गोंय विद्यापीठ
ताळगांव पठार,
गोंय – ४०३ २०६
फोन : +९१-८६६९६०९०४८

Goa University
Taleigao Plateau, Goa-403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

(Accredited by NAAC)

GU/Acad –PG/BoS -NEP/2025-26/236 Date: 09.07.2025

CIRCULAR

The Academic Council & Executive Council of the University has approved Ordinance OA-35A relating to PG Programmes offered at the University campus and its affiliated Colleges based on UGC 'Curriculum and Credit Framework for Postgraduate Programmes'. Accordingly, the University has proposed introduction of Ordinance OA-35A from the Academic year 2025-2026 onwards.

The Programme structure and syllabus of Semester I and II of the **Master of Computer Applications** Programme approved by the Academic Council in its meeting held on 13th & 14th June 2025 is attached.

The Dean & Vice-Dean (Academic) of the Goa Business School are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

ASHWIN VYAS LAWANDE
Digitally signed by ASHWIN VYAS LAWANDE
Date: 2025.07.09 11:14:24 +05'30'

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,
1. The Dean, Goa Business School, Goa University.
2. The Vice-Dean (Academic), Goa Business School, Goa University.

Copy to:
1. Chairperson, BoS in Computer Science & Technology, Goa University.
2. Programme Director, M.C.A., Goa University.
3. Controller of Examinations, Goa University.
4. Assistant Registrar Examinations (PG), Goa University.
5. Director, Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

GOA UNIVERSITY

MASTER OF COMPUTER APPLICATIONS

(Effective from the Academic Year 2025-2026)

ABOUT THE PROGRAMME
The Master of Computer Applications (MCA) is a two year postgraduate degree program that focuses on providing students with advanced knowledge and skills in computer applications, preparing them for careers in software development, IT management, and emerging technologies.

OBJECTIVES OF THE PROGRAMME
To create professionals who are employable in the software industry, corporate sector, academia, entrepreneurial pursuit and other IT services based on the AICTE and NEP guidelines.

PROGRAMME SPECIFIC OUTCOMES (PSO)	
PSO 1.	Apply the knowledge of computing fundamentals, mathematics and appropriate domain knowledge to design, implement and maintain real world economically feasible solutions.
PSO 2.	Apply appropriate techniques, resources, and IT tools including prediction and modeling of complex environments with an understanding of their limitations.
PSO 3.	Understand the importance of Professional ethics and social responsibilities while utilizing the computing knowledge in interdisciplinary domains with a concern for societal, environment, and cultural aspects.
PSO 4.	To develop communication skills, teamwork abilities and leadership qualities required for their professional multidisciplinary projects.
PSO 5.	Recognize the need for, and have the passion and ability to engage in independent and life-long learning in the broadest context of ever changing technological landscape.
PSO 6.	To cultivate a rational, objective and critical approach to understanding the world.

Approved by: Academic Council on 13th & 14th June 2025

PROGRAMME STRUCTURE

Master of Computer Applications

Effective from Academic Year 2025-26

BRIDGE COURSE			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	CSA-1000	Bridge Course for MCA	2

SEMESTER I				
Discipline Specific Core (DSC) Courses (16 credits)				
Sr. No.	**Course Code**	**Title of the Course**	**Credits**	**Level**
1	CSA-5000	Operating Systems	(3T)	400
2	CSA-5001	Object Oriented Software Engineering	(4T)	400
3	CSA-5002	Internet Technology	(3T)	400
4	CSA-5003	Problem Solving and Programming Lab	(2P)	400
5	CSA-5004	Linux Lab	(2P)	400
6	CSA-5005	Object Oriented Technology Lab	(2P)	400
Total Credits for DSC Courses in Semester I			**16**	
Discipline Specific Elective (DSE) Course (4 credits)				
Sr. No.	**Course Code**	**Title of the Course**	**Credits**	**Level**
1	CSA-5201	Mathematics for Computer Science -I	(2T)	400
2	CSA-5202	Fundamentals of Data Science	(2T)	400
3	CSA-5203	Operations Research	(4T)	400
Total Credits for DSE Courses in Semester I			**4**	
Total Credits in Semester I			**20**	

SEMESTER II				
Discipline Specific Core (DSC) Courses				
Sr. No.	**Course Code**	**Title of the Course**	**Credits**	**Level**
1	CSA-5006	Data Structures and Algorithms	(2T)	500
2	CSA-5007	Database Management Systems	(3T)	500
3	CSA-5008	Web Development	(1T)	500
4	CSA-5009	Machine Learning	(2T)	500
5	CSA-5010	Data Structures and Algorithms Lab	(2P)	500
6	CSA-5011	Database Management Systems Lab	(2P)	500
7	CSA-5012	Web Development Lab	(2P)	500
8	CSA-5013	Machine Learning Lab	(2P)	500
Total Credits for DSC Courses in Semester II			**16**	
Discipline Specific Elective (DSE) Courses (4 credits)				
Sr. No.	**Course Code**	**Title of the Course**	**Credits**	**Level**
1	CSA-5204	Mathematics for Computer Science - II	(2T)	400
2	CSA-5205	Secure Coding	(2T)	400
3	CSA-5206	Data Mining	(2T)	400
Total Credits for DSE Courses in Semester II			**4**	
Total Credits in Semester II			**20**	

Blooms Taxonomy Cognitive Levels	
Cognitive Level	**Notations**
K1	Remembering
K2	Understanding
K3	Applying
K4	Analyzing
K5	Evaluating
K6	Create

Title of the Course	Bridge course for MCA
Course Code	CSA-1000
Number of Credits	2
Theory/Practical	Theory and Practical
Level	Nil
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	Yes
Course for advanced learners	No

Pre-requisites for the Course:	Nil			
Course Objectives:	To equip students with foundational knowledge and skills in core areas of computer science, including mathematics, programming in C, computer organization, operating systems, object-oriented programming, software engineering, web development, database management systems, and internet technologies, preparing them for advanced study and practical application in the field.			
Course Outcomes:	CO 1. To understand the basic of mathematics CO 2. To understand the basics of programming in C. CO 3. To understand the basics of COA and OS. CO 4. To understand the basics of OOPs and Software Engineering. CO 5. To understand the basics of Web Development. CO 6. To understand the basics of DBMS and Internet Technology.			
Content:		**No of**	**Mapped**	**Cognitive**

		hours	to CO	Level
Module 1:	1.1 Mathematics: Set Theory, Probability and Statistics, Logarithms, Geometric and Harmonic progressions, Determinants and Matrices, Coordinate Geometry & Applications. Basic Calculus: Limit of functions, continuous function, differentiation of function, Integration and their applications. Trigonometry & applications. Vectors: Concepts of vectors & vector algebra, applications of Vectors. Fundamentals of logic, Relations and Functions, Counting Techniques: Basics of Counting, Pigeonhole Principle, Recurrence relations, Graphs: Basic concepts of Graph and its applications. Introduction to trees, Applications of trees, Boolean Algebra and Circuits.	**4**	CO1	K1, K2
	1.2 Programming in C and Data Structures Introduction to Algorithms, Flow charts, Pseudocode, Assembly language and high-level language, Basic Programming: keywords, tokens, identifiers, basic data types, constants, and variables, operators (arithmetic, relational, logical, bitwise), enumerated data types, sequence control, looping controls, arrays, strings, functions, pass-by-value and pass-by-reference, structures, and unions. Data Structures: Abstract data types, stacks, queues, Singly Linked Lists. Basic sorting algorithms: sorting (bubble, selection, insertion) and searching (linear search, binary search)	**6**	CO2	K1, K2
	1.3 Operating Systems : Input-Output Unit, Structure and functions of Central Processing Unit,Von Neumann Machine Architecture, Interconnection structures, Bus Interconnection. Conversion (Binary, Decimal, Octal, Hexa-Decimal), Data Representation, Binary Arithmetic, Data representation, Number System, Signed number, fixed, floating point, character representation, Addition, Subtraction, Multiplication, Hierarchical memory organization, Types of Memory-internal and external, Cache memory, Memory interleaving,	**8**	CO3	K1, K2

	Peripheral devices: Types of Peripheral Devices, I/O subsystem,programmed I/O, Interrupt-driven I/O, DMA, I/O channels and processors Instruction Set Architecture (ISA), Processor Organization, Registers organization, Instruction Execution Cycle, Instruction formats, Addressing Modes. Need of OS, Computer Systems Organization and Architecture, Operating Systems Services, System Programs, System Booting process, Storage management - Overview of Mass storage structure, Disk Structure, Disk Scheduling, Swap Space management, RAID Levels, File System Concept and Access methods, Directory Structures, File Protection, File Sharing, File System Implementation, Directory implementation, File Allocation Strategies, Free Space (disk) management, Concept of main Memory.			
	1.4 OOPs: Class, Object, Principles of OOP, Benefits of OOP, Applications of OOP, OOP Languages, Data Abstraction, Encapsulation, Inheritance, Types of Inheritance, Polymorphism, Types of Polymorphism, Message Passing, Dynamic Binding, Exceptions, Errors and Types of Errors, Static and Non-static members, Access specifiers, Abstract base class, Abstract Methods, Virtual class, Pure virtual class, Generics in Java, Introduction to OO analysis and design using UML.	**6**	CO4	K1, K2
	1.5 Software Engineering: Basic Software Development life cycle (SDLC), SDLC models.			
	1.6 Internet Technology: Types of Networks, Network Topologies, OSI Model, TCP/IP Protocol Suite, Network Design and Architecture, Introduction to Network Devices, TCP, UDP, IP, DHCP, DNS, FTP, Ethernet and WIFI	**4**	CO6	K1, K2
Module 2:	2.1 DBMS: Candidate key identification, functional dependency, normalization (1NF, 2NF and 3NF), ER Diagram, Types of Entities and Relationships, Keys and integrity constraints,Converting ER Diagram to Tables, Normal Forms , DDL and basic DML	**9**	CO6	K1, K2

	statements including joins.			
	2.2 Web Development Basic: HTML5, CSS3 , JS Introduction to internet and web design. Basic concepts of web architecture. Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images. Cascading style sheet (CSS): Concept of CSS, creating style sheet, importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, working with block elements and objects, working with lists and tables, CSS id and class, box model. Basics of JavaScript: Document object model, data types and variables, functions, methods and events, controlling program flow, built-in objects and operators, validations.	**8**	CO5	K1, K2
Pedagogy:	Self study			
Texts:	1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2006). Operating system principles. John Wiley & Sons. 2. Hanly, J. R., & Koffman, E. B. (2012). Problem solving and program design in C (7th ed.). Pearson Education. 3. Deitel, H. M., & Deitel, P. J. (2001). C: How to program. Addison Wesley.Balagurusamy, E. (2004). Programming in ANSI C. Tata McGraw-Hill. 4. Horowitz, Ellis, Sartaj Sahni, and Susan Anderson-Freed. “Fundamentals of data structures in C” WH Freeman & Co., Latest Edition. 5. Robert W. Sebesta, “Programming the World Wide Web”, Pearson Education. 6. Korth, Silberchartz, “ Database System Concepts” McGrawhill Publication			

