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 with Grade A+)

GU/Acad –PG/BoS -NEP/2026-27/153 Date: 18.06.2026

CIRCULAR

Ref. No.: GU/Acad –PG/BoS -NEP/2025-26/236 dated 09.07.2025

In supersession to the above referred Circular, the Programme structure and syllabus of Semester III and IV of the **Master of Computer Applications (M.C.A.)** Programme approved by the Standing Committee of the Academic Council in its meeting held 24th & 25th November 2025 is attached.

The Syllabus of Semester I and II approved earlier by the Academic Council in its meeting held on 13th & 14th June 2025 is also 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 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. 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.

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**	

SEMESTER III			
Research Specific Elective (RSE) Courses (12 credits)			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	**CSA-6000**	TDD and Design Patterns	2T
2	**CSA-6001**	TDD and Design Pattern Lab	2P
3	**CSA-6002**	Cryptography and Network Security	4T
4	**CSA-6003**	Design Thinking for UI/UX	4T
5	**CSA-6004**	Data Engineering	2T
6	**CSA-6005**	Data Engineering Lab	2P
7	**CSA-6006**	Modern Development Practices & DevOps	2T
8	**CSA-6007**	Modern Development Practices & DevOps Lab	2P
9	**CSA-6008**	Research Methodology	4T
		Total Credits for RSE Courses in Semester III	**12**
Discipline Specific Vocational Elective (DSVE) Courses (8 credits)			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	**CSA-6401**	Internet of Things	2T+2P
2	**CSA-6402**	Cloud Computing	2T+2P
3	**CSA-6403**	Mobile Application Development	2T+2P
4	**CSA-6404**	Ethical Hacking	2T+2P
5	**CSA-6405**	AI for Software Development	2T+2P
6	**CSA-6406**	Parallel Programming and Scientific Computing	2T+2P
7	**CSA-6407**	Functional Programming	2T+2P
		Total Credits for DSVE Courses in Semester III	**8**
		Total Credits in Semester III	**20**

MCA Sem III and Sem IV Discipline Specific Dissertation (DSI)(40 Credit Dissertation)			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	**CSA-6501**	Dissertation (inclusive of 4 Credits Course/s of Research Methodology and 4 credit Dissertation related courses)	40
		Total Credits in Semester IV	**40**

SEMESTER IV			
Generic Elective (GE) Courses (20 credits)			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	**CSA-6201**	Personal Development and Professional Ethics	4T
2	**CSA-6202**	Computer Security	4T
3	**CSA-6203**	Ethical Use of AI	4T
4	**CSA-6204**	Prompt Engineering	4T
5	**CSA-6205**	Spatial Data Applications	4T
6	**CSA-6206**	Educational Technology	4T
7	**CSA-6207**	Visual Modeling and 3-D Printing	4T
8	**CSA-6208**	Digital Transformation	4T
9	**CSA-6209**	eGovernance: Service Design & Delivery	4T
10	**CSA-6210**	Digital Story Telling & Gamification	4T
11	**CSA-6211**	Natural Language Processing	4T
12	**CSA-6212**	Computational Linguistics	4T
13	**CSA-6213**	Software Project Management	4T
		Total Credits for GE Courses in Semester IV	**20**

OR

A. MCA Semester IV (Discipline Specific Dissertation (DSD)(20 Credit Dissertation))			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	**CSA-6502**	Discipline Specific Dissertation, including 4 credits of Research Methodology / Dissertation-related courses	20
		Total Credits in Semester IV	**20**

OR

B. MCA Sem IV Discipline Specific Internship (DSI)			
Sr. No.	**Course Code**	**Title of the Course**	**Credits**
1	**CSA-6503**	Industry Internship (inclusive of one 4-credit course under DSVE during the 3rd Semester)	20
		Total Credits in Semester IV	**20**

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

BRIDGE COURSE

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 hours**	**Mapped to CO**	**Cognitive 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	**8**	CO3	K1, K2

	memory organization, Types of Memory-internal and external, Cache memory, Memory interleaving, 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),	**9**	CO6	K1, K2

	ER Diagram, Types of Entities and Relationships, Keys and integrity constraints,Converting ER Diagram to Tables, Normal Forms , DDL and basic DML 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			

[Back to Index]

SEMESTER I

Discipline Specific Core (DSC) 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	PSO1, PSO2, PSO5,

	compare different scheduling algorithms based on their performance characteristic		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).			

[Back to Index]

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 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.	**15**	CO6	K6

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

[Back to Index]

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 hours	Mapped to CO	Cognitive 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.

[Back to Index]

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 hours	Mapped to CO	Cognitive 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	**10**	CO4	K6

	(simulate tables). Write programs: Management system with file storage. Mini Project & 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.			

[Back to Index]

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. 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.	**10**	CO3	K2, K3
	2.2 Process Management: Concept of UNIX process. Role of init in process creation.	**4**	CO2,	K1, K2

	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.		CO3	
	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,			

[Back to Index]

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 oriented computation model. Use of static attributes and methods. Use of enums. Implementation level Inheritance: defining base class and derived class. Base class to	**10**	CO2	K3

	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 threads. UML: use of UML diagramming tool. Creation of class diagrams, sequence diagram, use case diagram, activity diagram.	**8**	CO6	K3
	2.3 Project 1. A mini project for a business application for which UML based analysis and design model is to be created.	**15**	CO6	K4,K6

	2. An implementation of the design of mini project using OOP (Java/Python etc)			
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.			

[Back to Index]

Discipline Specific Elective (DSE) 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 hours	Mapped to CO	Cognitive 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.

[Back to Index]

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)

[Back to Index]

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	PSO1, PSO6

	assignment problems.			
	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	**7**	CO 6	K2, K3,

	consideration. Advantages of Network.			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.			

[Back to Index]

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	PSO1, PSO2

	problem.			
	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, CO4	K3, K4, K6

	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.			

[Back to Index]

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. Concurrency Control: Serializability, Locking Techniques, Time stamp ordering, Granularity of Data items	**15**	CO5	K2, K3, K4

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.

[Back to Index]

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**	**Mapped**	**Cognitive**

		hours	to CO	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

[Back to Index]

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 algorithm. Reinforcement learning – Introduction- learning task-Q learing-non deterministic rewards and actions-temporal difference learning.	**6**	CO2, CO3	K3, K4
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.

[Back to Index]

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**	**Mapped**	**Cognitive**

		hours	to CO	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.			

[Back to Index]

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**	**Mapped**	**Cognitive**

		hours	to CO	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 Complex SQL Statements • Using aggregate functions (using Group by and having clauses).	**18**	CO1, CO2	K2, K3, K4

	• 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 design, including triggers that may need to be written (v) User Manual Peer reviews of ERDs are held in the class.	**6**	CO4	K4, K6
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.

[Back to Index]

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 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,	**15**	CO1, CO2, CO3	K2, K3, K6

	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

[Back to Index]

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**	**Mapped**	**Cognitive**

		hours	to CO	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 Systems via Matrix Factorization-Text summarization - language Translation - Sentimental	**10**	CO3	K2, K3 and K6

	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			

[Back to Index]

Discipline Specific Elective (DSE) 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.			

[Back to Index]

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 hours	Mapped to CO	Cognitive 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,

[Back to Index]

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/	1. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets* (2nd ed.). Cambridge University			

Readings:	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.

[Back to Index]

SEMESTER III

Research Specific Elective (RSE) Courses

Title of the Course	TDD & Design Patterns
Course Code	CSA-6000
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-5001, CSA-5005	
Course Objectives:	Explore and apply the principles of Test-Driven Development (TDD) and common design patterns to develop high-quality, modular, and testable software, while critically evaluating their effectiveness and efficiency in real-world software engineering projects.	
Course Outcomes:		**Mapped to PSO**
	CO1. Introduce the principles and methodology of Test-Driven Development (TDD) in software engineering.	PSO1, PSO2

	CO2. Identify and explain common design patterns (Creational, Structural, Behavioral)		PSO1, PSO2	
	CO3. Apply TDD and design patterns together to produce high-quality, modular, and testable code		PSO1, PSO2, PSO3	
	CO4. Evaluate software designs for correctness, test coverage, and maintainability		PSO3, PSO5	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1	Introduction to TDD: Concepts, Red-Green-Refactor cycle, unit testing frameworks Writing Effective Tests: Test case design, mock objects, code coverage Introduction to Design Patterns: Importance, types, GoF patterns overview Creational Patterns: Singleton, Factory, Builder, Prototype Structural Patterns: Adapter, Composite, Decorator, Facade	**15**	CO1, CO2	K2 K3
Module 2:	Behavioral Patterns: Observer, Strategy, Command, Template Method Integrating TDD & Design Patterns: Refactoring, modular design, maintainable code Evaluation & Best Practices: Test coverage, CI/CD integration, metrics, code reviews	**15**	CO2, CO3, CO4	K3, K4, K5
Pedagogy:	Interactive lectures with real-world coding examples Demonstrations using Java/Python unit testing frameworks (JUnit, pytest) Case studies on design pattern applications Group discussions and analysis of TDD practices			
Texts:	1. Beck, K. (2003). Test-driven development: By example. Addison-Wesley. 2. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: Elements of reusable object-oriented software. Addison-Wesley. 3. Freeman, E., Freeman, E., & Sierra, K. (2014). Head First Design Patterns. O'Reilly Media. 4. Meszaros, G. (2007). xUnit test patterns: Refactoring test code. Addison-Wesley.			

