use of procedures, functions, triggers, and cursors. Execute the script and document the results.

Activity 2: Analyze the behavior and performance of the PL/SQL elements used in your script. Discuss their effectiveness and efficiency in the context of the hypothetical business scenario.

Activity 3: Modify your PL/SQL script to improve database management efficiency. Explain the changes made and their expected impact on database management.

References:

1. Korth, H., Silberschatz, A., & Sudarshan, S. (2005). Database System Concepts (5th ed.). McGraw-Hill.
2. Elmasri, R., & Navathe, S. B. (2008). Fundamentals of Database Systems (3rd ed.). Pearson Education India.
3. Raj P, Deka GC. A deep dive into NoSQL databases: the use cases and applications. Academic Press; 2018 Apr 20

2.5 Major (Elective-II) -A

Course Title	**Ethical Hacking Major (Elective) Theory**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Understand the core concepts related to malware, hardware and software vulnerabilities and their causes. • Understand ethics behind hacking and vulnerability disclosure. • Appreciate the Cyber Laws and impact of hacking. • Exploit the vulnerabilities related to computer system and networks using state of the art tools and technologies.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Define and understand fundamental concept of Ethical hacking. • Understand ethics of hacking • Comprehend legal surrounding of ethical hacking. • Apply protocols for proper and ethical disclosure of security vulnerabilities.
Content Outline	• Ethics of Ethical Hacking, Ethical Hacking And the legal system, Proper and Ethical Disclosure
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Used Penetration testing tool efficiently. • Develop skills using Metasploit which is penetration testing tool with demonstration of BackTrack.
Content Outline	• Using Metasploit, Using BackTrack Live CDLinux Distribution.
Module 3(Credit 1)	

Learning Outcomes	**After learning the module, learners will be able to**
	• Understand and differentiate different vulnerability analysis technique. • Develop expertise in advanced reverse engineering methodologies • Apply Sulley's Intelligent fuzzing technique to find exploit weaknesses.
Content Outline	• Vulnerability Analysis: Passive Analysis, Advanced Static Analysis with IDA Pro, Advanced Reverse Engineering. • Client-side browser exploits: Exploiting Windows Access Control Model for Local Elevation Privilege, Intelligent Fuzzing with Sulley, From Vulnerability to Exploit.
Module 4(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Understand Fundamentals of malware • Define the Hacking malware. • Describe way to collect and analyse the malware • Illustrate the case study
Content Outline	• Consider given/ real-time security scenario. • Apply the penetration testing using penetration tool • Perform vulnerability analysis on scenario. • Check for client side browsing exploits using Sulley's Intelligent fuzzing. • Perform malware analysis by collecting and analysing malware. • Explore case study.

References:

1. Harris, Shon, Allen Harper, Chris Eagle, and Jonathan Ness.
2. Gray Hat Hacking: The Ethical Hackers Handbook. TMH Edition.
3. Erickson, Jon. Hacki

2.5 Major (Elective-II) -B

Course Title	Introduction to Data Science
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	• Analyze different types and sources of data, along with the techniques to collect and interpret it. • Apply understanding of various statistical concepts like distributions, hypothesis testing, confidence intervals, and correlation • Evaluate various machine learning algorithms such as Linear Regression, Logistic Regression, Decision Trees, Clustering, etc. • Design and implement text mining techniques to analyze unstructured data, possibly including hands-on lab sessions for practical understanding.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Analyze various types of data, identifying and differentiating between them based on their characteristics and use cases.

	• Apply understanding of data analysis to construct models for predictive or descriptive purposes. • Evaluate and implement feature extraction methodologies to enhance model performance and interpretability. • Design illustrations to explain the stages within the Data Science lifecycle and their significance in problem-solving.
Content Outline	• Introduction to Data :What is Data? Different kinds of data, Data Sources, Different types of data sources, • Exploratory Data Analysis (EDA), Data Science lifecycle, Data Collection • Data Extraction, Data Analysis & Modeling • Data transformations: Dimension reduction, Feature extraction, Smoothing and aggregating.
Module 2 (credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Analyze and perform various mathematical operations efficiently on NumPy arrays, a powerful tool for mathematical computation in Python. • Apply understanding of the data structures provided by Pandas (Series, DataFrame) and utilize them effectively in data analysis tasks. • Evaluate and execute merging operations to combine datasets efficiently for comprehensive analysis. • Design strategies to apply various types of joins effectively to merge datasets based on common columns or indices.
Content Outline	• The World of arrays with Numpy: creating an array, Mathematical operations, Indexing and slicing, Shape manipulation. • Empowering Data analysis with pandas: the data structure of pandas, Inserting and exporting data • Data Cleansing: checking missing data, filling missing data, merging operations • Data Operations: Aggregation operations, Joins
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Analyze various probability distributions (normal, binomial, Poisson, etc.), identifying and differentiating between them based on their characteristics and real-world applications. • Apply control over line properties in various chart types for better visualization, enhancing the clarity and effectiveness of data representation. • Evaluate and differentiate between various types of Machine Learning (supervised, unsupervised, reinforcement learning), understanding their unique characteristics and applications
Content Outline	• Inferential Statistics: Various forms of distribution, z-score, p-value, Type 1 and Type 2 errors, Confidence Interval, Correlation, Chi-square distribution, ANOVA • Making Sense of Data Through Visualization: Controlling the line properties of a chart, creating multiple plots, styling your plots, Boxplots, Heatmaps, Scatter plots with histogram, Bubble charts • Uncovering Machine Learning: Different types of Machine Learning, Linear Regression, Logistic Regression, Decision Tree, K-means Clustering, Hierarchical Clustering
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**