SEMESTER I

Discipline Specific Core Courses

Title of the Course	Operating System
Course Code	CSA-5000
Number of Credits	3
Theory/Practical	Theory
Level	400
Effective from AY	2025 -26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	This course focuses on the principles, understanding and application of fundamental concepts of process management, memory management and interprocess communication in an operating system and evaluates their trade-off in various environments.	
Course Outcomes:		**Mapped to PSO**
	CO 1. To understand the design and services provided by an operating system.	PSO1, PSO2
	CO 2. To understand the concept, states and transitions of processes and to analyze and compare different scheduling algorithms based on their performance characteristic	PSO1, PSO2, PSO5, PSO6

	CO 3. To understand and apply techniques of interprocess communication and synchronization between processes using semaphores in writing programs		PSO1, PSO2, PSO5	
	CO 4. To understand various deadlock handling mechanisms and resolution techniques		PSO1, PSO2	
	CO 5. To understand and apply the knowledge of threads while writing code.		PSO1, PSO2, PSO5, PSO6	
	CO 6. To understand various memory management schemes and techniques.		PSO1, PSO2, PSO5	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Introduction and Systems Structures Overview of Computing Environments and Operating Systems Services,	**2**	CO1	K1, K2
	1.2 Process Management Process - Concept and states, Process Creation and Control, Scheduling Criteria, Scheduling Algorithms, MultiLevel Queues, Multiprocessor scheduling	**7**	CO2, CO3	K2
	1.3 Threads Motivation and Challenges, Multithreading Models, Threading Issues, Thread libraries, Thread scheduling	**6**	CO5	K2, K3, K4
Module 2:	2.1 Process Synchronization Cooperating processes and Race Conditions, The critical-section problem, Peterson's solution, mutex locks, Synchronization Hardware, Semaphores and their Implementation, Classic problems of synchronization	**5**	CO3	K2, K3, K4
	2.2 Inter process Communication Overview of IPC, IPC mechanisms in Client	**5**	CO3	K2, K3, K4

	Server Systems			
	2.3 Deadlocks: System Model, Deadlock characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery From Deadlock.	**5**	CO4	K2, K3
Module 3:	3.1 Memory Management: Hardware Support, Address Binding, Swapping, Contiguous Memory Allocation, Fragmentation, Memory Protection, Paging, Segmentation, Example: Intel architecture.	**5**	CO6	K2
	3.2 Virtual-Memory Management: Background, Demand Paging, Page Replacement, algorithms, Allocation of Frames, Thrashing, Allocating Kernel, Memory. System Structure and directory implementation.	**10**	CO6	K2
Pedagogy:	Lectures/ tutorials/assignments/class presentations and debates/peer reviews.			
Texts:	Silberschatz, A., Galvin, P. B., & Gagne, G. (2006). Operating system principles. John Wiley & Sons.			
References/ Readings:	1. Deitel, H. M. (1990). An introduction to operating systems. Addison-Wesley Longman Publishing Co., Inc.. 2. Milenkovic, M. (1992). Operating systems: concepts and design. McGraw-Hill, Inc.. 3. Tanenbaum A. S., Modern Operating Systems", Prentice Hall of India Pvt. Ltd., Latest Edition 4. Tanenbaum, A. S. (2001). Modern operating systems, prentice hall. Inc., Upper Saddle River, NJ.			
Web Resources:	Operating System Concepts. Available at: http://www.os-book.com/ (Accessed: 22 May 2025).			

Title of the Course	Object Oriented Software Engineering
Course Code	CSA-5001
Number of Credits	4
Theory/Practical	Theory
Level	400
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	To equip students with foundational knowledge of object-oriented principles and their application in software analysis, design, and testing, enabling them to model and develop robust, modular, and maintainable software solutions to real-world problems	
Course Outcomes:		**Mapped to PSO**
	CO 1. Explain the challenges in software engineering and describe various software development life cycle models and supporting processes.	PSO1, PSO6
	CO 2. Apply object-oriented principles such as abstraction, encapsulation, inheritance, and polymorphism to develop modular and reusable software components	PSO1, PSO2
	CO 3. Analyze problem statements to identify classes, objects, responsibilities, and interactions using techniques like CRC cards and UML diagrams.	PSO1, PSO2

	CO 4. Evaluate object-oriented designs using principles of modularity, cohesion, and coupling, and assess architectural styles such as MVC and layered architecture.		PSO1, PSO2, PSO6	
	CO 5. Develop software models using UML diagrams including class, sequence, use case, and activity diagrams to support object-oriented analysis and design.		PSO1, PSO2, PSO4	
	CO 6. Design and implement a testable object-oriented solution for a real-world problem using tools like JUnit and Selenium within a CI/CD pipeline.		PSO1, PSO2, PSO4, PSO5	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	Foundations of Software Engineering (a) Programming in the small versus Programming in the large, Industrial strength software (b) Major problems with Software- (i) Expensive (ii) Late & Unreliable (iii) Maintenance & Rework. (c) Software Engineering Challenges: (i) Scale (ii) Quality and Productivity (iii) Consistency & Repeatability (iv) Change. (d) Software Engineering Approach: phased Development process, managing the process; (e) Software Development Life cycle Models (i) Software Process: Process and Process Models, ETVX approach for process specification. (ii) Desired characteristics of Software Process: predictability, support testability and maintainability, support change, early defect removal, process improvement and feedback. (iii) Software Development Process Models: waterfall, prototyping, iterative development, Spiral, timeboxing, agile models.	**15**	CO1, CO 4	K2, K5

	(iv) Other software processes: project management process, inspection process, software configuration process, requirements change management process, process management process			
Module 2:	Introduction to Object Oriented Technology: (a) Software Quality (i) External and Internal factors (ii) Three Criteria of Object Orientation-Method and language, Implementation and Environment, Libraries (b) Modularity- criteria, rules, principles. (c) Object Oriented Programming: (i) The static structure: Classes, role of classes- module and type, uniform type system, OO style of computation (ii) The run time structure: Objects, references, object construction and destruction, overloading, Memory Management: three modes of object management, attachment/detachment, reachable/unreachable objects. OO Memory management approaches, automatic memory management (iii) Introduction to Inheritance: Single inheritance, overriding, abstract classes, interfaces (iv) Exception Handling & I/O handling: Exception hierarchy, I/O: console based and file handling. (v) Genericity and collections : Horizontal and vertical type generalization. Generic classes, collection frameworks (vi) Multiple Inheritance, inheritance Techniques, Using Inheritance well (vii) Typing and Binding- static typing and dynamic binding and its advantages (viii) Concurrency: Thread mechanism (ix) Reflection, Persistence, assertions (x) Library functions, APIs, Frameworks	**15**	CO2, CO6	K3, K6

Module 3:	Object Oriented Software Engineering (OOSE) using UML(Unified Modeling Language) (a) Assigning responsibilities: Identifying classes- Noun-Verb analysis; Class-Responsibility-Collaboration (CRC cards); (b) Introduction to UML: Main UML diagrams- class diagram, sequence diagram, Object Diagram, Activity diagram, Use Case diagram. (c) OO Analysis. Use case modeling & specification, Use case realization using sequence and activity diagrams. Generalization, includes, extends, creating analysis model using boundary, control and entity stereotypes (d) Object Oriented Design: Software Design Principles- Modularity, abstraction, cohesion, coupling; Design class diagram, visibility. Mapping Analysis model to Design Model. Association class, qualified associations, reflexive associations, ordered associations. Sequence diagram, activity diagram, state chart diagram. Code generation from UML diagrams and reverse engineering. (e) Advanced Class diagram: Concepts, attributes, operations. Association, Aggregation, composition and containment, generalization and interfaces, delegation versus inheritance (f) Other UML diagrams: component, package, deployment diagram. (g) Software Architecture: Software architectural styles: MVC, Layered, Microservices	**15**	CO3, CO4, CO5	K3. K4, K5
Module 4:	Object Oriented Testing (a) Testing and Quality Assurance: Software Testing and Quality Assurance, Object Oriented Metrics (b) Testing principles and levels: unit testing, integration testing, system testing, acceptance testing (c) Test case design & Methods: Testing Approaches: black-box , white-box, glass box; Testing methods: Cyclomatic complexity(CC), boundary value, equivalence	**15**	CO6	K6

	class partitioning; Testing Strategies: Top down, Bottom up, Integration testing, continuous Integration, CI/CD pipelines with integrated testing. (d) Test-driven development (TDD). Manual vs automated testing. Testing tools: JUnit (unit testing), Selenium (UI testing). Code coverage and quality metrics.			
Pedagogy:	Lectures/Tutorials, Flipped classroom, assignments, peer-teaching, role playing, games			
Texts:	1. Meyer, B. (2000). Object-oriented software construction (2nd ed.). PTR Prentice Hall Pearson. 2. Jalote, P. (2005). An integrated approach to software engineering (3rd ed.). Narosa Publishing House. 3. Fowler, M. (2018). UML distilled: A brief guide to the standard object modeling language (3rd ed.). Addison-Wesley. 4. Jacobson, I. (1992). Object oriented software engineering: A use case driven approach. ACM Press. 5. McGregor, J. D., & Sykes, D. A. (2001). A practical guide to testing object-oriented software. Addison-Wesley Professional.			
References/ Readings:	1. Timothy Budd (2001), "An Introduction to Object Oriented Programming", 3rd Edition, Pearson Education. 2. Mughal, K. A., & Rasmussen, R. W. (2003). A programmer's guide to Java certification: a comprehensive primer. Addison-Wesley Professional. 3. Arnold, K., Gosling, J., & Holmes, D. (2005). The Java programming language. Addison Wesley Professional. 4. Stroustrup, B. (2013). The C++ programming language. Pearson Education. 5. Glenford J. Myers, (1979), "The Art of Software Testing" (1st edition).			
Web Resources:	1. Tutorials Point. *UML Tutorial*. Tutorials Point. https://www.tutorialspoint.com/uml/index.htm Accessed May 22, 2025. 2. Tutorials Point. Java - OOPs concepts. Tutorials Point. https://www.tutorialspoint.com/java/java_oops_concepts.htm Accessed May 22, 2025 3. https://c2.com/xp/ExtremeProgramming.html Accessed May 22, 2025 4. https://martinfowler.com/ Accessed May 22, 2025			