References/ Readings:	1. Hunt, A., & Thomas, D. (1999). The Pragmatic Programmer. Addison-Wesley. 2. van Vliet, H. (2008). Software engineering: Principles and practice (3rd ed.). Wiley.
Web Resources:	1. JUnit. *Official documentation*. JUnit. Retrieved October 13, 2025, from https://junit.org 2. pytest. *pytest documentation*. pytest. Retrieved October 13, 2025, from https://docs.pytest.org 3. Refactoring Guru. *Design patterns in object-oriented programming*. Refactoring Guru. Retrieved October 13, 2025, from https://refactoring.guru/design-patterns

[Back to Index]

Title of the Course	Test-Driven Development & Design Patterns Lab
Course Code	CSA-6001
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, CSA-5005,CSA-5012,	
Course Objectives:	Develop robust, maintainable, and modular software by applying Test-Driven Development (TDD) principles and common design patterns, while evaluating code quality through test coverage and best practices.	
Course Outcomes:		**Mapped to PSO**
	CO1. Apply TDD by writing unit tests first and developing code iteratively	PSO1, PSO2
	CO2. Implement Creational, Structural, and Behavioral design patterns in code	PSO2, PSO5
	CO3. Integrate design patterns with TDD in mini-projects	PSO1, PSO2, PSO3
	CO4. Evaluate code quality, test coverage, and maintainability	PSO3, PSO5

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1	1. Environment setup, IDE, unit testing frameworks 2. Writing first unit test & Red-Green-Refactor cycle 3. Test doubles & Mock objects 4. Singleton and Factory pattern implementation 5. Builder & Prototype pattern implementation 6. Adapter & Composite pattern implementation 7. Decorator & Facade pattern implementation 8. Observer & Strategy pattern implementation 9. Command & Template Method pattern implementation	**30**	CO1, CO2	K2,K3
Module 2:	1. Apply TDD to a small module 2. Integrate design patterns into module 3. Refactoring & code improvements 4. Test coverage analysis & evaluation 5. Continuous Integration simulation	**30**	CO1,CO2, CO3, CO4	K4,K5,K6
Pedagogy:	Hands-on coding exercises using TDD Implementation of multiple design patterns Mini-projects integrating TDD with design patterns Peer review, test coverage analysis, and evaluation			
Texts:	1. Beck, K. (2003). Test-Driven Development: By Example. Addison-Wesley Professional. 2. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional. 3. Freeman, E., Robson, E., Bates, B., & Sierra, K. (2014). Head First Design Patterns. O'Reilly Media. 4. Meszaros, G. (2007). xUnit Test Patterns: Refactoring Test Code. Addison-Wesley Professional.			

References/ Readings:	1. Hunt, A., & Thomas, D. (2019). *The Pragmatic Programmer: Your Journey to Mastery* (20th Anniversary ed.). Addison-Wesley Professional. 2. van Vliet, H. (2008). Software Engineering: Principles and Practice (3rd ed.). Wiley. 3. Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley Professional.
Web Resources:	1. JUnit. *Official documentation*. JUnit. Retrieved October 13, 2025, from https://junit.org 2. pytest. *pytest documentation*. pytest. Retrieved October 13, 2025, from https://docs.pytest.org 3. Refactoring Guru. *Design patterns in object-oriented programming*. Refactoring Guru. Retrieved October 13, 2025, from https://refactoring.guru/design-patterns 4. GitHub. *Repositories for Test-Driven Development (TDD) and design pattern examples*. GitHub. Retrieved October 13, 2025, from https://github.com/topics/test-driven-development 5. Mockito. *Mockito documentation: Mocking framework for Java*. Mockito. Retrieved October 13, 2025, from https://site.mockito.org 6. GitHub Docs. *GitHub Actions documentation*. GitHub. Retrieved October 13, 2025, from https://docs.github.com/en/actions 7. Jenkins. *Jenkins user documentation*. Jenkins. Retrieved October 13, 2025, from https://www.jenkins.io/doc/

[Back to Index]

Title of the Course	Cryptography and Network Security
Course Code	CSA-6002
Number of Credits	4
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 basic security concepts, equip students with the knowledge and skills to protect data confidentiality and integrity, verify user and data authenticity in network environment, identify threats and understand countermeasures to be adapted to protect data, to understand cryptographic principles, design and security protocols	
Course Outcomes:		**Mapped to PSO**
	CO1. Understanding of core concepts of security encompassing confidentiality, integrity, authentication, non-repudiation and availability of data.	PSO1, PSO5
	CO2. Apply various symmetric and asymmetric encryption techniques.	PSO1, PSO5
	CO3. Understand the use of hashing algorithms and Message Authentication Codes (MACs) for data integrity.	PSO1, PSO5
	CO4. Understand the use of digital signatures and their associated digital certificates	PSO1, PSO2, PSO5

	for authenticity and non-repudiation			
	CO5. Apply security features within network protocols like IPSec, SSL/TLS, and HTTPS, WiFi Security.		PSO1, PSO2, PS05, PSO6	
	CO6. Apply security concepts and algorithms in practical settings, including network infrastructure and web applications to enhance security.		PSO1, PSO2, PSO5, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1	**Introduction** Computer Security Concepts, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, Fundamental Security Design Principles, Attack Surfaces and Attack Trees, A Model for Network Security **Classical Encryption Techniques** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques **Block Ciphers and the Data Encryption Standard** Traditional Block Cipher Structure, The Data Encryption Standard, The Strength of DES, Block Cipher Design Principles **Advanced Encryption Standard** Finite Field Arithmetic, AES Structure, AES Transformation Functions, AES Key Expansion, AES Implementation 197 **Block Cipher Operation** Multiple Encryption and Triple DES, Electronic Codebook, Cipher Block	**15**	CO1, CO2	K1, K2, K3 K4

	Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode, XTS-AES Mode for Block-Oriented Storage Devices, Format-Preserving Encryption			
Module 2:	**Random Bit Generation and Stream Ciphers** Principles of Pseudorandom Number Generation, Pseudorandom Number Generators, Pseudorandom Number Generation Using a Block Cipher, Stream Ciphers, RC4 **Public-Key Cryptography and RSA** Principles of Public-Key Cryptosystems, The RSA Algorithm **Other Public-Key Cryptosystems** Diffie-Hellman Key Exchange, Elgamal Cryptographic System, Elliptic Curve Arithmetic, Elliptic Curve Cryptography **Cryptographic Hash Functions** Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA), SHA-3	**15**	CO1, CO2, CO3	K1, K2, K3, K4
Module 3:	**Message Authentication Codes** Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MACs, MACs Based on Hash Functions: HMAC, MACs Based on Block Ciphers: DAA and CMAC, Authenticated Encryption: CCM and GCM, Key Wrapping **Digital Signatures** Digital Signatures, Elgamal Digital Signature Scheme, Schnorr Digital	**15**	CO3, CO4 CO5	K1, K2, K3, K4, K5

	Signature Scheme, NIST Digital Signature Algorithm, Elliptic Curve Digital Signature Algorithm, RSA-PSS Digital Signature Algorithm **Key Management and Distribution** Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public-Key Infrastructure **User Authentication** Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption Kerberos, Remote User-Authentication Using Asymmetric Encryption, Federated Identity Management, Personal Identity Verification			
Module 4:	**Network Access Control and Cloud Security** Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control, Cloud Computing, Cloud Security Risks and Countermeasures, Data Protection in the Cloud, Cloud Security as a Service, Addressing Cloud Computing Security Concerns **Transport-Level Security** Web Security Considerations, Transport Layer Security, HTTPS **Wireless Network Security** Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN Overview, IEEE 802.11i Wireless LAN Security **IP Security** IP Security Overview, IP Security Policy, Encapsulating Security Payload,	**15**	CO1, CO2, CO3 CO4 CO5, CO6	K1, K2, K3, K4, K5

	Combining Security Associations, Internet Key Exchange, Cryptographic Suites			
Pedagogy:	Lectures/tutorials/ assignments/ presentations/ case studies/ class discussions.			
Texts:	1. Stallings, W. (2017). Cryptography and network security: Principles and practice (7th ed.). Pearson Education Limited.			
References/ Readings:	1. Kahate, A. (2008). Cryptography and network security (2nd ed.). McGraw Hill. 2. Forouzan, B. A. (2008). Introduction to cryptography and network security. McGraw Hill.			

[Back to Index]

Title of the Course	Design Thinking for UI/UX
Course Code	CSA-6003
Number of Credits	4T
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-5001, CSA-5005, CSA-5008	
Course Objectives:	1. To apply Design Thinking principles and processes—empathy, problem definition, ideation, prototyping, and evaluation—toward creating user-centered interactive systems that integrate UI/UX principles and usability heuristics.	
Course Outcomes:		**Mapped to PSO**
	CO1. Explain phases & principles of Design Thinking and relevance to UI/UX	PSO1, PSO2
	CO2. Conduct user research, empathy mapping, persona creation	PSO1, PSO4
	CO3. Apply ideation, prototyping, and usability testing	PSO1, PSO2, PSO3
	CO4. Evaluate UI/UX designs for usability & accessibility	PSO2, PSO5