	• Apply understanding of User-Based Collaborative Filtering techniques to generate recommendations based on user similarities. • Analyze and explain the theory behind text mining and its applications in analyzing unstructured data, providing clear and comprehensive insights.
Content Outline	• Generating Recommendations Systems: User Based collaborative filtering, Item Based collaborative filtering, Context Based filtering • Case Study Theory: Analyzing Unstructured Data using Text mining techniques. (Case Study Practical Implementation to be performed in lab as part of Practical's)

Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1: Activity: Conduct an exploratory data analysis on a dataset of your choice. Identify the different types of data present, apply your understanding of data analysis to construct predictive models, implement feature extraction methodologies to enhance model performance, and design illustrations to explain your process and findings.

Module 2: Activity: Using Python, create a NumPy array and perform various mathematical operations on it. Then, use a dataset of your choice and apply your understanding of data structures using Pandas (Series, DataFrame) for data analysis tasks. Perform merging operations and apply various types of joins to combine datasets.

Module 3: Activity: Analyze different probability distributions (normal, binomial, Poisson, etc.) and their real-world applications. Create different types of charts with controlled line properties for better visualization. Lastly, differentiate between various types of Machine Learning (supervised, unsupervised, reinforcement learning), and implement one of them on a dataset.

Module 4: Activity: Implement a User-Based Collaborative Filtering technique to generate recommendations based on user similarities on a dataset. Explain the theory behind your approach. Also, find a set of unstructured data and apply text mining techniques to analyze it. Provide comprehensive insights from your analysis.

References:-

1. Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly Publication.

2. McKinney, W. (2017). Python for Data Analysis. O'Reilly Publication.

3. Newbold, P., & Carlson, W. L. (2012). Statistics for Business and Economics. Pearson Publication

4. Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. O'Reilly Publication.

5. Madhavan, S. (2015). Mastering Python for Data Science: Explore the World of Data Science Through Python and Learn How to Make Sense of Data. Packt Publishing. Packt Publishing.

6. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly.

2.5 Major (Elective-II) -C

Course Title	**Mobile App Development Technologies**
Course Credits	**4**
Course Outcomes	**After Completion of this Course, students will be able to** • Understand mobile app architecture, platforms (Android, iOS), and development environments. • Develop responsive, user-friendly mobile applications using cross-platform frameworks. • Integrate device features like camera, location, and sensors into mobile applications. • Deploy mobile apps and understand app publishing, testing, and security practices.
Module 1: Mobile App Fundamentals and Development Tools(Credit 1)	
Learning Outcomes	**After learning this module, learners will be able to** • Explain mobile platform differences and app types (native, hybrid, web). • Set up development environments (Android Studio, Flutter, etc.) • Understand app architecture, UI guidelines, and lifecycle.
Content Outline	• Mobile ecosystem: Android vs iOS, hybrid vs native vs web apps • App architecture (MVC, MVVM) • IDEs: Android Studio, Visual Studio Code • Flutter, React Native, Kotlin, Swift overview • App structure: activities, fragments, views, layouts • Emulator setup and debugging basics
Module 2: UI Design, Layouts & Navigation (Credit 1)	
Learning Outcomes	**After learning this module, learners will be able to** • Design and implement interactive and responsive UI. • Handle user interaction with forms, buttons, and menus. • Navigate between screens using routing/navigation techniques.
Content Outline	• Flutter widgets: text, images, buttons, input fields • Layouts: Column, Row, Stack, GridView • Navigation and routing between screens • State management (Provider, setState, Bloc – overview) • App theme and localization
Module 3: Device Integration & Data Handling (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to** • Access mobile hardware features like GPS, camera, and accelerometer. • Read/write to internal and external storage. • Connect mobile apps to databases and APIs.

Content Outline	• Accessing sensors: camera, GPS, accelerometer • Handling permissions • Local storage: SharedPreferences, SQLite, Hive • REST APIs: fetching and sending data using HTTP • JSON parsing • Firebase overview (Auth, Realtime DB)
Module 4: App Testing, Deployment & Security (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to** • Understand app testing strategies and performance optimization. • Prepare and publish apps to Play Store/App Store. • Apply security and privacy measures in mobile apps.
Content Outline	• Debugging and logging • Unit testing and UI testing (Flutter Test, Espresso – overview) • App performance tips: lazy loading, image optimization • App signing, APK generation • Publishing to Google Play Store (steps, guidelines) • Best practices: secure API handling, data encryption

Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Build a basic "Hello World" app in Flutter and explore its UI structure.

Module 2:

- Create a multi-screen user registration/login app with form validation.

Module 3:

- Build a location-aware app that captures photos and stores them with GPS data.

Module 4:

Project: Fully functional app (e.g., To-do app, Weather app, Expense Tracker) with storage, navigation, and API integration.

Two sample projects are given below

Project 1: Design a mobile application to help users track their daily health and fitness activities like steps, calories burned, water intake, and sleep hours.

Project 2: Create a mobile app to manage daily expenses, categorize spending, and generate simple reports for budgeting.

References:-

1. Flutter in Action – Eric Windmill
2. Android Programming: The Big Nerd Ranch Guide – Bill Phillips

3. Beginning iOS Programming with Swift – Simon Ng
4. Pro Android with Kotlin – Peter Späth

2.7 OJT

Course Title	**On Job Training (OJT)**
Course Credits	**2 Credits**