Title of the Course	Internet Technology
Course Code	CSA-5002
Number of Credits	3
Theory/Practical	Theory
Level	400
Effective from AY	2025 -26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil			
Course Objectives:	The objective of the course is to introduce the TCP/IP architecture and allied protocols of the Internet by following a top-down approach.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. Developing understanding of layered communication architecture (TCP/IP) and knowledge of some of the important networking protocols		PSO1, PSO2	
	CO 2. Analyze the functionality of core protocols		PSO1, PSO2	
	CO 3. Understand the concepts of reliable data transfer and how TCP implements these concepts.		PSO1, PSO2	
	CO 4. Basic knowledge of routing algorithms and security in computer networks.		PSO1, PSO2	
Content:		**No of**	**Mapped**	**Cognitive**

		hours	to CO	Level
Module 1:	1.1 Computer Networks and the Internet: Networking and Internetworks, Internetworking devices, Internet: Network edge, and the Network core. TCP/IP protocol stack: Protocol stack, Connection-oriented, connectionless services, Packet switching, circuit switching, Delay, Loss, and Throughput in Packet-Switched Networks.	**7**	CO1	K1
	1.2 Application layer: Principles of Application Layer Protocols, the Web and HTTP, MIME, mail access protocols, DNS, Peer to Peer Applications, Video Streaming, and Content Distribution Networks.	**8**	CO2	K2, K4
Module 2:	2.1 Transport layer: Transport-layer services, Multiplexing and demultiplexing, UDP protocol, Principles of reliable data transfer, Connection-oriented transport - TCP protocol, Principles of congestion control, TCP congestion control.	**8**	CO2, CO3	K2, K4
	2.2 Network layer: Packet switching: virtual circuit & datagram networks, Forwarding and Routing (Network Data and control planes). The Internet Protocol (IP): IPv4 Datagram format, fragmentation, IPv4 Addressing in the Internet, route aggregation, subnetting, CIDR, Network Address Translation, DHCP, ICMP.	**7**	CO4	K2, K4
Module 3:	3.1 Control Plane: Routing protocols- shortest path, link state routing algorithm, distance vector routing. Autonomous Systems (AS), IntraAS Routing in the Internet: OSPF, Internet routing: RIP, OSPF, BGP, Address Resolution Protocol (ARP), and RARP.	**5**	CO4	K2, K3, K4
	3.2 Wireless and Mobile Networks: WiFi (802.11 Wireless LAN), Bluetooth, and Cellular Internet Access. Security in Computer Networks: Basic cryptography concepts, Secure Socket Layer (SSL), Internet Security Protocol (IPSec), Virtual Private Network (VPN).	**10**	CO4	K2, K3, K4
Pedagogy:	Lectures/ Tutorials/Assignments/ Flipped classroom			
Texts:	1. Kurose, J. F., & Ross, K. W. (2017). Computer networking: A top-down approach (6th ed.). Pearson.			

	2. Tanenbaum, A. S. (2011). Computer networks (5th ed.). Prentice Hall of India.
References/ Readings:	Forouzan, B. A., & Mosharraf, F. (2012). Computer networks: A top-down approach. McGraw-Hill.

Title of the Course	Problem Solving and Programming Lab
Course Code	CSA-5003
Number of Credits	2
Theory/Practical	Practical
Level	400
Effective from AY	2025 - 26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil			
Course Objectives:	To introduce the principles of computational thinking and structured problem solving using programming language.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To provide the fundamental concepts of computational thinking and algorithmic problem solving.		PSO1, PSO2	
	CO 2. To design flowcharts and algorithms for simple computational problems.		PSO1, PSO2	
	CO 3. To use arrays, strings, and pointers to process data efficiently, simulating memory and system-level tasks.		PSO1, PSO2	
	CO 4. To develop modular and reusable programs to solve real world problems.		PSO1, PSO2, PSO6	
Content:		**No of**	**Mapped**	**Cognitive**

		hours	to CO	Level
Module 1:	1.1 Computational Thinking & Problem Solving: Introduction to computational thinking: decomposition, pattern recognition, abstraction, algorithms, Steps in problem solving, Designing algorithms and flowcharts, Case studies: Simple real-life problem modeling.	**4**	CO1, CO2	K2, K3
	1.2 Basics of Programming: Structure of a program, compiling and running, Data types, variables, constants, Operators and expressions, Input/output functions, Writing simple programs.	**8**	CO1, CO2	K2, K3
	1.3 Control Structures and Functions, Conditional statements: if, if-else, switch Looping: for, while, do-while Functions: declaration, definition, call, return Use of recursion (simple examples) Write programs:Menu-driven programs (e.g., calculator, unit converter), Simulate system tasks: memory allocation view, page navigation	**8**	CO3	K3
	1.4 Arrays & Strings — 1D/2D arrays: search, sort, filter (use cases: student marks, stock prices), String handling and validation logic, searching (linear). Write Program: Tabular DBMS-like records (CRUD with arrays), Employee records (like HR system) with filters and search.	**10**	CO3	K3, K4
Module 2:	2.1 Functions & Modular Thinking: Functions, scope, header files, Modular design (interface vs logic separation), Simulate functional decomposition. Write programs: Recursive calculator, factorial, GCD/LCM, Separate logic into modules for billing system or student result portal.	**10**	CO4	K6
	2.2 Pointers & Memory View: Pointers, memory access, pointer arrays, pointer to functions, Dynamic memory (malloc, calloc), Intro to command-line arguments. Write programs:Pointer-based array reversal, string operations, Dynamic array simulation (heap use case), Simulate memory blocks with structs and pointers.	**10**	CO3	K3, K4
	2.3 Structures, File I/O: Structures and nested structures, File operations: reading/writing, appending, record-based processing, Data structuring in flat files (simulate tables). Write programs: Management system with file storage. Mini Project	**10**	CO4	K6

	& Debugging: Management system (multi-user features)(e.g Leave Management System)			
Pedagogy:	Tutorials/Lab Assignments/Mini Project			
Texts:	1. Hanly, J. R., & Koffman, E. B. (2012). Problem solving and program design in C (7th ed.). Pearson Education. 2. Dromey, R. G. (1998). How to solve it by computer. Prentice-Hall of India (PHI).			
References/ Readings:	1. Srivastava, S. K. (2004). C in depth. BPB Publications. 2. Deitel, H. M., & Deitel, P. J. (2001). C: How to program. Addison Wesley.Balagurusamy, E. (2004). Programming in ANSI C. Tata McGraw-Hill. 3. Kernighan, B. W., & Ritchie, D. (1990). C programming language. Prentice-Hall of India (PHI). 4. Gottfried, B. S. (1996). Schaum's outline of programming with C (2nd ed.). McGraw-Hill. 5. C Manual https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.pdf Accessed May 22, 2025.			

Title of the Course	Linux Lab
Course Code	CSA-5004
Number of Credits	2
Theory/Practical	Practical
Level	400
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil			
Course Objectives:	The objective is to introduce students to the Linux operating system environment and provide knowledge of basic Linux commands and shell scripting and system call API.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To apply basic Linux commands and shell utilities.		PSO1	
	CO 2. To analyze and manage the Linux file system structure, file permissions and ownership.		PSO1, PSO2	
	CO 3. To apply various advanced LINUX tools such as grep, SED and AWK		PSO1, PSO2	
	CO 4. To write shell script using LINUX OS.		PSO1, PSO2	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	1.1 LINUX Environment: Linux Installation and disk partitioning. Shell, Linux commands, Internal and External Commands, using the documentation/manual, users in Linux: user id, effective user id, use of commands su, sudo, id Basic commands: echo, who, whoami, date, cal, ls, passwd, history, shutdown. Input and output redirection operators (<,<<, >, >>)	**12**	CO1	K1, K2
	1.2 The Linux File System, File and Directory management: Structure of LINUX file system. Parent-child relationship. Concept of Home directory, current working directory and referring to home directory. Special Files: . and .. Absolute and relative pathnames. Use of PATH variable, Use of command: mkdir, rmdir, pwd, ls and cd. Use of file management commands: nano, touch, cat, cp, mv and rm. FIND command: Searching for a file using find, Finding List of files and directories. Concept of hard disk partitions, file system, Superblock and Inodes, General structure of Linux inode. use of stat command. Analysing the output of ls -l command. File type and permission. Use of chmod command. File ownership: Changing ownership using chown and chgrp commands. Modification and access times. Default file and directory permissions. Use of umask command. Concept of symbolic links. Hard and soft links. Use of ln command to create hard and soft links. Use of commands du, df, tar, zip, gzip, type, which.	**12**	CO2	K1. K2, K3
	1.3 Filters: File commands- sort, wc, uniq, comm, cmp, diff, pg, tail, head, less, and more, Cut and Paste command Shells' sequence of interpretation of a command; Connecting commands with pipes Regular expressions: grep & sed command	**6**	CO3	K2, K3
Module 2:	2.1 AWK script:Selection criteria and action- The BEGIN and END sections, Splitting a line into fields and using printf. Getline function and reading input from files.	**10**	CO3	K2, K3