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	Introduction to Design Thinking and UI/UX Empathize Phase: User research, empathy maps, personas	**15**	CO1, CO2, CO3	K2, K3, K4
Module 2:	Define Phase: Problem statements, insights synthesis Ideate Phase: Brainstorming, SCAMPER, storyboarding	**15**	CO1, CO2, CO3	K2, K3, K4
Module 3:	Prototype Phase: Wireframes, mockups Test Phase: Usability testing, heuristic evaluation	**15**	CO2, CO3, CO4	K3,K4, K5
Module 4:	DT in Practice: Agile integration, design sprints Emerging Trends: Voice UX, AR/VR, AI personalization	**15**	CO2, CO3, CO4	K3,K4, K5
Pedagogy:	Lectures/Tutorials, Flipped classroom, assignments, peer-teaching, role playing, games			
Texts:	1. Brown, T. (2009). Change by Design: How Design Thinking Creates New Alternatives for Business and Society. Harper Business. 2. Norman, D. A. (2013). The Design of Everyday Things: Revised and Expanded Edition. Basic Books. 3. Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014). About Face: The Essentials of Interaction Design (4th ed.). Wiley. 4. Krug, S. (2014). Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd ed.). New Riders.			
References/ Readings:	1. Liedtka, J., & Ogilvie, T. (2011). Designing for Growth: A Design Thinking Toolkit for Managers. Columbia University Press. 2. IDEO.org. (2015). The Field Guide to Human-Centered Design: Design Kit. IDEO.org.			
Web Resources:	1. Figma. Learn design. Figma. Retrieved October 13, 2025, from https://www.figma.com/resources/learn-design/ 2. Nielsen Norman Group. Home. Retrieved October 13, 2025, from https://www.nngroup.com/ 3. Design sprint kit. Retrieved October 13, 2025, from https://designsprintkit.withgoogle.com/			

[Back to Index]

Title of the Course	Data Engineering
Course Code	CSA-6004
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-5006,CSA-5010, CSA-5007, CSA-5011	
Course Objectives:	To provide students with a comprehensive understanding of data engineering principles, tools, and best practices, enabling them to design, build, and manage efficient, reliable, and scalable data systems.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand the role of data engineering and the structure of data pipelines	PSO1, PSO2
	CO2. Apply data modeling and transformation using modern open-source tools	PSO2, PSO5
	CO3. Analyze and ensure data quality and governance in design	PSO2, PSO3
	CO4. Build and present meaningful data visualizations using tools.	PSO2, PSO5, PSO6

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	**Data Engineering Concepts:** Introduction to Data Engineering Role in data teams (Analyst, Scientist, ML Engineer) Data Lifecycle – Generation, Ingestion, Storage, Processing, Serving Data Types & Properties– Structured, semi-structured, unstructured;– Volume, variety, velocity Data Formats– CSV, JSON, Parquet Data Storage– Data warehouses, lakes, lakehouses (conceptual) Row-based vs Column-based storage Extract, Transform, Load(ETL) vs ELT processes (conceptual) NoSQL Concepts– Document, key-value, graph, column-family Data Modeling– Star and snowflake schemas (conceptual)	**15**	CO1	K1,K2
Module 2:	**Data Processing, Tools, Automation & Architectures:** Introduction to the Modern Data Stack (open-source oriented)– Use cases:Business Intelligence (BI) stacks. Overview of Key Tools:– dbt (Cloud), Apache Airflow Workflow Automation– Airflow concepts: DAGs Data Transformation– Cleaning, enrichment and Transform. Data Quality– Validation and profiling. Governance– Data lineage, documentation, glossary, Identity and Access Management (IAM) Data Visualization– Dashboards, storytelling; Tools: Tableau, Looker Studio.	**15**	CO2, CO3, CO4	K3,K4
Pedagogy:	Lectures/Tutorials, Flipped classroom, assignments, peer-teaching, role playing, games			

Texts:	1. Crickard, P. (2020). Data engineering with Python: Work with massive datasets to design data models and automate data pipelines using Python. Packt Publishing. 2. Reis, J., & Housley, M. (2022). Fundamentals of data engineering: Plan and build robust data systems. O'Reilly Media. 3. Kretz, A. (2020).The data engineering cookbook 4. Kimball, R., & Ross, M. (2013). The data warehouse toolkit: The definitive guide to dimensional modeling (3rd ed.). Wiley.
References/ Readings:	1. Nussbaumer Knaflic, C. (2015). Storytelling with data: A data visualization guide for business professionals. Wiley.
Web Resources:	1. dbt Labs. *Introduction — dbt Documentation*. Retrieved October 2025, from https://docs.getdbt.com/docs/introduction docs.getdbt.com 2. The PostgreSQL Global Development Group. *PostgreSQL Documentation*. Retrieved October 2025, from https://www.postgresql.org/docs/ 3. Apache Software Foundation. (n.d.). *Apache Airflow Documentation (stable)*. Retrieved October 2025, from https://airflow.apache.org/docs/apache-airflow/stable/index.html 4. Soda. (n.d.). *Soda Core / Soda Documentation*. Retrieved October 2025, from https://docs.soda.io/ 5. Docker, Inc. *Get Started with Docker*. Retrieved October 2025, from https://docs.docker.com/get-started/ 6. GitHub, Inc. *GitHub Actions Documentation*. Retrieved October 2025, from https://docs.github.com/en/actions 7. Snowflake Inc. *Data Engineering with Apache Airflow, Snowflake, Snowpark, dbt & Cosmos*. Retrieved October 2025, from https://quickstarts.snowflake.com/guide/data_engineering_with_apache_airflow/index.html

[Back to Index]

Title of the Course	Data Engineering Lab
Course Code	CSA-6005
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-5006,CSA-5010, CSA-5007, CSA-5011	
Course Objectives:	To introduce students to core data engineering concepts, tools, and workflows for building reliable, automated, and scalable data systems.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand the role of data engineering and the structure of data pipelines	PSO1, PSO2,
	CO2. Identify and work with various data types, formats, and storage architectures.	PSO2, PSO5
	CO3. Analyze and ensure data quality and governance in design	PSO2, PSO3
	CO4. Create automated and visualized end-to-end project using open/free platforms	PSO2, PSO5, PSO6

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1	1. Data Cleaning and Profiling with Pandas in Colab 2. Convert and Compare CSV and Parquet Formats: Convert data between CSV and Parquet, formats and compare their structure, size, and efficiency. 3. Ingest Google Sheets Data into Snowflake:Connect Google Sheets as a data source and load data into a Snowflake table. 4. Build a Star Schema and Write Basic SQL Queries:Design a star schema data model and query it using basic SQL commands. 5. Use Cloud for ELT Transformation Models:Develop and run ELT transformation models using Cloud. 6. Perform data transformation tasks by connecting to a SQL database using DBeaver and writing SQL scripts 7. Explore NoSQL data handling by analyzing sample datasets in MongoDB	**30**	CO1, CO2	K1,K2
Module 2:	1. Containerize a basic Python-based ETL script using Docker. 2. Set up and execute ETL workflows using Apache Airflow (manual or Docker-based). 3. Perform data quality checks on datasets using Soda CLI. 4. Integration of Soda with Apache Airflow: Automate data quality checks within an Airflow DAG using Soda. 5. Data Visualization using Looker Studio:Visualize processed data from Snowflake or Spreadsheets using Looker Studio. 6. Mini Project – End-to-End Data Pipeline:Implement a complete data pipeline integrating ETL, automation, testing, CI/CD, and visualization tools.	**30**	CO2, CO3, CO4	K3,K4,K5,K6
Pedagogy:	Tutorials/Lab Assignments/Mini Project			

Texts:	1. Crickard, P. (2020). Data engineering with Python: Work with massive datasets to design data models and automate data pipelines using Python. Packt Publishing. 2. Reis, J., & Housley, M. (2022). Fundamentals of data engineering: Plan and build robust data systems. O'Reilly Media. 3. Kretz, A. (2020).The data engineering cookbook 4. Kimball, R., & Ross, M. (2013). The data warehouse toolkit: The definitive guide to dimensional modeling (3rd ed.). Wiley.
References/ Readings:	1. Nussbaumer Knaflic, C. (2015). Storytelling with data: A data visualization guide for business professionals. Wiley.
Web Resources:	1. dbt Labs. (n.d.). *Introduction — dbt Documentation*. Retrieved October 2025, from https://docs.getdbt.com/docs/introduction docs.getdbt.com 2. The PostgreSQL Global Development Group. (n.d.). *PostgreSQL Documentation*. Retrieved October 2025, from https://www.postgresql.org/docs/ 3. Apache Software Foundation. (n.d.). *Apache Airflow Documentation (stable)*. Retrieved October 2025, from https://airflow.apache.org/docs/apache-airflow/stable/index.html 4. Soda. (n.d.). *Soda Core / Soda Documentation*. Retrieved October 2025, from https://docs.soda.io/ 5. Docker, Inc. (n.d.). *Get Started with Docker*. Retrieved October 2025, from https://docs.docker.com/get-started/ 6. Snowflake Inc. (n.d.). *Data Engineering with Apache Airflow, Snowflake, Snowpark, dbt & Cosmos* [Quickstart guide]. Retrieved October 2025, from https://quickstarts.snowflake.com/guide/data_engineering_with_apache_airflow/index.html

[Back to Index]

Title of the Course	Modern Development Practices & DevOps
Course Code	CSA-6006
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-5000, CSA-5002, CSA-5008	
Course Objectives:	To provide a comprehensive understanding of modern development practices, with a focus on containerization, cloud computing paradigms, and DevOps.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand the concepts, need and advantages of containerization	PSO2, PSO5
	CO2. Apply various techniques of API querying	PSO2
	CO3. Apply the concepts of DevOps to solve problems in software development and deployment	PSO2, PSO5
	CO4. Understand the fundamental concepts of cloud computing, differentiating between IaaS, PaaS, and SaaS service models and public, private, and hybrid deployment models.	PSO1, PSO2, PSO5

Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Introduction** ● Ever-changing development terrain ● Importance of development at scale ● Emergence of Cloud Services and Devops **Containerization** ○ Introduction to Containerization ○ Container Lifecycle ○ Case study of any one containerization tool (e.g. Docker, etc) which should include namespaces, commands, CLI, image creation, image registry **APIs** ○ Introduction to APIs, Types of API ○ API Query- REST API and GraphQL **Overview of DevOps:** ○ Introduction to DevOps ○ DevOps Lifecycle ○ DevOps Delivery Pipeline ○ **Continuous Integration/ Continuous Delivery (CI/CD)** ○ Introduction to CI/CD ○ Continuous Delivery v/s Continuous Deployment ○ Case study of any one CI/CD tool(CircleCI/Jenkins, etc). Case study should include architecture, pipeline	**15**	CO1, CO2, CO3	K1, K2, K3

	and plugin management			
Module 2:	**Cloud Computing** ● Overview ● Cloud Models - IaaS, PaaS, SaaS, Public/Private/Hybrid Cloud ● Components - Virtualization & VMs, File Storage, Server Instances, etc. ● Cloud Services - Virtualization on cloud, IAM and Service accounts, Server-less service, cloud based CI/CD ● Case study of any one cloud (e.g. Amazon AWS/ Google Cloud/ MS Azure) **Continuous Monitoring** ● Introduction to continuous monitoring ● Types: Infrastructure Monitoring, Application Monitoring and Network Monitoring ● Case study of any one continuous monitoring tool(e.g. Nagios, Prometheus, etc) **Configuration Management** ● Introduction to Configuration Management ● Case study of any one Configuration Management(e.g. Ansible, Chef, etc). Case study should include Infrastructure as Code, Inventory Management, playbooks/cookbooks	**15**	CO3, CO4	K1, K2, K3, K4
Pedagogy:	Hands-on assignments / tutorials / peer-teaching			
Texts:	1. Verona, J. (2016). *Practical DevOps* (1st ed). Packt Publishing. 2. Erl, T. (with Mahmood, Z., & Puttini, R.). (2013). Cloud Computing: Concepts, Technology & Architecture. ServiceTech Press.			
References/ Readings:	1. Buyya, R., Broberg, J., & Goscinski, A. M. (2010). Cloud computing: Principles and paradigms. John Wiley &			

	Sons.
Web Resources:	1. (N.d.). Retrieved October 9, 2025, from https://docs.aws.amazon.com/ 2. Google cloud documentation. (n.d.). Google Cloud. Retrieved October 9, 2025, from https://cloud.google.com/docs 3. JnHs. (n.d.). Azure documentation. Retrieved October 9, 2025, from https://learn.microsoft.com/en-us/azure/ 4. Home. (2025, September 12). Docker Documentation. https://docs.docker.com/ 5. Introduction to graphql | graphql. (n.d.). Retrieved October 9, 2025, from https://graphql.org/learn/ 6. Getting started with ansible—Ansible community documentation. (n.d.). Retrieved October 9, 2025, from https://docs.ansible.com/ansible/latest/getting_started/index.html

[Back to Index]

Title of the Course	Modern Development Practices & DevOps Lab
Course Code	CSA-6007
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 Problem Solving and Programming Lab CSA-5004 Linux Lab CSA-5012 Web Development Lab	
Course Objectives:	To provide hands-on skills to the students on the practical use and aspects of modern development technologies, tools and platforms prevalent in the software development industry with a focus on containerization, cloud computing, and DevOps.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand container and image creation using CLI and GUI	PSO2
	CO2. Understand how to query various public APIs	PSO2
	CO3: Analyze and use various DevOps tools to solve problems in software development and deployments	PSO1, PSO2

	CO4: Analyze various cloud programming models and apply them to solve problems on the cloud.		PSO1, PSO2	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Containerization** ● Assignments should be based on creating containers from pre-existing images using tools like Docker, creating own container images and pushing container images to Docker Hub via CLI and via GUI. **APIs** ● Assignments should be based on querying REST API and Graph QL **CI/CD** ● Assignments should be based on constructing a CI/CD Pipeline with the use of various build schedules and build triggers.	**30**	CO1, CO2, CO3	K2, K3, K4
Module 2:	**Cloud Services** Assignments should encompass implementation of ● Storage service ● Database service ● Compute services ● Server-less service ● IAM ● Creation and use of Service Accounts ● Cloud-based CI/CD **Continuous Monitoring** ● Assignments should be based on continuous monitoring for	**30**	CO3, CO4	K2, K3, K4, K5

	Infrastructure, Application & Network using tools like . Nagios, Prometheus, etc **Configuration Management** ● Assignments should be based on Configuration Management using tools like Ansible, Chef etc. **Mini-Project** ● Ideally done in a group. Concepts and tools (or similar) learnt in the course to be implemented/incorporated.			
Pedagogy:	Hands-on assignments / tutorials / peer-teaching			
Texts:	1. Verona, J. (2016). *Practical DevOps* (1st ed). Packt Publishing. 2. Erl, T. (with Mahmood, Z., & Puttini, R.). (2013). Cloud Computing: Concepts, Technology & Architecture. ServiceTech Press.			
References/ Readings:	1. Buyya, R., Broberg, J., & Goscinski, A. M. (2010). Cloud computing: Principles and paradigms. John Wiley & Sons.			
Web Resources:	1. (N.d.). Retrieved October 9, 2025, from https://docs.aws.amazon.com/ 2. Google cloud documentation. (n.d.). Google Cloud. Retrieved October 9, 2025, from https://cloud.google.com/docs 3. JnHs. (n.d.). Azure documentation. Retrieved October 9, 2025, from https://learn.microsoft.com/en-us/azure/ 4. Home. (2025, September 12). Docker Documentation. https://docs.docker.com/ 5. Introduction to graphql | graphql. (n.d.). Retrieved October 9, 2025, from https://graphql.org/learn/ 6. Getting started with ansible—Ansible community documentation. (n.d.). Retrieved October 9, 2025, from https://docs.ansible.com/ansible/latest/getting_started/index.html			

[Back to Index]

Title of the Course	Research Methodology
Course Code	CSA-6008
Number of Credits	4T
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:	Nil	
Course Objectives:	• This course aims to impart a comprehensive understanding of research methodology, covering the techniques for defining research problems and delving into the meaning of interpretation along with associated techniques and precautions in the research process.	
Course Outcomes:		**Mapped to PSO**
	CO1: Understand the fundamentals of research, including types, methods, and ethical considerations, and differentiate between qualitative and quantitative approaches.	PSO1, PSO6
	CO2: Develop the ability to formulate research problems, objectives, and hypotheses suitable for scientific investigation.	PSO1, PSO2
	CO3: Acquire skills in data collection, sampling techniques, and analysis using appropriate statistical and research tools.	PSO1, PSO2, PSO5, PSO4

	CO4: Demonstrate the capability to write, present, and critically evaluate research reports and papers according to academic standards.		PSO2, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	Introduction to research, Definitions and characteristics of research, Types of Research, Research Process, Problem definition, Objectives of Research, Research Questions, Research design, Quantitative vs. Qualitative Approach, Building and Validating Theoretical Models, Exploratory vs. Confirmatory Research, Experimental vs. Theoretical Research, Importance of reasoning in research.	**15**	CO1, CO4	K1, K2, K3, K4
Module 2:	Problem Formulation, Understanding Modeling & Simulation, Literature Review, Referencing, Information Sources, Information Retrieval, Indexing and abstracting services, Citation indexes, Development of Hypothesis, Measurement Systems Analysis, Error Propagation, Validity of experiments, Statistical Design of Experiments, Data/Variable Types & Classification, Data collection, Numerical and Graphical Data Analysis: Sampling, Observation, Interpretation of Results.	**5**	CO1, CO3	K2, K3, K4
Module 3:	Statistics: Probability & Sampling distribution, Estimation, Measures of central Tendency, Arithmetic mean, Median, Mode, Standard deviation, Co efficient of variation (Discrete serious and continuous serious), Hypothesis testing & application, Correlation & regression analysis, Orthogonal array, ANOVA, Standard error, Concept of point and interval estimation, Level of significance, Degree of freedom, Analysis of variance, One way and two way classified data, 'F' test.	**30**	CO1, CO2, CO3, CO4	K1, K3, K5
Module 4:	Preparation of Dissertation and Research Papers, Tables and illustrations, Guidelines for writing the abstract, introduction, methodology, results and discussion, conclusion sections of a manuscript. References, Citation and listing system of documents. Intellectual property rights (IPR) patents copyrights Trademarks Industrial design	**30**	CO2, CO3, CO4	K3, K4,

	geographical indication. Ethics of Research Scientific Misconduct Forms of Scientific Misconduct. Plagiarism, Unscientific practices in thesis work, Ethics in science.			
Pedagogy:	Lectures/tutorials/ assignments/ mini-projects/ PPT presentations/ case studies/ class discussions.			
Texts:	1. Bordens, K. S., & Abbott, B. B. (2002). Research design and methods: A process approach. McGraw-Hill. 2. Kothari, C. R. (2004). Research methodology: Methods and techniques. New Age International.			
References/ Readings:	1. Douglas C. M., &George C. R.(2007). Applied Statistics & probabilityfor Engineers, 3rd edition, Wiley. 2. Robert P. M., Peter S. M., & Mark A. L. (2012) Intellectual Property in New TechnologicalAge. Aspen Law & Business; 6th edition 3. Shirore C. (2015). A Beginners Guide to Latex 4. Sekaran, U., & Bougie, R. (2020). *Research methods for business: A skill-building approach* (8th ed.). Wiley.			
Web Resources:	1. Trochim, W. M. (2006). *Research methods knowledge base*. Atomic Dog Publishing. Retrieved October 9, 2025, from https://socialresearchmethods.net/kb/ 2. Johnson, A. (n.d.). A beginner's guide to qualitative research. *Research Hub*. https://www.researchhub.com/qualitative-guide 3. SAGE Publications. (n.d.). *SAGE research methods*. Retrieved October 9, 2025, from https://methods.sagepub.com/ 4. American Psychological Association. (2022, March 10). Ethical guidelines for conducting research. *APA Ethics Center*. https://www.apa.org/ethics/research-guidelines			

[Back to Index]

Discipline Specific Vocational Elective (DSVE) Courses

Title of the Course	Internet of Things
Course Code	CSA-6401
Number of Credits	(2T+2P)
Theory/Practical	Theory + Lab
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:	Nil	
Course Objectives:	To introduce the concepts, architectures, and applications of IoT, and to enable students to design and implement IoT solutions by applying programming knowledge to real-world scenarios using sensors, microcontrollers, and communication technologies.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand IoT concepts, architectures, components, and communication protocols.	PSO1, PSO6
	CO2. Develop IoT applications by interfacing sensors, actuators, and microcontrollers.	PSO1, PSO2
	CO3. Implement data acquisition, storage, visualization, and basic analytics for IoT systems.	PSO1, PSO2, PSO5

	CO4. Analyze IoT systems for performance, security, and scalability considerations.		PSO1, PSO2, PSO6	
	CO5. Work collaboratively in teams to design, prototype, and present IoT solutions.		PSO3, PSO4	
	CO6. Apply ethical practices and sustainable approaches while designing IoT solutions for real-world applications.		PSO3, PSO5, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Introduction to IoT Understanding IoT fundamentals, IoT Architecture and Protocols, Various Platforms for IoT, Real-time Examples of IoT, Overview of IoT Components & Communication Technologies, Challenges in IoT	**4**	CO1, CO4	K1, K2, K3, K4
	1.2 Arduino Environment Arduino Uno Architecture, IDE Setup & Writing Arduino Programs, Arduino Libraries, Basics of Embedded C Programming, Interfacing LED, Push Button, Buzzer & LCD	**6**	CO2	K2, K3, K4
	1.3 Sensors & Actuators Overview of Sensors & Actuators, Analog & Digital Sensors, Interfacing Temperature, Humidity, Motion, Light & Gas Sensors, Interfacing Relay Switch & Servo Motor	**4**	CO2, CO3	K2, K3, K4
	1.4 Basic Networking with ESP8266 & Raspberry Pi Basics of Wireless Networking, Introduction to ESP8266 Wi-Fi Module, Wi-Fi Libraries & Web Server, Posting Sensor Data to Web Server, Introduction to Raspberry Pi boards, GPIO interfacing, setting up Raspbian OS and Python environment, sending sensor data to cloud or local server, basic MQTT communication, small IoT project with Raspberry Pi.	**6**	CO2, CO3	K2, K3, K4
Module 2:	2.1 IoT Protocols Wireless Technologies for IoT: IEEE 802.15.4, Zigbee, NFC	**2**	CO1, CO3	K2, K3, K4

	IP Based Protocols for IoT IPv6, CoAP, MQTT			
	2.2 Cloud Platforms for IoT Virtualization Concepts & Cloud Architecture, Cloud Computing: Benefits & Services (SaaS, PaaS, IaaS), Cloud Providers & Offerings, IoT Cloud Platforms & ThingSpeak API, Interfacing ESP8266 with Web Services	**5**	CO3	K2,K3,K4
	2.3 Ethics in IoT Professional Ethics & Social Responsibility, Privacy & Data Protection, Sustainable IoT Practices	**3**	CO6	K4,K5
Module 3: LAB	List of suggested Experiments: 3.1 Arduino and Sensor/Actuator Interfacing ● Blink LED and control buzzer using Arduino. ● Read input from the push button and display it on the serial monitor. ● Display messages on LCD using Arduino. ● Interfacing temperature sensor and reading data. ● Interfacing humidity sensor and reading data. ● Interfacing motion sensor and reading data. ● Interfacing light sensor and reading data. ● Interfacing gas sensor and reading data. ● Interfacing relay switch with Arduino. ● Controlling servo motor using Arduino.	**30**	CO2, CO3	K2, K3, K4, K5
Module 4: LAB+Mini Project	4.1 List of suggested Experiments with ESP8266/ESP32 ● Configure ESP8266 Wi-Fi module and connect to the network. ● Post sensor data to a web server. ● Implement MQTT communication between devices and servers. ● Interface ESP8266 with ThingSpeak cloud platform and create dashboards. 4.2 Mini Project	**30**	CO3, CO6	K2, K3, K4, K5

	• Self-balancing robot • Fire-extinguishing robot • Automated plant watering system • Smoke detector with mobile app notifications • Home security system with motion detection • Writing machine using stepper motors • Smart home automation (lights, fans, appliances) • Line follower robot • Obstacle-avoiding robot • Smart door lock with fingerprint/RFID • Automated pet feeder • IoT-based weather monitoring system • Voice-controlled robot or appliance • Smart irrigation system with soil moisture sensing • Image recognition robot or surveillance system			
Pedagogy:	Lectures/tutorials/ assignments/ mini-projects/ PPT presentations/ case studies/ class discussions.			
Texts:	1. Chaouchi, H. (Ed.). (2010). *The Internet of Things: Connecting objects to the web*. Wiley.			
References/ Readings:	1. Qureshi, K. N., & Newe, T. (2024). Artificial intelligence of things (AIoT): New standards, technologies and communication systems (1st ed.). CRC Press. 2. Bahga, A., & Madisetti, V. (2014). Internet of Things: A hands-on approach. VPT. 3. Ovidiu, V., & Peter, S. (2017). IoT fundamentals: Networking technologies, protocols, and use cases for the Internet of Things. Cisco Press. 4. Greengard, S. (2015). The Internet of Things. MIT Press. 5. Rose, K., Eldridge, S., & Chapin, L. (2015). The Internet of Things: An overview. Internet Society. 6. Halsall, F. (2018). Computer networking and the Internet of Things. Wiley.			
Web Resources:	1. Arduino - Home. https://www.arduino.cc/. Accessed 16 Oct. 2025.			

	2. “Circuits on Tinkercad.” Tinkercad, https://www.tinkercad.com/circuits. Accessed 16 Oct. 2025. 3. Wokwi - World’s Most Advanced ESP32 Simulator. https://wokwi.com/. Accessed 16 Oct. 2025. 4. IoT Analytics - ThingSpeak Internet of Things. https://thingspeak.mathworks.com/. Accessed 16 Oct. 2025.

[Back to Index]

Title of the Course	Cloud Computing
Course Code	CSA-6402
Number of Credits	2T+2P
Theory/Practical	Theory + Lab
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, CSA-5005	
Course Objectives:	To provide a comprehensive understanding of core cloud computing paradigms, architectural strategies, and operational models, enabling students to evaluate, design, secure, and manage scalable, highly available, and cost-efficient solutions across modern cloud infrastructure environments.	
		Mapped to PSO
Course Outcomes:	CO1 To explain the fundamental concepts of cloud computing, differentiating between IaaS, PaaS, and SaaS service models and public, private, and hybrid deployment models.	PSO1, PSO5, PSO6
	CO2 To design modern, resilient, and scalable cloud-native solutions using microservices, event-driven patterns, and multi-region strategies, based on an analysis of monolithic architectures.	PSO1, PSO2, PSO6
	CO3 To manage core cloud infrastructure components, including compute services, storage	PSO1, PSO2

	types, and virtual networking.			
	CO4 To implement fundamental cloud security controls (shared responsibility, zero-trust, and IAM) and appropriate data management strategies using cloud storage, databases, and CDNs.		PSO1, PSO2, PSO3	
	CO5 To Design highly available, scalable cloud solutions using load balancing and auto-scaling, and automate infrastructure deployment by applying Infrastructure as Code (IaC) principles.		PSO1, PSO2, PSO5	
	CO6 To Deploy and scale applications effectively on Platform as a Service (PaaS) environments, integrating databases and leveraging built-in development frameworks.		PSO1, PSO2	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	Overview of Computing Paradigm 1.1 Cloud Computing- Types of Cloud Deployment Models Private, Public, Hybrid, Agency Clouds Cloud Service Models: Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service (SaaS), Anything as a Service(XaaS). 1.2 Introduction to Virtualization, Virtualization and cloud computing, Need of virtualization on cloud, limitations. 1.3 Cloud maturity model, Challenges of monolithic architectures, Strategies for decomposition, Common migration pitfalls and solutions, Analyzing monolith to microservices migration 1.4 Architectural Patterns and Resilience. Event-driven architecture principles, Designing for elasticity and fault tolerance. 1.5 Cloud Regions, Zones, and Global Infrastructure Region and zone selection criteria: Latency, compliance, Multi-region architectures and data replication 1.6 Cloud Infrastructure and Virtualization	**15**	CO1, CO2	K2, K3, K4

	Cloud infrastructure components: Compute (Virtual Machines instances), storage, networking.			
Module 2:	2.1 Docker, Kubernetes basics,Virtual Private Cloud (VPC) and subnets,Software-defined networking (SDN), Serverless computing: Event-triggered design, AWS Lambda, Azure Functions, Google Cloud Functions, 2.2 Cloud Storage and Data Management Types of cloud storage: Object, block, file, Data durability and redundancy, Storage services: AWS S3, Azure Blob Storage, Google Cloud Storage, Data lifecycle management,Content delivery networks (CDNs), Cloud databases 2.3 Cloud Security and Compliance Shared responsibility model, Zero-trust security model, Identity and Access Management (IAM), Encryption and key management, Service Account management, Network Firewalls. 2.4 Scalability, Availability, and Performance Scalability and elasticity, High availability (HA) and fault tolerance, Load balancing and auto-scaling, Monitoring tools: CloudWatch, Prometheus 3.1Configuration Management and Infrastructure as Code Configuration Management, Backup and replication techniques, Infrastructure as Code (IaC): Terraform, CloudFormation. 3.2 Platform as a Service Architecture and components, Application deployment and management, Database integration, Scalability and auto-scaling in PaaS.	**15**	CO3, CO4, CO5, CO6	K2, K3, K4
Module 3:	List of suggested Practicals 4.1 Cloud Infrastructure, Data, and Security Provision a Virtual Machine (VM) and configure it within a Virtual Private Cloud (VPC) subnet, applying necessary Security Group/Network Firewall rules for access.	**30**	CO2, CO3, CO4	K5