	Writing output to file and pipes. Awk system variables. Using regular expressions. Relational and Boolean operations. Command line parameters and environment variables. Programming constructs: if, for, while.			
	2.2 Process Management: Concept of UNIX process. Role of init in process creation. Process ID and exit status of a process. Displaying process attributes using ps command, Killing processes, foreground and background processes. Use of commands job, fg, bg Package management: Installing & removing packages.	**4**	CO2, CO3	K1, K2
	2.3 Shell Script: Shell scripts and execution methods. The dot command, Interactive and Non Interactive execution. Use of export command, liases and command history. Shell variables, Special variables, Built-in shell parameters. Command line arguments. Escaping and quoting. Difference between single and double quotes. Command substitution, brace and tilde expansion, I/O using read and echo. Escape sequences, 'test' command, arithmetic expressions, operators, Control flow: For, If, While, Case. Shell functions, error handling, debugging.	**16**	CO4	K3, K6
Pedagogy:	Practical/ Tutorials/Assignments			
Texts:	Das, S. (2006). Unix concepts and applications. Tata McGraw-Hill.			
References/ Readings:	1. Glass, G., & Ables, K. (2003). Unix and shell programming. Pearson Education. 2. Free Software Foundation. ls (1). In Unix command manual. https://man7.org/index.html Accessed date May 22, 2025.			
Web Resources:	W3Schools. Bash scripting tutorial. W3Schools. from https://www.w3schools.com/bash/, Accessed date May 22, 2025,			

Title of the Course	Object Oriented Technology Lab
Course Code	CSA-5005
Number of Credits	2
Theory/Practical	Practical
Level	400
Effective from AY	AY 2025-26
New Course	No
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	To enable students to practically apply object-oriented programming concepts by developing software solutions using core OOP constructs, advanced features, and multithreading. The course also aims to develop skills in software modeling through UML diagrams and culminates in the creation of a mini-project demonstrating object-oriented analysis and design principles.	
Course Outcomes:		**Mapped to PSO**
	CO 1. Demonstrate the setup and use of development environments, version control and visualization tools for Object Oriented	PSO1, PSO2, PSO3
	CO 2. Implement Object-Oriented constructs such as classes, constructors, method overloading and object referencing.	PSO1, PSO2, PSO3
	CO 3. Develop programs incorporating inheritance, polymorphism and access modifiers	PSO1, PSO2, PSO3

	CO 4. Apply advanced Object-Oriented programming concepts including exception handling, cloning, generics, collections and assertions.		PSO1, PSO2, PSO3	
	CO 5. Implement persistence and input/output operations for object storage and retrieval.		PSO1, PSO2, PSO3	
	CO 6. Develop concurrent applications using multithreading concepts and Create UML diagrams to model software design and develop a complete mini-project based on OOAD principles.		PSO1, PSO2, PSO3	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Understanding OO computation through visualization: Using BlueJ or Alice. Creation of a simple class and use of data types, declaring attributes and operations followed by discussion. Use of two classes and linking them. Setting up an environment such as workspace and Github account. Understanding OO Program compilation, execution, debugging mechanism. Understanding and setting up runtime environment such as Java Runtime Environment (JRE), source code to bytecode compilation, bytecode execution and interoperability, write once, run everywhere mechanism Declaring class and executing OO program using IDE like Eclipse: A simple class is to be declared with few attributes and methods. Execute the program using the main() method. No console input but only passing of values and watching the result. Method overloading. Ex Bank Account class may be used. Understanding construction mechanism: role of constructors, constructor overloading, referring to the object itself such as "this"	**10**	CO1	K2, K3
	1.2 Object referencing: declaring another object as an attribute. Passing of an object as a parameter in methods. Side effects of Object referencing. Constructors for such a class. Understanding the message passing mechanism. Adding more classes and references and understanding the object	**10**	CO2	K3

	oriented computation model. Use of static attributes and methods. Use of enums. Implementation level Inheritance: defining base class and derived class. Base class to have few methods and operations, derived class to have additional attributes and methods. Execution of such a program with instances of Base class and instances of derived class. Assigning object of derived class to reference of base class and invoking methods. Method overloading between base and derived class. Use of public, private and protected access modifiers. Type/Interface Inheritance: Method overriding, abstract methods, abstract/interface class. Understanding class as a type.			
	1.3 Multiple inheritance: multiple implementation inheritance and multiple interface inheritance. Understanding the differences. library utilities/packages: use of math and other libraries.use of documentation tools such as Javadoc. Generating Javadoc for your own code. Exception handling: manage runtime errors effectively by using mechanisms like try-catch block, finally block, throwing Exceptions, Custom Exception handling. Object cloning: shallow and deep cloning.	**10**	CO3	K3, K4
Module 2:	2.1 Generics and Collections frameworks, assertions, documentation: use of collection utilities such as List, Queue, set, Map etc. Implementing one to many and many to many relationships, creating classes, interfaces, and methods where the type of the data is specified as a parameter. Covariance and contravariance. Use of assertions, Javadoc Reflection, Persistence and I/O: storing and retrieving objects, performing I/O through console, files, RTTI.	**7**	CO5	K3
	2.2 Multithreading: Concurrency through threads, use of Thread class, Runnable interface, thread lifecycle. Example simulation of game or traffic or elevator using	**8**	CO6	K3

	threads. UML: use of UML diagramming tool. Creation of class diagrams, sequence diagram, use case diagram, activity diagram.			
	2.3 Project 1. A mini project for a business application for which UML based analysis and design model is to be created. 2. An implementation of the design of mini project using OOP (Java/Python etc)	**15**	CO6	K4,K6
Pedagogy:	Practical/ Tutorials/Assignments/Mini-Project			
Texts:	1. Barnes, D. J., & Kölling, M. (2017). Objects first with Java: A practical introduction using BlueJ (6th ed.). Pearson. 2. Mughal, K. (2012). A programmer's guide to Java programming certification (3rd ed.). Pearson Education. 3. Fowler, M. (2018). UML distilled: A brief guide to the standard object modeling language (3rd ed.). Addison-Wesley Object Technology Series.			
References/ Readings:	Jeff Langr, Agile Java: Crafting Code with Test-Driven Development (Robert C. Martin Series) Paperback, 2005			
Web Resources:	Oracle. Javadoc tool for Windows. Oracle. https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.htm Accessed May 22, 2025.			

Discipline Specific Elective Courses

Title of the Course	Mathematics for Computer Science - I
Course Code	CSA-5201
Number of Credits	2
Theory/Practical	Theory
Level	400
Effective from AY	AY 2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil			
Course Objectives:	To understand fundamental concepts and tools in linear algebra with emphasis on their applications to computer science in particular data science/machine learning.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To apply linear algebra techniques to data analysis problems.		PSO1, PSO2	
	CO 2. To solve problems using calculus concepts like differentiation, integration		PSO1, PSO2	
	CO 3. To analyze and interpret data using probability distributions and statistical methods.		PSO1, PSO2	
	CO 4. To evaluate and apply numerical methods and optimization techniques.		PSO1, PSO2, PSO6	
Content:		**No of**	**Mapped**	**Cognitive**

		hours	to CO	Level
Module 1:	1.1 Linear Algebra: Scalars, Vectors, Matrices and Tensors -Multiplying Matrices and Vectors - Identity and Inverse Matrices -Linear Dependence and Span -Norms - Special Kinds of Matrices and Vectors - Eigen decomposition -Singular Value Decomposition -The Trace Operator - The Determinant - Example: Principal Components Analysis. Numerical Computation: Overflow and Underflow -Poor Conditioning - Gradient-Based, Optimization - Constrained Optimization -Example: Linear Least Squares.	**8**	CO1, CO4	K2, K3, K5
	1.2 Calculus: Overview of Differential and Integral Calculus, Partial Derivatives Product and chain rule-Taylor's series, infinite series summation/integration concepts Fundamental and mean value-theorems of integral calculus, evaluation of definite and improper integrals-Beta and Gamma functions, Functions of multiple variables, limit, continuity, partial derivatives-Basics of ordinary and partial differential equations - Applications of Calculus.	**7**	CO2	K2, K3
Module 2:	2.1 Probability, Statistics, and Information Theory Why Probability? -Random Variables -Probability Distributions - Marginal Probability - Bayesian networks. Independence -Expectation, Variance and Covariance -Common Probability Distributions - Useful Properties of Common Functions.	**7**	CO3	K3, K4
	2.2 Statistics: Data summaries and descriptive statistics, central tendency, variance, covariance, correlation-Basic. Probability distribution functions: uniform, normal, binomial, chisquare, Student's t-distribution, central limit theorem-Sampling, measurement, error, random number generation-Hypothesis testing, A/B testing, confidence intervals, p-values, ANOVA, t-test-Linear regression, regularization.	**8**	CO3	K3, K4
Pedagogy:	Lectures/ Tutorials/Assignments/ Flipped classroom			
Texts:	1. Field, A., Miles, J., & Field, Z. (latest edition). Discovering statistics using R. SAGE.			

	2. Inouye, O. M. (latest edition). Introductory calculus for infants. Publisher. 3. Witte, R. S., & Witte, J. S. (2017). Statistics (11th ed.). Wiley. 4. Strang, G. (2016). Introduction to linear algebra (5th ed.). Wellesley-Cambridge Press.
References/ Readings:	Strang, G. (2016). Introduction to linear algebra (5th ed.). Wellesley-Cambridge Press.
Web Resources:	https://math.mit.edu/~gs/dela/ Accessed date May 22, 2025.

Title of the Course	Fundamentals of Data Science
Course Code	CSA-5202
Number of Credits	2
Theory/Practical	Theory
Level	400
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	To equip students with foundational knowledge and skills in data handling, analysis, and visualization.	
Course Outcomes:		**Mapped to PSO**
	CO 1. Identify and describe the methods and techniques commonly used in data science	PSO1, PSO2
	CO 2. Demonstrate proficiency with the methods and techniques for obtaining, organizing, exploring, and analyzing data.	PSO1, PSO5, PSO6
	CO 3. Recognize how data analysis, inferential statistics, modeling, machine learning, and statistical computing can be utilized in an integrated capacity.	PSO1, PSO2, PSO5
	CO 4. Demonstrate the ability to clean and prepare data for analysis and assemble data from a variety of sources.	PSO2, PSO5

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	Introduction: What is Data Science, Case for Data Science, Data Science classification Data Science process: Prior Knowledge, Data preparation, Modeling, Application Data Exploration: Datasets, types of Data, Descriptive Statistics, Data Visualization: Univariate, multivariate, higher dimensional data visualization, Roadmap for Data exploration.	**15**	CO1, CO2, CO4	K1, K2, K3, K4
Module 2:	Linear regression, Model Evaluation: R-squared, RMSE, MAE. Logistic regression-Support vector machine kernel- Model selection and feature selection-Ensemble methods: Random Forest, Boosting, Bagging. Decision trees, Rule induction, K-nearest Neighbors, Naive Bayesian. Evaluation Metrics: Accuracy, Precision, Recall, F1-Score, K-Means Clustering: Centroid-based clustering and optimization. Hierarchical Clustering: Agglomerative and divisive approaches. DBSCAN: Density-based clustering for arbitrary shapes. Association Analysis: Mining Association Rules, Apriori Algorithm, Frequent pattern growth algorithm. Evaluation Metrics: Silhouette Score, Elbow Method, and Davies-Bouldin Index. Comparing Models: Performance trade-offs for classification, regression, and clustering. Model complexity vs. interpretability. Computational efficiency vs. prediction accuracy. Algorithm Selection: Criteria for choosing the right algorithm based on problem type and data characteristics. Cross-Validation: Model tuning and performance evaluation.	**15**	CO1 CO2 CO3	K1, K2, K3, K4
Pedagogy:	Lectures/ Tutorials/Assignments/ Flipped classroom			
Texts:	Khotu, V., & Deshpande, B. (2014). “Data science: Concept and practice” (2nd ed.). Morgan Kaufmann Publishers.			