	Create a Dockerfile for a simple web application, build the container image, and run it locally, demonstrating image layers and container isolation. Implement and deploy an Event-triggered Serverless Function (e.g., Lambda, Azure Function, or Cloud Function) that responds to an HTTP request or a simple event. Configure a cloud storage bucket, host static assets, apply appropriate durability settings, and integrate with a CDN to test latency improvement. Provision a Managed Cloud Database (relational or NoSQL) and connect a cloud instance to it, demonstrating connection security. Create IAM (Identity and Access Management) roles/users and policies to grant only the minimum required permissions (e.g., read-only access to storage) to a Service Account or user. Set up a Load Balancer and configure an Auto-Scaling Group with High Availability (HA) across multiple zones, defining scaling triggers based on a metric like CPU utilization.			
Module 4:	List of Suggested practicals. 4.1 Configuration, IaC, and PaaS ● Write advanced Terraform configuration (using modules) to provision a complete infrastructure: VPC, Subnets, Load Balancer, two Application VMs, and associated Security Groups/Firewalls. Apply, test connectivity, and destroy the stack. ● Enhance the IaC script to integrate basic Configuration Management (CM) using user-data scripts, ansible or cloud-init to automatically install and configure a web server (e.g., Nginx or Apache) on the provisioned VMs. ● Deploy a multi-component application (e.g., a simple API + managed database) onto a Platform as a Service (PaaS) offering (e.g., Google App Engine, Azure App Services). Configure environment variables and database integration.	**15**	CO2, CO5, CO6	K5, K6

	• Configure Automated Backup and Cross-Region Replication for a managed database or storage bucket.			
	Mini project 1. A mini project to develop Cloud Native applications. 2. A mini project to develop an automated, fault-tolerant web application deployment. 3. A mini project to develop a documented infrastructure-as-code (IaC) template to deploy a multi-tier application environment. 4. Use a Platform as a Service (PaaS) offering to deploy a full-stack application and implement its native auto-scaling features.	**15**		
Pedagogy:	Lectures/tutorials/ assignments/ mini-projects/ PPT presentations/ case studies/ class discussions.			
Texts:	1. Erl, T., Mahmood, Z., & Puttini, R. (2013). Cloud computing: Concepts, technology & architecture. Prentice Hall. 2. Laszewski, T., Arora, K., Farr, E., & Zonooz, P. (2018). Cloud native architectures: Design high-availability and cost-effective applications for the cloud. Packt Publishing.			
References/ Readings:	1. Buyya, R., Broberg, J., & Goscinski, A. M. (2013). Cloud computing: Principles and paradigms. Wiley. 2. Davis, C. (2019). Cloud native patterns: Designing change-tolerant systems. O'Reilly Media. 3. Morris, K. (2016). Infrastructure as code: Dynamic systems for the cloud age. O'Reilly Media.			
Web Resources:	1. AWS. (n.d.). AWS documentation. Retrieved October 8, 2025, from https://docs.aws.amazon.com/ 2. Google Cloud Platform. (n.d.). Google Cloud documentation. Retrieved October 8, 2025, from https://cloud.google.com/docs 3. HashiCorp. (n.d.). Terraform. Retrieved October 8, 2025, from https://developer.hashicorp.com/terraform 4. Microsoft Azure. (n.d.). Azure documentation. Retrieved October 8, 2025, from https://learn.microsoft.com/en-us/azure/?product=popular			

[Back to Index]

Title of the Course	Mobile Application Development
Course Code	CSA-6403
Number of Credits	(2T+2P)
Theory/Practical	Theory + Lab
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:	Basic knowledge of programming and DBMS	
Course Objectives:	To provide students with knowledge and practical skills in mobile application development by introducing frameworks, UI/UX design, backend integration, testing, and deployment for creating secure and optimized mobile apps.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand mobile application development lifecycle, frameworks, and programming basics.	PSO1, PSO5
	CO2. Design user-friendly and accessible mobile app interfaces applying UI/UX principles.	PSO1, PSO2, PSO4
	CO3. Apply backend integration using databases, REST APIs, and cloud services like Firebase.	PSO1, PSO2
	CO4. Analyze app performance, debugging issues, and apply optimization techniques.	PSO2, PSO5

	CO5. Implement security best practices and user authentication in mobile apps.		PSO3, PSO5	
	CO6. Create and deploy feature-rich mobile applications as a mini project.		PSO1, PSO4, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 **Introduction to Mobile App Development**: Overview of mobile vs. web apps, evolution of mobile OS (Android, iOS), app development lifecycle (planning, design, coding, testing, deployment).	**3**	CO1	K2
	1.2 **Development Environments**: Setting up Android Studio, React Native CLI, Flutter SDK, Xcode (overview); project creation and basic configuration.	**2**	CO1	K2, K3
	1.3 **Frameworks & Languages**: React Native, Flutter, Kotlin, Swift; programming basics with JavaScript/TypeScript & Dart; when to use each framework.	**3**	CO1	K2, K3
	1.4 **Version Control**: Git & GitHub for collaborative app development, commits, branching, merging, pull requests.	**2**	CO1	K3
	1.5 **Responsive Design Principles**: Mobile-first approach, use of layouts (Flexbox, Grid, ConstraintLayout), ensuring compatibility across devices.	**2**	CO1, CO2	K2, K3
	UI/UX Foundations: Introduction to user interface vs. user experience, consistency, simplicity, wireframing basics.	**3**	CO2	K2, K3
Module 2:	2.1 **UI Components & Interaction**: Designing with widgets (buttons, text fields, sliders, lists, images); navigation (menus, tab bars, navigation drawers); gesture handling.	**4**	CO2	K3
	2.2 **Enhancing UI**: Incorporating animations (transitions, fades, button animations), working with icons, colors, and themes; accessibility considerations.	**3**	CO3	K3,K4

	2.3 **Backend Integration**: Databases (SQLite, Firebase), CRUD operations, real-time updates with Firestore, connecting REST APIs & handling JSON data.	**4**	CO3	K3,K4
	Authentication & Security: Firebase Authentication, session management, password storage, common app vulnerabilities, applying secure coding practices.	**2**	CO3, CO5	K3, K4
	Testing & Deployment: Debugging tools in Android Studio/Flutter DevTools, error handling, writing unit & integration tests, app optimization (reducing latency, memory usage), preparing app for Play Store/App Store.	**2**	CO4, CO6	K4, K5
Module 3: LAB	List of suggested experiments 3.1 Practical Exercises in Mobile App Development ● Set up Android Studio, Flutter, and React Native environments. ● Create a "Hello World" mobile app in Flutter and React Native. ● Design a basic login screen with username and password fields. ● Implement navigation between two screens (e.g., Home ↔ Profile). ● Build a simple calculator app using buttons and text fields. ● Develop a to-do list app with add/delete functionality using local state. ● Store and retrieve data from SQLite (CRUD operations). ● Implement Firebase Firestore for real-time data storage and retrieval. ● Connect to a public REST API (e.g., weather API) and display results.	**30**	CO1, CO2, CO3, CO4, CO5	K2, K3, K4, K5
Module 4: LAB+Mini project	4.1 List of suggested experiments ● Add authentication using Firebase Auth (sign-in/sign-up). ● Apply basic animations (fade-in, button press animation). ● Create a themed app with custom colors, icons, and responsive design. ● Test the app using emulator and real device debugging tools.	**30**	CO3, CO4, CO6	K3, K4, K5

	● Optimize app performance (reduce load time, manage memory). ● Prepare an APK file and deploy locally for testing. 4.2 Mini Project Develop and present a mobile application that integrates UI, backend, and deployment concepts. Suggested mini projects: ● Expense tracker app with Firebase backend. ● Event reminder app with push notifications. ● E-commerce demo app with product listing and cart. ● News aggregator app consuming public APIs. ● Chat application with Firebase authentication and real-time database. ● Fitness tracker app with local storage and progress visualization. ● Food delivery clone (UI + dummy API). ● Student attendance management app with CRUD. ● Weather app using REST API + location services. ● Smart home controller app (mock IoT integration). ● Quiz app with multiple categories and scoring system. ● Online notes app with cloud sync. ● Portfolio/resume app with personal branding. ● Travel guide app with maps and POI info. ● Book library app with search and filter features.			
Pedagogy:	Lectures/tutorials/ assignments/ mini-projects/ PPT presentations/ case studies/ class discussions.			
Texts:	1. Griffiths, D. (2017). Head first Android development (2nd ed.). O'Reilly Media. 2. Napoli, M. L. (2019). Beginning Flutter: A hands-on guide to app development (1st ed.). Wiley.			
References/	1. Scott, A. D. (2020). JavaScript everywhere: Building cross-platform applications with GraphQL, React, React			

Readings:	Native, and Electron (1st ed.). O'Reilly Media. 2. Staiano, F. (2022). Designing and prototyping interfaces with Figma (1st ed.). Packt Publishing Limited. 3. Dabit, N. (2019). React Native in action (1st ed.). Manning Publications.
Web Resources:	1. Flutter - Build Apps for Any Screen. //flutter.dev/. Accessed 16 Oct. 2025. 2. React Native · Learn Once, Write Anywhere. https://reactnative.dev/. Accessed 16 Oct. 2025. 3. "Android Mobile App Developer Tools." Android Developers, https://developer.android.com/. Accessed 16 Oct. 2025. 4. "Firebase | Google's Mobile and Web App Development Platform." Firebase, https://firebase.google.com/. Accessed 16 Oct. 2025. 5. "GitHub · Build and Ship Software on a Single, Collaborative Platform." GitHub, 2025, https://github.com/.