References/ Readings:	Blum, A., Hopcroft, J., & Kannan, R. (2018, January 4). Foundations of data science.
Web Resources:	W3Schools. Data science introduction. W3Schools. https://www.w3schools.com/datascience/ds_introduction (Accessed May 22, 2025)

Title of the Course	Operations Research
Course Code	CSA-5203
Number of Credits	4
Theory/Practical	Theory
Level	400
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	To provide students the theoretical knowledge to effectively formulate linear programming problems and optimization techniques and approaches.	
Course Outcomes:		**Mapped to PSO**
	CO 1. To understand applications of Operation Research.	PSO1
	CO 2. To formulate a Linear programming problem.	PSO1, PSO2
	CO 3. To recognize competitive forces in the marketplace and develop appropriate reactions based on existing constraints and resources.	PSO2, PSO6
	CO 4. To understand primal dual relationships.	PSO1
	CO 5. To solve specialized linear programming problems like the transportation and assignment problems.	PSO1, PSO6

	CO 6. To understand the basic terminology and concepts related to networks.		PSO 1, PSO 6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Introduction to Operations Research: Introduction-Mathematical models of Operation Research - Scope and applications of Operation Research - Phases of Operation Research study - Characteristics of Operation Research - Limitations of Operation Research.	**4**	CO1	K1, K2, K3
	1.2 Linear Programming: Introduction –Properties of Linear Programming-Basic assumptions-Mathematical formulation of Linear Programming-Limitations or constraints-Methods for the solution of LP Problem- Linearly independent / dependent vectors, Graphical analysis of LP-Graphical LP Maximization problem-Graphical LP Minimization problem.	**16**	CO2	K2, K3, K6
	1.3 Linear Programming Models: Simplex Method-Basics of Simplex Method - Formulating the Simplex Method-Simplex Method with two variables - Simplex Method with more than two variables - Two Phase Method; M-Charnes Method, Special cases in LPP.	**14**	CO2, CO3	K2, K3
Module 2:	2.1 Dual Linear Programming: Introduction- Primal and Dual problem - Dual problem properties-Solution techniques of Dual problem - Dual Simplex method-Relations between direct and dual problemEconomic interpretation of Duality. Sensitivity analysis: Changes in cost and resource vector	**11**	CO2, CO3	K2, K4
	2.2 Transportation and Assignment Models: Introduction: Transportation problem - Balanced - Unbalanced - Methods of basic feasible solution Optimal solution-MODI method. Assignment problem-Hungarian Method.	**8**	CO5	K2, K5

	2.3 Network Analysis: Basic concepts-Construction of Network-Rules and precautions-CPM and PERT Networks Obtaining critical path. Probability and cost consideration. Advantages of Network.	**7**	CO 6	K2, K3, K4
Pedagogy:	Lectures/ Tutorials/Hands-on assignments/			
Texts:	Hamdy A. Taha: Operations Research-An Introduction, Prentice Hall, 8th Edition, 2008.			
References/ Readings:	1. Gupta, P. K., & Hira, D. S. (2022). Introduction to Operations Research. S. Chand Publishing. 2. J K Sharma (2007), Operations Research Theory & Applications, 3e, Macmillan India Ltd. 3. Maurice Solient, Arthur Yaspen, Lawrence Fridman, OR methods and Problems (2003), New Age International Edition. 4. P. SankaraIyer, (2008), Operations Research, Tata McGraw-Hill. 5. Philips, D. T. (2007). Operations research: Principles and practice. John Wiley & Sons, Incorporated. 6. S.D. Sharma (2000). Operations Research. Nath & Co., Meerut.			

SEMESTER II

Discipline Specific Core Courses

Title of the Course	Data Structures and Algorithms
Course Code	CSA-5006
Number of Credits	2
Theory/Practical	Theory
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5003	
Course Objectives:	The aim of the course is to emphasize the importance of data structures in implementing efficient algorithms. It provides an exposure to various algorithm design techniques and an introduction to algorithm analysis.	
Course Outcomes:		**Mapped to PSO**
	CO 1. Implement common data structures such as lists, stacks, queues, graphs, and binary trees for solving programming problems.	PSO1, PSO2
	CO 2. Identify and use appropriate data structures in the context of a solution to a given problem.	PSO1, PSO2

	CO 3. To analyze the complexity of a given algorithm.		PSO1, PSO2	
	CO 4. To create a project using an appropriate data structure.		PSO1, PSO2, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Algorithm Representation: - Pseudocode and flowcharts, Three level Approach, Abstract Data Types (ADTs), Basic Linear Data Structures (LinkedList, Stack, Queue)	**4**	CO1, CO2	K1, K2, K3
	1.2 Algorithm Analysis: Analysis of Algorithms, Algorithm Complexity: Space and Time, Cases of Complexity: Best, Worst and Average, Growth of Functions: Asymptotic Notation.	**3**	CO1, CO2, CO3	K3, K4
	1.3 Advanced Linear Data Structures Variants of Linked List and its applications (e.g., Polynomial addition, Sparse matrices) Applications of stacks (e.g., Infix-to-Postfix conversion, Evaluating Postfix Expressions, Bracket Matching) Variants of Queue and Applications	**4**	CO1, CO2	K3, K4
	1.4 Divide & Conquer Strategy: Algorithms based on the Divide and Conquer Strategy: Sorting Algorithms (QuickSort, MergeSort), Binary Search.	**4**	CO1, CO2, CO3, CO4	K3, K4
Module 2:	2.1 Nonlinear Data Structures: Trees: Binary Search Trees, AVL Trees, B-trees & variants. Tree Traversal Algorithms: Heaps and their applications (e.g., implementation of Priority Queue), Graph: Adjacency Matrix and Adjacency List Representations, Graph Traversal Algorithms: Breadth First Search and Depth First Search.	**10**	CO2, CO3, CO4	K3, K4
	2.2 Greedy Algorithms: Huffman Coding Algorithm, Minimum Cost Spanning Tree (Prim's, Kruskal's), Single Source Shortest Path (Dijkstra's).	**2**	CO2, CO3,	K3, K4, K6

			CO4	
	2.3 Dynamic Programming: Coin Change Problem, Longest Common Subsequence, All-pair shortest Path (floyd-warshall).	**2**	CO2, CO3, CO4	K3, K4, K6
Pedagogy:	● Lectures/Tutorials/Assignments/Quizzes ● Each data structure should be explained along with the implementation of its ADT, its applications, and its complexity			
Texts:	1. Horowitz, E., Sahni, S., & Anderson-Freed, S. "Fundamentals of data structures in C" (Latest ed.). W.H. Freeman & Co. 2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. "Introduction to algorithms" (Latest ed.). The MIT Press.			
References/ Readings:	1. Weiss, M. A. Data structures and algorithm analysis in C (Latest ed.). Pearson Education India. 2. Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. Algorithms. McGraw-Hill. 3. Hanly, J. R., & Koffman, E. B. (2012). Problem solving and program design in C (7th ed.). Pearson Education. 4. Dromey, R. G. How to solve it by computer (Latest ed.). PHI Learning.			
Web Resources:	1. W3Schools. Introduction to Data Structures and Algorithms. W3Schools. https://www.w3schools.com/dsa/dsa_intro.php Accessed on May 28, 2025. 2. GeeksforGeeks. (2025, May 27). DSA Tutorial – Learn Data Structures and Algorithms. GeeksforGeeks. https://www.geeksforgeeks.org/dsa-tutorial-learn-data-structures-and-algorithms/ Accessed on May 28, 2025.			

Title of the Course	Database Management Systems
Course Code	CSA-5007
Number of Credits	3
Theory/Practical	Theory
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5000, CSA-5001	
Course Objectives:	This course will enable the learner to understand the different issues involved in the design and implementation of a database system and provide both theoretical knowledge and practical skills required in the creation and use of a Relational Database Management System.	
Course Outcomes:		**Mapped to PSO**
	CO 1. Understand and evaluate the role of a DBMS in information Technology applications in Organizations.	PSO1
	CO 2. Recognise and use logical design methods and tools required in the design of DB applications.	PSO1, PSO2
	CO 3. Understand the relational database design principles and implement a database Solution to an IT Platform.	PSO1, PSO2, PSO6

	CO 4. Understand the basics of SQL and construct queries using SQL. And develop sophisticated queries to extract information from databases.		PSO1, PSO2, PSO6	
	CO 5. Use embedded SQL queries in a host-level language. Understand how the DBMS manages and recovers from concurrent and multiple transactions.		PSO1, PSO2, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Basic concepts: Database & Database Users, Characteristics of the Database Approach, Database Systems, Concepts & Architecture, Data Models(RDBMS, Legacy systems, Object Oriented, NoSQL), Schemes & Instances DBMS Architecture of Data Independence, Database languages & Interfaces.	**3**	CO1	K1, K2
	1.2 Data Modelling using the Entity–Relationship approach	**4**	CO1, CO2	K1, K2
	1.3 Relational Model, Languages & Systems: Relational Data Model & Relational Algebra Relational Model, Concepts Relational Model Constraints, Relational Algebra/Relational Calculus	**5**	CO2, CO3	K3
	1.4 SQL-A Relational Database Language Data: SQL - DDL, DML. Views & Queries in SQL. Specifying Constraints & Indexes in SQL. Nested Subqueries, correlated Subqueries.	**3**	CO4	K3, K4
Module 2:	2.1 Advanced SQL: Embedded SQL, Dynamic SQL, Triggers, and Stored Procedures.	**8**	CO4	K6
	2.2 Relational Database Design Function Dependencies & Normalization for Relational Database Functional Dependencies Normal forms based on primary keys (1NF, 2NF, 3NF, BCNF) Covers of Functional Dependencies, Canonical covers. Lossless join and Dependency preserving decomposition algorithms.	**7**	CO3, CO4	K4, K6
Module 3:	Transactions and Recovery Techniques: Concept of a transaction, Recovery concepts, Recovery Techniques.	**15**	CO5	K2, K3, K4

	Concurrency Control: Serializability, Locking Techniques, Time stamp ordering, Granularity of Data items			
Pedagogy:	Hands-on assignments / tutorials / peer-teaching / troubleshooting/Flipped learning/Assignments			
Texts:	1. Korth, H. F., & Silberschatz, A. Database system concepts (Latest ed.). McGraw-Hill Education. 2. Elmasri, R., & Navathe, S. B. Fundamentals of database systems (Latest ed.). Addison-Wesley. 3. Ramakrishnan, R., & Gehrke, J. Database management systems (Latest ed.). McGraw-Hill.			
References/ Readings:	1. Desai, B. An introduction to database concepts (Latest ed.). Galgotia Publications. 2. Coronel, R., & Morris, S. Database systems: Design, implementation, and management (Latest ed.). Cengage Learning. 3. Date, C. J. An introduction to database systems (Latest ed.). Publication House.			
Web Resources:	GeeksforGeeks. Introduction of DBMS (Database Management System) |. https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/ Accessed on 28th May 2025.			

Title of the Course	Web Development
Course Code	CSA-5008
Number of Credits	1
Theory/Practical	Theory
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5002			
Course Objectives:	To introduce students to various Website Development Technologies.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To understand the use of web technology and its purpose.		PSO1, PSO 2	
	CO 2. To understand and develop applications using popular technologies in website development.		PSO2, PSO3	
	CO 3. To understand the architecture of web applications and the design decisions		PSO1, PSO3	
	CO 4. To develop applications using recent architecture and server-side programming.		PSO3	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	1.1 Introduction: Evolution of Internet & World Wide Web, Client-Server Architecture, Revisit HTML & CSS. Enhancing HTML & CSS: HTML5, CSS3. Front-end Design: Good Design Rubrics, Separation of concerns for HTML & CSS; structure vs visual representation, HTML DOM, CSS Features: Box Model, pseudo -classes & -elements, CSS animation, Adaptive & responsive design, viewport & media queries, mobile-first design, Introduction to a design library and/or & framework (e.g. Bootstrap).	**4**	CO1, CO2	K1, K2
	1.2 Client-side Scripting: Dynamic web pages, JavaScript, programming features, JavaScript events & functions, Manipulating DOM, Introduction to a JavaScript library and frameworks (e.g. Query, ReactJS)	**4**	CO1, CO2	K2, K3, K6
	1.3 HTTP & Middle-ware: HTTP, Request & Response, methods & error code, headers, URL encoding & decoding, XML, data & XPath, JSON. Server-side Programming: Server instance, Request handling & response creation, Session management & application data, Database connectivity, Introduction to a Server-side library and/or template engine and/or framework (e.g. PHP - Laravel; JSP - Spring).	**4**	CO3, CO4	K2, K3, K6
	1.4 Advanced Web Development: Model-View-Controller (MVC) & Model-View-ViewModel and others, Web service architecture and micro-services, REST calls, Asynchronous JavaScript and XML (AJAX), Introduction to Web stacks, JAM stack & full stack development.	**3**	CO3, CO4	K2, K3, K6
Pedagogy:	Hands-on assignments/tutorials / peer-teaching / flip classroom/ presentations			
Texts:	1. Sebesta, R. W. Programming the world wide web (Latest ed.). Pearson Education. 2. Holzner, S. HTML 5 black book (Latest ed.). 3. Zammetti, F. W. Modern full-stack development (Latest ed.). Apress.			

References/ Readings:	Dabit, N. Full stack serverless (Latest ed.). O'Reilly Media.
Web Resources:	1. W3Schools Online Web Tutorials. https://www.w3schools.com/ Accessed on 22nd May 2025 2. TutorialsPoint: Learn programming, web development. https://www.tutorialspoint.com/ Accessed on 22nd May 2025. 3. GeeksForGeeks - Web Design Tutorial: https://www.geeksforgeeks.org / Accessed on 22nd May 2025

Title of the Course	Machine Learning
Course Code	CSA-5009
Number of Credits	2
Theory/Practical	Theory
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5003, CSA-5005			
Course Objectives:	To introduce to students with an in-depth introduction to three, main areas of Machine Learning: supervised and unsupervised and reinforcement learning.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To understand the steps involved in learning from data and building model.		PSO1, PSO2	
	CO 2. To understand the working of algorithms.		PSO1, PSO2, PSO5	
	CO 3. To perform the evaluation of learning algorithms and model selection.		PSO1, PSO2, PSO5, PSO6	
	CO 4. To understand the importance of evaluation metric.		PSO1, PSO2, PSO5, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	1.1 Introduction:- well posed learning problem – designing a learning system-perspectives and issues in machine learning.	**3**	CO1	K1, K2
	1.2 Concept learning – concept learning task –notation –inductive learning hypothesis-concept learning as search- version space and candidate elimination algorithm-decision tree –random forest.	**4**	CO2	K2, K3
	1.3 Linear regression - logistic regression-Support vector machine kernel-Model selection and feature selection-Ensemble methods: Bagging, boosting. Evaluating and debugging learning algorithms. Evaluation metric: Various evaluation metric: F-score, accuracy.	**5**	CO2, CO3	K3, K4
	1.4 Continuous Latent Variables-Revision of Principal Component Analysis - Applications of PCA - PCA for high-dimensional data.	**3**	CO2, CO3, CO4	K3, K4
Module 2:	2.1 Neural Networks -Feed-forward Network Functions –perceptron - Weight-space symmetries -Network Training - Parameter optimization -Local quadratic approximation - Use of gradient information - Gradient descent optimization - Error Backpropagation - Evaluation of error-function derivatives - A simple example - Efficiency of backpropagation .	**5**	CO2, CO3	K3, K4
	2.2 Probabilistic model – The normal distribution and its geometric, interpretation-probabilistic models for categorical data -using naïve , Bayes model for classification, training a naïve Bayes model - discriminative learning by optimizing conditional likelihood - probability models with hidden variables: Expectation-Maximization, Gaussian mixture model.	**4**	CO2, CO3	K3, K4
	2.3 Sequential Data - Markov Models - Hidden Markov Models - Maximum likelihood for the HMM -The forward-backward algorithm - The sum-product algorithm for the HMM -Scaling factors - The Viterbi	**6**	CO2, CO3	K3, K4

	algorithm. Reinforcement learning – Introduction- learning task-Q learing-non deterministic rewards and actions-temporal difference learning.			
Pedagogy:	Lectures/ tutorials/assignments			
Texts:	1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. Springer. 2. Alpaydin, E. Introduction to machine learning (Latest ed.). MIT Press. 3. Duda, R. O., Hart, P. E., & Stork, D. G. Pattern classification (Latest ed.). Wiley.			
References/ Readings:	1. Flach, P. Machine learning (Latest ed.). Cambridge University Press. 2. Bishop, C. M. Pattern recognition and machine learning (Latest ed.). Springer. 3. Goodfellow, I., Bengio, Y., & Courville, A. Deep learning (Latest ed.). MIT Press. 4. Michele, T. Machine learning (Latest ed.). McGraw-Hill.			
Web Resources:	1. GeeksforGeeks. (2025, May 3). *Machine Learning Tutorial*. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/ Accessed on 28th May 2025. 2. Google Developers. (2025). Machine Learning Crash Course. Google. https://developers.google.com/machine-learning/crash-course Accessed on 28th May 2025.			

Title of the Course	Data Structures and Algorithms Lab
Course Code	CSA-5010
Number of Credits	2
Theory/Practical	Practical
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5003			
Course Objectives:	To develop skills to design and implement linear and nonlinear data structures and to identify the most appropriate data structure for solving a real world problem.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To implement the basic data structures for solving programming problems.		PSO1, PSO2	
	CO 2. To implement the non-linear data structure for solving real world problems		PSO1, PSO2	
	CO 3. To implement dynamic programming and sorting techniques for solving real problems.		PSO1, PSO2	
	CO 4. To implement application and analyze the different data structure, which is specifically needed to solve real world problems.		PSO1, PSO2	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	1.1 Advanced Linear Data Structures: Infix-to-Postfix conversion, Evaluating Postfix Expressions, Bracket Matching.	**8**	CO1	K1, K2, K3
	1.2 Non-linear data structures: Binary Trees, Tree Traversal Algorithms Binary Search Trees, Heap, Priority Queue using Heap, Heap Sort. Graph implementation using Adjacency list and matrix, Graph Traversal Algorithms	**22**	CO2	K3, K4
Module 2:	Divide & Conquer Strategy: MergeSort, QuickSort, Binary Search Algorithm. Greedy Algorithms: Huffman Coding Algorithm, Prims' and Kruskal's Algorithm, Dijkstra's Algorithm Dynamic Programming: Coin Change Problem, Longest Common, Subsequence, Floyd-Warshall Algorithm. A Mini Project.	**30**	CO3, CO4	K4, K6
Pedagogy:	Programming assignments/ discussions/ self-review/ peer-review/ testing of code/ debugging of code/ projects/Flipped Learning.			
Texts:	1. Horowitz, E., Sahni, S., & Anderson-Freed, S. Fundamentals of data structures in C (Latest ed.). W. H. Freeman & Co. 2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. Introduction to algorithms (Latest ed.). The MIT Press.			
References/ Readings:	1. Weiss, M. A. Data structures and algorithm analysis in C (Latest ed.). Pearson Education India. 2. Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2017). Algorithms. McGraw-Hill Education.			
Web Resources:	1. W3Schools. Introduction to Data Structures and Algorithms. W3Schools. https://www.w3schools.com/dsa/dsa_intro.php Accessed on May 28, 2025. 2. GeeksforGeeks. (2025, May 27). DSA Tutorial – Learn Data Structures and Algorithms. GeeksforGeeks. https://www.geeksforgeeks.org/dsa-tutorial-learn-data-structures-and-algorithms/ Accessed on May 28, 2025.			