[Back to Index]

Title of the Course	Ethical Hacking
Course Code	CSA-6404
Number of Credits	2T+2P
Theory/Practical	Theory + 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 5002, CSA-5004	
Course Objectives:	To understand ethical hacking phases, identifying and mitigating vulnerabilities in networks and applications, performing simulated attacks using tools like Kali Linux and Metasploit, mastering reconnaissance and penetration testing techniques, and learning to document and report findings responsibly to protect systems.	
		Mapped to PSO
Course Outcomes:	CO1. Understand legal and ethical boundaries for security testing and responsible disclosure.	PSO1, PSO3
	CO2. Set up and use penetration-testing labs (Kali, VMs, snapshots).	PSO1, PSO2
	CO3. Perform reconnaissance, scanning, and enumeration on target systems	PSO1, PSO2
	CO4. Exploit common vulnerabilities in network services and web applications in a controlled environment	PSO1, PSO2

	CO5. Use Metasploit and manual exploit techniques responsibly.		PSO1, PSO2, PSO5	
	CO6. Perform basic privilege escalation, post-exploitation simulation, and data exfiltration in lab		PSO1, PSO2, PSO5	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	The importance of security, The various phases involved in hacking, rules of engagement, ethics & law, definitions, authorized testing, disclosure, report templates. Fundamentals: TCP/IP, routing, DNS, ports — review of networking concepts relevant to pentesting. Linux basics & toolchain overview, shells, file permissions, package managers, Kali Linux overview. Reconnaissance & OSINT, passive & active recon, tools (whois, dig, recon-ng, theHarvester). Scanning & enumeration, Nmap, banner grabbing, service enumeration: Null Session, NetBIOS Enumeration, SNMP enumeration. Vulnerability discovery & management, Common Vulnerabilities and Exposures (CVE), Common Platform Enumeration (CPE), scanners (Nessus/OpenVAS), verification & triage. Web app basics & OWASP Top 10, HTTP, sessions, cookies, common web flaws.	**15**	CO1, CO2, CO3, CO4, CO5	K1, K2, K3 K4
Module 2:	Web app testing techniques, Burp Suite basics, parameter tampering, XSS, SQLi,	**15**	CO4,	K1, K2,

	Command injection, Upload vulnerabilities, CSRF. Exploitation basics & Metasploit, exploit modules, payloads, listeners, safe use, post exploitation techniques. Network services exploitation, SMB, FTP, SMTP misconfigurations. Post-exploitation & persistence, lateral movement, credential harvesting, basic forensics awareness. Privilege escalation, Linux/Windows techniques, identifying vectors. Wireless & mobile basics — Wi-Fi attacks (WEP/WPA), Bluetooth overview, mobile threat considerations. Reporting, remediation & capstone prep, executive and technical report structure, re-testing.		CO5, CO6, CO7	K3, K4
Module 3: LAB	Lab 1 — Lab setup & Kali basics: Install Kali in VM, snapshots, networking modes; create vulnerable VM images. Lab 2 — OSINT & reconnaissance: Use recon-ng, theHarvester, Google dorking, subdomain discovery; produce target profile. Lab 3 — Nmap & port/service discovery: Advanced Nmap scans, NSE scripts, fingerprinting, output parsing. Lab 4 — Vulnerability scanning & verification: Run OpenVAS/Nessus/Nikto; analyze findings and triage false positives.	**30**	CO3, CO4 CO5	K1, K2, K3, K4, K5

	Lab 5 — Web app pentest I: Burp & XSS: Intercept and modify requests; find and exploit reflected/stored XSS on DVWA/Juice Shop. Lab 6 — Web app pentest II: SQLi & auth flaws: Manual SQLi techniques, blind SQLi, exploit flawed authentication flows.			
Module 4: LAB	● Lab 7 — Exploitation with Metasploit: Use Metasploit to exploit a known vulnerable service and use meterpreter safely. ● ● Lab 8 — Network services exploits & pivoting: Exploit SMB/FTP misconfigurations; pivoting/tunneling to internal hosts. ● ● Lab 9 — Privilege escalation & persistence: Local privilege escalation on Linux/Windows vulnerable VMs; persistence and cleanup. ● ● Lab 10 — Wireless lab: Isolated AP lab demonstrating weak WPA/WEP cracking in a controlled environment (with authorization). ● ● Lab 11 — Capture The Flag practice: TryHackMe/HTB style rooms to practice full recon, exploit, root chain. ● ● Lab 12 — Final capstone: Controlled mini-engagement: full pen test in lab, written report and presentation.	**30**	CO1, CO2, CO3, CO4, CO5, CO6	K1, K2, K3, K4, K5
Pedagogy:	Lectures/tutorials/ class assignments/ presentations/ case studies/ class discussions/lab demos/lab assignments			
Texts:	1. Weidman, G. (2014). Penetration testing: A hands-on introduction to hacking. No Starch Press. 2. Stuttard, D., & Pinto, M. (2011). The web application hacker's handbook (2nd ed.). Wiley Publishing. 3. Erickson, J. (2008). Hacking: The art of exploitation (2nd ed.). No Starch Press.			

Web Resources:	1. OWASP Top Ten | OWASP Foundation. https://owasp.org/www-project-top-ten/. Accessed 13 Oct. 2025. 2. MITRE ATT&CK®. https://attack.mitre.org/. Accessed 13 Oct. 2025. 3. "Kali Docs | Kali Linux Documentation." Kali Linux, https://www.kali.org/docs/. Accessed 13 Oct. 2025. 4. TryHackMe | Cyber Security Training. https://tryhackme.com/. Accessed 13 Oct. 2025. 5. "Hack The Box: The #1 Cybersecurity Performance Center." Hack The Box, https://www.hackthebox.com. Accessed 13 Oct. 2025. 6. Download Burp Suite Community Edition - PortSwigger. https://portswigger.net. Accessed 13 Oct. 2025. 7. Technical guide to information security testing and assessment. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-115.pdf. Accessed 13 Oct. 2025.

[Back to Index]

Title of the Course	AI for Software Development
Course Code	CSA-6405
Number of Credits	2T+2P
Theory/Practical	Theory+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-5009	
Course Objectives:	To equip students with the foundational knowledge and practical skills required to strategically integrate, utilize, and govern Generative AI (specifically Large Language Models) into all phases of the Software Development Lifecycle (SDLC), enabling the efficient, ethical, and secure development of modern software systems.	
Course Outcomes:		**Mapped to PSO**
	CO1. To propose viable business cases for integrating Generative AI tools and LLM APIs across all phases of the Software Development Lifecycle (SDLC).	PSO1,PSO5
	CO2. To Apply advanced prompt engineering techniques to reliably steer Large Language Models, including generating structured data outputs.	PSO2,PSO4,PSO6
	CO3. To Synthesize core SDLC artifacts, including user stories, acceptance criteria, code, refactored structures, and high-coverage test scripts, using Large Language Models..	PSO1,PSO2

	CO4 To define the core components of an Autonomous AI Agent , and design a multi-step agent architecture capable of solving complex problems.		PSO1,PSO2,PSO6	
	CO5 To Understand technical and procedural mitigation strategies to govern AI usage, based on an evaluation of ethical implications like bias, fairness, and the "Human-in-the-Loop" principle.		PSO3,PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	Foundations, Generative AI, and Human Integration 1.1 Introduction to AI in SDLC Defining AI, ML, and Deep Learning; The business case for AI in software development; Overview of the AI-enhanced SDLC. 1.2 Core Concepts of Generative AI & API Interaction The transformer architecture (overview); Introduction to LLMs; Differentiating foundation models, fine-tuning, and prompt tuning; Accessing and utilizing external AI models via APIs. 1.3 Prompt Engineering Fundamentals Principles of effective prompting (Role, Task, Context, Format); Zero-shot, One-shot, and Few-shot prompting; Using constraints and structured output (JSON, XML) for reliable API responses. 1.4 AI-Driven Architectural & Design AI assistance in selecting technology stacks; Generating UML diagrams (Class, Sequence) from specifications; Evaluating architectural patterns using AI analysis; Introduction to Agentic Architecture. 1.5 AI Governance, Ethics, and Human Oversight The Human Factor: Human-in-the-Loop (HIL) principles and mandatory	**15**	CO1, CO2	K2, K3, K4

	human oversight;			
Module 2:	AI-Enhanced Coding, Testing, and Quality Assurance 2.1 AI for Code Generation and Refactoring, Generating boilerplate code and function stubs; Code completion and contextual suggestions; Using LLMs for code translation between languages; AI-powered code review suggestions. 2.2 Advanced Testing with Generative AI Automated unit test generation (e.g., Jest, JUnit); Generating comprehensive test data; Generating integration and end-to-end test scripts (e.g., Selenium, Playwright). 2.3 AI for Debugging and Analysis Using LLMs to interpret stack traces and error messages; Root cause analysis assistance; AI-powered log analysis and anomaly detection; Static code analysis enhancements (identifying code smells and complexity) 2.5 AI Agents Defining autonomous AI agents; Agent architecture (Planning, Memory, Tool Use); Introduction to frameworks for building agents; The future role of the software developer in an agent-driven world	**15**	CO3, CO4	K3, K4, K5
Module 3:	List of suggested Practical 3.1 Foundations, Prompting ● Setup a development environment and successfully make an authenticated API call to an external LLM. ● Design prompts to generate project requirements (e.g., 10 user stories) strictly in JSON format using an LLM's response schema feature. ● Use an LLM to generate a basic UML Sequence Diagram and a suggested cloud infrastructure using a text-to-diagram language (e.g., Mermaid). ● Review generated artifacts for biases and use an LLM to generate mitigation strategies for two identified risks.	**30**	CO2, CO3, CO4, CO5	K5