Title of the Course	Database Management Systems Lab
Course Code	CSA-5011
Number of Credits	2
Theory/Practical	Practical
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5001			
Course Objectives:	This course aims at introducing the students to develop a skill set to design and implement a realistic application, representative of a typical real world software system.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To design and implement a database schema for a given problem domain.		PSO1, PSO2	
	CO 2. To create and maintain tables using SQL.		PSO1, PSO2	
	CO 3. To use Transaction Control Language, Creating and Using User Defined Data Types. Writing Triggers & Stored Procedure.		PSO1, PSO2	
	CO 4. To create reports and Application development using PL/SQL & front end tools		PSO1, PSO2, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	1.1 Installation of DBMS Softwares Data Definition Language(DDL) Statements • Creating a Database. • Creating a table, with or without constraints. • Understanding Data types. • Altering the structure of the table like adding attributes at a later stage, modifying size of attributes or adding constraints to attributes. • Removing the table created, i.e Drop table in SQL. • Creating Sequence (Auto increment field)	**6**	CO1, CO2	K2, K3
	1.2 Query in Data Dictionary • To view the structure of the table created by the user. • To view user information. • To view integrity constraints. • Altering Session Parameters	**2**	CO1, CO2	K2, K3, K4
	1.3 Data Manipulation Language(DML) Statements • Inserting Data into the table. • Updating Data into the table. • Deleting Data from the table.	**4**	CO1, CO2	K2, K3, K4
	1.4 Simple SQL statements • Displaying all the attributes and tuples from the table. • Displaying selected attributes/tuples from the table. • Using Logical and comparison operators. • String manipulation • Date Comparisons	**18**	CO1, CO2	K2, K3, K4

	Complex SQL Statements • Using aggregate functions (using Group by and having clauses). • Sorting Data. • Creating SQL Aliases and Views. • Joins and Nested queries. • Correlated subquery • Derived tables • Given a complex table structure, display records from tables.			
Module 2:	2.1 Transaction Control Language(TCL) statements • Transactions could be made permanent in memory • To rollback the transaction.	**4**	CO3	K4, K6
	2.2 Embedded SQL statements • Loops/ if else statements • Creating Triggers/Procedures/packages • ArrayList and Cursor.PL/SQL Strings • PL/SQL Object Oriented • Exceptions • No SQL	**20**	CO3	K4, K6
	2.3 Project • The analysis of project • Design (ER diagram and normalized tables) and Implementation of a real-life project of students' choice. • The project report that they submit consists of (i) Feasibility study (ii) ER Diagrams (iii) Tables normalized in an appropriate normal form with integrity and domain constraints noted. (iv) User Interface Design -Form and Report	**6**	CO4	K4, K6

	design, including triggers that may need to be written (v) User Manual Peer reviews of ERDs are held in the class.			
Pedagogy:	Hands-on assignments / tutorials / peer-teaching / troubleshooting			
Texts:	1. Korth, H. F., & Silberschatz, A. Database system concepts (Latest ed.). McGraw-Hill Education. 2. Elmasri, R., & Navathe, S. B. Fundamentals of database systems (Latest ed.). Addison-Wesley. 3. Ramakrishnan, R., & Gehrke, J. Database management systems (Latest ed.). McGraw-Hill Education.			
References/ Readings:	1. Desai, B. An introduction to database concepts (Latest ed.). Galgotia Publications. 2. Coronel, R., & Morris, S. Database systems: Design, implementation, and management (Latest ed.). Cengage Learning. 3. Date, C. J. An introduction to database systems (Latest ed.). Publication House.			
Web Resources:	GeeksforGeeks. Introduction of DBMS (Database Management System) |. https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/ Accessed on 28th May 2025.			

Title of the Course	Web Development Lab
Course Code	CSA-5012
Number of Credits	2
Theory/Practical	Practical
Level	500
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CS-5001, CSA-5002, CSA-5005			
Course Objectives:	This course will focus on the practical use and aspects of the different website development technologies			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To understand and create complete websites		PSO1, PSO2	
	CO 2. To decide on the appropriate web technology and its purpose.		PSO1, PSO2	
	CO 3. To understand the architecture of web applications and the design decisions.		PSO1, PSO2	
	CO 4. To develop web applications and host it locally.		PSO1, PSO2, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	Web Design Assignments Suggested Sample (non-exhaustive) Assignments: - • Create a website on a topic given by the instructor. Evaluating the website with rubrics for good web design. • Build a website using HTML & CSS by looking at a screenshot/picture of a website component given by the instructor. • Websites built with tables, forms, images, iframes, etc. • A website for each of design strategies (fixed, adaptive, responsive, fluid, mobile-first, etc.). • Assignments using css pseudo-classes & -elements; grid & flex design; understanding the CSS box model & working with the browser developer tools; CSS transformations, transitions & animations • Assignment to create a website built with Bootstrap based on a topic given by the instructor	**15**	CO1, CO2	K2, K3, K6
Module 2:	Client-side Scripting Assignments Suggested Sample (non-exhaustive) Assignments: - • An assignment for understanding the programming aspects of JavaScript and working with the browser developer tools. The use of the newer features of JavaScript (after ECMA 4) is encouraged. • An assignment working with regular expressions. A search and filter utility can be built. • Assignments for form data processing and validation and use of HTML5 form elements. A web page with form and validated data could be put in a table. The code could be written using table DOM methods and/or HTML DOM methods and/or XML DOM methods. • Assignments using various events (mouse, keyboard, etc. events for the form	**15**	CO1, CO2, CO3	K2, K3, K6

	elements, drag-and-drop, window, browser, etc.). • A web component built using HTML, CSS & JavaScript based on an existing Bootstrap component (e.g. Accordion) • Assignment with the use of a JavaScript library (JQuery, AngularJS, ReactJS, etc.)			
Module 3:	3.1 Developing a Game with HTML, CSS & JavaScript. The game should have at least 500 lines of (HTML+Javascript) code and make use of various mouse/keyboard events.	**4**	CO2, CO3	K2, K3, K6
	3.2 Server-side Programming Assignments Suggested Sample (non-exhaustive) Assignments: - • Assignments to work with HTTP headers for passing data and meta-data, cookies, localStorage • Assignments to handle data from web forms; handling the request and response payload • Assignment to manage web sessions • Assignment to develop a CRUD functionality by connecting to a database; AJAX calls	**12**	CO2, CO3	K2, K3, K6
	3.3 Full Stack Web Development: Develop a CRUD application with MEAN/MERN stack	**2**	CO3, CO4	K2, K3, K6
	3.4 Mini-project, Ideally done in a group. It should include design and implementation of a web application. Project implementation should mandatorily be built using a templating engine or programming framework (clientside and/or server-side). Project should also use a design framework (e.g. Bootstrap). Conduct and progress of the project could follow industry practices (e.g. git, scrum etc.).	**12**	CO3, CO4	K2, K3, K6
Pedagogy:	Hands-on assignments / tutorials / peer-teaching / projects			
Texts:	1. Sebesta, R. W. Programming the world wide web (Latest ed.). Pearson Education.			

	2. Holzner, S. HTML 5 black book (Latest ed.). Dreamtech Press.
References/ Readings:	1. Zammetti, F. W. (2020). Modern full-stack development: Using TypeScript, React, Node.js, Webpack, and Docker. Apress. 2. Dabit, N. (2020). Full stack serverless: Modern application development with React, AWS, and GraphQL. O'Reilly Media.
Web Resources:	1. W3Schools Online Web Tutorials. https://www.w3schools.com/ Accessed on 22nd May 2025 2. TutorialsPoint: Learn programming, web development. https://www.tutorialspoint.com/ Accessed on 22nd May 2025. 3. GeeksForGeeks - Web Design Tutorial: https://www.geeksforgeeks.org / Accessed on 22nd May 2025

Title of the Course	Machine Learning Lab
Course Code	CSA-5013
Number of Credits	2
Theory/Practical	Practical
Level	500
Effective from AY	AY 2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	CSA-5003, CSA-5005			
Course Objectives:	The objective is to learn to build various machine learning models by doing a set of assignments and mini projects.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To develop models using basic supervised and unsupervised machine learning algorithms.		PSO1, PSO2, PSO3	
	CO 2. To understand the use of data and evaluation metric and perform detailed error analysis		PSO1, PSO2, PSO3	
	CO 3. To understand and create neural network models and reinforcement learning.		PSO1, PSO2, PSO3	
	CO 4. To understand, create, and evaluate models implemented using deep learning.		PSO1, PSO2, PSO3	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**

Module 1:	1.1 Introduction to python libraries for machine learning - scikit learn, tensor flow, keras, pytorch, pandas, matplotlib, seaborn, numpy and other relevant libraries.	**5**	CO1	K2, K3
	1.2 Four branches of machine learning-supervised, unsupervised, self supervised, reinforcement, Evaluating machine learning models, Data pre-processing, feature engineering and feature learning, overfitting and underfitting - Numerical Programming fundamentals-finding nearest neighbours via Euclidean distance-splitting data sets into training and testing.	**10**	CO2, CO3	K2, K3
	1.3 Regression, cross validation and regularization-polynomial regression -model selection on a fixed validation set -Polynomial Regression - Model Selection with Cross-Validation-Polynomial Regression with L2 Regularization - Model Selection with Cross-Validation-Comparison of methods on the test set. Evaluating Binary Classifiers and Implementing Logistic RegressionBinary Classifier for movies reviews-classifying newswires-predicting house prices - Computing the Loss for Logistic Regression without Numerical Issues	**15**	CO1, CO2	K3, K4
Module 2:	2.1 Neural Networks and Stochastic Gradient Descent-MLPs with L-BFGS: What model size is effective?-MLPs with SGD: What batch size and step size?-Producing your own figure comparing batch size and learning rate.	**10**	CO3	K2, K3 and K6
	2.2 Trees and Random Forests for Bag of Words-Code Implementation of Decision Tree Regression-Decision Trees for Review Classification - Random Forests for Review Classification -Comparing Trees to Linear Models for Review Classification. Implementation of CNN, RNN, LSTM, Implementation of Boltzmann machine and Transformers (BERT, GPT3) .Generative deep learning (GAN).	**10**	CO4	K2, K3 and K6
	2.3 Project discussions -Classifying Images with Feature TransformationsClassifying Sentiment from Text Reviews-Recommendation	**10**	CO3	K2, K3 and K6

	Systems via Matrix Factorization-Text summarization - language Translation - Sentimental analysis- speech to text translatioXiv, Explore the keras ecosystem.			
Pedagogy:	Programming in lab and practical exercises			
Texts:	1. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly Media. 2. Chollet, F. (2018). Deep learning with Python (1st ed.). Manning Publications.			
References/ Readings:	1. Sarkar, D. (2016). Text analytics with Python: A practitioner's guide to natural language processing. Apress. 2. Chollet, F., & contributors. Keras: The Python deep learning API. Keras.io. https://keras.io/ Accessed on 22nd May 2025			
Web Resources:	1. Hughes, M. C. (2020). COMP 135: Introduction to Machine Learning – Fall 2020 assignments. Tufts University. https://www.cs.tufts.edu/comp/135/2020f/assignments.html Accessed on 22nd May 2025 2. Python Software Foundation. The Python standard library. https://docs.python.org/3/library/index.html Accessed on 22nd May 2025			

Discipline Specific Elective Courses

Title of the Course	Mathematics for Computer Science –II
Course Code	CSA-5204
Number of Credits	2
Theory/Practical	Theory
Level	400
Effective from AY	AY 2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	To build a strong foundation in maths required for learning computer science/data science subjects.	
Course Outcomes:		**Mapped to PSO**
	CO 1. To apply mathematics concepts in the modelling and design of computational problems.	PSO1, PSO2
	CO 2. To apply graph concepts to design computational problems.	PSO1, PSO2
	CO 3. To understand logical operations to design computational problems.	PSO1, PSO2
	CO 4. To understand set theory to design computational problems.	PSO1, PSO2

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	1.1 Mathematical logic: Statement (Proposition), Logical Connectives, Conditional, Bi-conditional, Converse, Inverse, Contrapositive, Exclusive OR, NAND, NOR, Tautology, Contradiction, Satisfiable, Duality Law, Algebra of propositions.	**5**	CO1, CO3	K1, K2, K3
	1.2 Functions and Relations: Basics of Set theory, Application of set theory, Relations and their properties, n-ary relations and their applications, representing relations, closures of relations, equivalence relations, partial orderings. Functions, properties of functions, Composition of Functions, Recursive functions.	**10**	CO1, CO4	K1, K2, K3
Module 2:	2.1 Graphs: Basic Concepts of Graphs, Computer Representations of Graphs, Isomorphic Graphs, Paths, Cycles and Circuits, Eulerian and Hamiltonian Graphs, Planar Graphs, Graph Coloring, Applications of Graphs. Trees: Trees, Spanning trees, Minimal Spanning Trees, Rooted Trees, Binary Trees, Binary Search Trees.	**15**	CO1, CO2	K1, K2,K3
Pedagogy:	lectures/assignements			
Texts:	Rosen, K. H. Discrete mathematics and its applications (Latest ed.). Tata McGraw-Hill Publishing Company.			
References/ Readings:	Goodaire, E. G., & Parmenter, M. M. Discrete mathematics with graph theory (Latest ed.). PHI Learning Pvt. Ltd.			

Title of the Course	Secure Coding
Course Code	CSA-5205
Number of Credits	2
Theory/Practical	Theory
Level	400
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil			
Course Objectives:	This course aims to equip students with knowledge and skills to write good-quality secure code that is resistant to common attacks and vulnerabilities.			
Course Outcomes:			**Mapped to PSO**	
	CO 1. To understand common software attacks and vulnerabilities.		PSO1, PSO2, PSO3	
	CO 2. To apply defensive coding practices and controls, implement programming safeguards using defensive coding principles.		PSO1, PSO2, PSO3	
	CO 3. To learn static and dynamic code analysis techniques to identify and mitigate vulnerabilities.		PSO1, PSO2, PSO3	
	CO 4. To learn about security testing methodologies and techniques.		PSO1, PSO2, PSO3	
Content:		**No of**	**Mapped**	**Cognitive**

		hours	to CO	Level
Module 1:	1.1 Introduction: Need for a secure System, CIA Triad, Security Concepts - exploit, threat, vulnerability, risk. Security Attacks, Security Services, Security Mechanisms. Security Principles: SD3, Secure by Design, by default, in deployment. Security principles. Threat Modelling: Threat Modelling process and its benefits: Identifying the Threats by Using Attack Trees and rating threats using DREAD, Risk Mitigation Techniques, Security Best Practices. Security techniques: authentication, authorization, tamper-resistant and privacy-enhancing techniques, Encryption, MACS, Digital Signatures, Auditing, Least Privilege, secure installation. Security testing, Security Code review, Handling privacy, and General good practices.	**15**	CO1, CO2	K1,K2, K3, K4, K5
Module 2:	2.1 Types of Security vulnerabilities: Buffer overflow, Invalid input, race conditions, access control problems, and poor cryptographic practices. Secure Coding Techniques: Protection against DoS attacks, Application Failure Attacks, CPU Starvation attacks. Buffer Overrun- Stack Overrun, Heap Overrun, Array Indexing Errors, Format String Bugs. Code Injection Attacks, countermeasures using tools like StackGuard Security Issues in C Language: String Handling, Avoiding Integer Overflows and Underflows and Type Conversion Issues, Memory Management Issues, and Canonical Representation Issues. Database and Web-specific issues: SQL Injection Techniques and Remedies XSS scripting attack and its types –Persistent and Non-persistent attack XSS Countermeasures and Bypassing the XSS Filters.	**15**	CO2, CO3, CO4	K1,K2, K3, K4
Pedagogy:	Lectures/ Tutorials/Assignments/ Flipped classroom			
Texts:	1. Howard, M., & LeBlanc, D. (2003). *Writing secure code* (2nd ed.). Microsoft Press.			

	2. Stallings, W., & Brown, L. (2010). Computer security: Principles and practice (1st ed.). Pearson Prentice Hall.
References/ Readings:	1. McConnell, S. (2004). Code complete (2nd ed.). Microsoft Press. 2. Deckard, J. (2005). Buffer overflow attacks: Detect, exploit, prevent (1st ed.). Syngress. 3. Swiderski, F., & Snyder, W. (2004). Threat modeling (1st ed.). Microsoft Professional.
Web Resources:	SecureCoding. *SecureCoding.org*. from https://www.securecoding.org/ Accessed on May 28, 2025,

Title of the Course	Data Mining
Course Code	CSA-5206
Number of Credits	2
Theory/Practical	Theory
Level	400
Effective from AY	2025-26
New Course	Yes
Bridge Course/ Value added Course	No
Course for advanced learners	No

Pre-requisites for the Course:	Nil	
Course Objectives:	To provide students with practical and theoretical knowledge of data mining techniques applied to software engineering data, enabling them to analyze, interpret, and extract actionable insights from software repositories and related sources.	
Course Outcomes:		**Mapped to PSO**
	CO 1. Identify and describe the methods and techniques commonly used in data mining	PSO1, PSO2
	CO 2. Demonstrate proficiency with the methods and techniques for obtaining, organizing, exploring, and analyzing data.	PSO1, PSO2, PSO5
	CO 3. Recognize how data analysis, inferential statistics, modeling, machine learning, and statistical computing can be utilized in an integrated capacity.	PSO1, PSO2
	CO 4. Demonstrate the ability to clean and prepare data for analysis and assemble data from a variety of sources.	PSO1, PSO2, PSO5

	CO 5. To use text and sentiment mining techniques to analyze software documentation, issue discussions, and app reviews using tools like PyDrill.			
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	Introduction to Data Mining, Data Sources & Preprocessing: Mining Git, bug tracking systems (e.g., JIRA), Cleaning and handling noisy, imbalanced software data. Pattern Discovery with Clustering: Types of clustering techniques, Self-Organizing maps, Clustering similar code segments and modules, Developer team clustering based on git commit behavior. Time Series forecasting: taxonomy of time series forecasting, time series decomposition, smoothing-based methods, regression-based methods, machine learning models, Performance evaluation. Anomaly Detection: Causes of outliers, Anomaly detection techniques, distance-based outlier detection, Density-based outlier detection	15	CO1 CO2 CO3 CO4	K1, K2, K3, K4
Module 2:	Deep learning: Neural Networks, Gradient descent, backpropagation, More than two classes, Dense layer, dropout layer, Introduction to CNN, RNN, and autoencoders. Feature Selection: Classification of Feature selection methods, Principal Component analysis, Information Theory-based filtering, based filtering. Text and Sentiment Mining: Mining requirements and documentation, Summarization and classification of technical text, Sentiment analysis on app reviews, and issue discussions. Tools, Platforms, and Ethics Overview: PyDriller and NLTK.	15	CO1 CO2 CO3 CO5	K1, K2, K3, K4, K5
Pedagogy:	Lectures/ Tutorials/Assignments/ Flipped classroom			
Texts:	1. Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to data mining. Pearson Education. 2. Khotu, V., & Deshpande, B. Data science: Concept and practice (2nd ed.). Morgan Kaufmann Publishers. 3. Pujari, A. K. *Data mining techniques*. Universities Press.			

References/ Readings:	1. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets* (2nd ed.). Cambridge University Press. 2. Han, J., & Kamber, M. (2001). Data mining: Concepts and techniques (1st Indian reprint ed.). Harcourt India Private Limited.
Web Resources:	1. Loria, S. "TextBlob: Simplified text processing". https://textblob.readthedocs.io/en/dev/ Accessed on May 28, 2025, 2. Cosentino, F. "PyDriller: Python framework for mining software repositories". https://pydriller.readthedocs.io/en/latest/ Accessed on May 28, 2025, 3. Bird, S., Klein, E., & Loper, E. "Natural Language Toolkit — NLTK 3.8 documentation". https://www.nltk.org/ Accessed on May 28, 2025, 4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. "scikit-learn: Machine learning in Python"., https://scikit-learn.org/stable/ Accessed on May 28, 2025 5. Han, J., Kamber, M., & Pei, J. (2012). Data mining: Concepts and techniques (3rd ed.). https://dataminingbook.info/book_html/ Accessed on May 28, 2025. 6. GeeksforGeeks. (2024, December 16). Deep learning tutorial. https://www.geeksforgeeks.org/deep-learning-tutorial/ Accessed on May 28, 2025.