	3.2 API-Driven Development and Quality Assurance ● Take procedural code and use an LLM API to refactor it into an object-oriented or functional structure, preserving functionality. ● For the refactored code from 2.1, use an LLM API call to generate a comprehensive set of unit tests achieving >90% code coverage. ● Using LLMs, generate a dataset of 50 synthetic user profiles, including specific edge cases, for stress-testing application logic.			
Module 4:	4.1 AI Integration, Agents, and Operationalization ● Implement a lightweight proxy server (an "LLM Gateway") that logs request/response metadata (timestamp, token count, latency) before forwarding to an LLM API. ● Design a conceptual AI Agent capable of multi-step problem solving. Output the detailed, step-by-step Reasoning Trace and the simulated Tool Calls in a structured JSON format. ● Take a long, unstructured text (e.g., an email) and use the LLM's response schema/JSON output feature to accurately extract and format specific fields for a hypothetical API.	**30**	CO2, CO3, CO4, CO5	K5, K6
	4.2 Mini project Concepts or tools learnt in the course to be implemented /incorporated. Document the AI driven development process.			
Pedagogy:	Lectures/tutorials/ assignments/ PPT presentations/ case studies/ class discussions.			
Texts:	1. Berryman, J., & Ziegler, A. (2025). Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications. O'Reilly Media. 2. Russell, S. J., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson. 3. Gehring, J., Kundzich, O., & Johnson, P. (2024). AI for Mass-Scale Code Refactoring and Analysis. O'Reilly Media. 4. Morgan, J. (2025). Coding with AI. Manning Publications. 5. Taulli, T. (2024). AI-Assisted Programming. O'Reilly Media.			

	6. Winteringham, M. (2024). Software Testing with Generative AI. Manning Publications.
References/ Readings:	1. McKinsey & Company. (2023). How an AI-enabled software product development life cycle will fuel innovation. Retrieved from [Insert URL if available] 2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS).
Web Resources:	1. Google. (n.d.). Gemini API documentation. Retrieved October 8, 2025, from https://ai.google.dev/gemini-api/docs 2. LangChain Documentation. (n.d.). Prompt templates and chains. Retrieved October 8, 2025, from https://www.langchain.com/docs 3. Mermaid JS. (n.d.). Mermaid documentation. Retrieved October 8, 2025, from https://mermaid.js.org/ 4. OpenAI. (n.d.). API reference: Introduction. Retrieved October 8, 2025, from https://platform.openai.com/docs/api-reference/introduction

[Back to Index]

Title of the Course	Parallel Programming and Scientific Computing
Course Code	CSA-6406
Number of Credits	2T+ 2P
Theory/Practical	Theory + Lab
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-5003, CSA-5006, CSA-5201	
Course Objectives:	This course introduces the fundamentals of parallelism and its importance in accelerating scientific computations. It enables students to design and implement efficient parallel algorithms using frameworks like MPI, OpenMP, and CUDA. Learners will apply these techniques to solve complex numerical and data-intensive scientific problems..	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand concepts, models, and architectures of parallel and distributed computing.	PSO1, PSO6
	CO2. Design and implement parallel algorithms for computational problems.	PSO1, PSO2
	CO3. Apply scientific computing methods for solving large-scale numerical and real-world problems.	PSO1, PSO2, PSO5, PSO4
	CO4. Analyze the scalability and efficiency of parallel solutions.	PSO2, PSO3

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	**Introduction to Parallel Computing** Introduction to Parallel Computing – Need, benefits, and challenges; Parallel vs. Serial computation. Parallel Architecture – Flynn's Taxonomy, Shared and Distributed memory, GPU and Multicore systems **Parallel Programming Models** – Data, Task, Pipeline parallelism, Shared-memory (OpenMP) and Message-passing (MPI) models. Synchronization and Communication – Threads, barriers, race conditions, message passing **Performance Metrics** – Speedup, efficiency, scalability, Amdahl's and Gustafson's Laws. Parallel Programming Tools – OpenMP, MPI basics	**15**	CO1, CO2	K2, K3, K4
Module 2:	**Overview of Scientific Computing:** – Definition, characteristics, and role in research. Numerical Linear Algebra – Matrix operations, solving linear systems (Gauss elimination, LU decomposition) **Numerical Methods** – Integration, differentiation, ODE solving (Euler, Runge-Kutta). Parallelization of Numerical Algorithms – Matrix operations, FFT, Monte Carlo methods **Parallel Libraries and Tools** – BLAS, LAPACK, ScaLAPACK, CUDA basics. Case Studies – Parallel simulations in data analysis and optimization	**15**	CO3, CO4	K2, K3, K4
Module 3 Lab:	**C/C++/Python with OpenMP** Exp 1: Installation and Setup of Parallel Programming Environments – OpenMP, MPI, and CUDA. Exp 2: Divide a large array among multiple threads and compute total sum using reduction Exp 3: Implement matrix multiplication using OpenMP parallel loops and compare with serial version.	**30**	CO1, CO2, CO3, CO4	K1, K3, K5

	Exp 4: Implement a parallel version of the trapezoidal rule for numerical integration Exp 5: Implement a parallel program that finds all prime numbers up to a given integer N. Exp 6: Implement sequential and a parallel version of any sorting technique. Quantify the speedup achieved by the parallel version.			
Module 4 Lab:	**MPI / CUDA programming** Exp 7: Count the number of words in a large text file using multiple threads or processes. Exp 8: Write an MPI program for point-to-point communication (Send/Receive). Exp 9: Implement a distributed program using MPI to perform a vector-scalar multiplication followed by a vector dot product. Exp 10: Divide a matrix among MPI processes and perform transpose. Exp 11: Add two large vectors element-wise using CUDA kernels to leverage GPU parallelism. Exp 12: Multiply two matrices using CUDA kernels. Show the speedup over the serial CPU version and verify correctness.	**30**	CO2, CO3, CO4	K3, K4,
Pedagogy:	Lectures/tutorials/ assignments/ mini-projects/ PPT presentations/ case studies/ class discussions.			

Texts:	1. Grama, A. (2008). *An Introduction to Parallel Computing: Design and Analysis of Algorithms, 2/e*. Pearson Education India. 2. Heath, M. T. (2002). *Scientific Computing: An Introductory Survey*. McGraw-Hill. *NY, USA*. 3. Pacheco, P., & Malensek, M. (2021). *An introduction to parallel programming*. Morgan Kaufmann.
References/ Readings:	1. Weinzierl, T. (2021). *Principles of parallel scientific computing: A first guide to numerical concepts and programming methods*. Springer. 2. Storti, D., & Yurtoglu, M. (2015). *CUDA for engineers: An introduction to high-performance parallel computing*. Addison-Wesley.
Web Resources:	1. OpenMP Architecture Review Board. (n.d.). *Tutorials and articles.* OpenMP. Retrieved October 8, 2025, from https://www.openmp.org/resources/tutorials-articles/ 2. Lawrence Livermore National Laboratory. (n.d.). *OpenMP tutorial*. HPC Tutorials. Retrieved October 8, 2025, from https://hpc-tutorials.llnl.gov/openmp/ 3. Carleton University Research Computing Services. (n.d.). *Introduction to OpenMP*. Carleton University. Retrieved October 8, 2025, from https://carleton.ca/rcs/rcdc/introduction-to-openmp/ 4. Princeton University Research Computing. (n.d.). *OpenMP and MPI learning resources.* Princeton University. Retrieved October 8, 2025, from https://researchcomputing.princeton.edu/education/external-online-resources/openmp

[Back to Index]

Title of the Course	Functional Programming
Course Code	CSA-6407
Number of Credits	(2T+2P)
Theory/Practical	Theory + Lab
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, CSA-5005	
Course Objectives:	1. To understand the foundations and principles of functional programming 2. To develop recursive, modular, and type-safe programs using higher-order functions. 3. To apply algebraic data types, polymorphism, and abstraction effectively. 4. To explore lazy evaluation, monads, and testing for robust functional software.	
Course Outcomes:		**Mapped to PSO**
	CO1. To understand the foundational principles of functional programming.	PSO 1, PSO 3, PSO 6
	CO2. To apply functional programming constructs to solve problems.	PSO 1, PSO 2, PSO 6
	CO3. To analyze and design recursive and algebraic data structures.	PSO 1, PSO 2, PSO 5, PSO 6
	CO4. To evaluate the efficiency and modularity of functional programs.	PSO 1, PSO 2, PSO 5, PSO 6

	CO5. To create functional programs using lazy evaluation and monads.		PSO 1, PSO 2, PSO 4, PSO 5, PSO 6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Introduction to Functional Programming: Declarative vs. imperative paradigms Expressions, evaluation, and immutability Functions as first-class citizens Higher-order functions Function composition and currying Pure functions and referential transparency	**15**	CO 1 CO 2 CO 3 CO 4	K2, K3, K5
	1.2 Compound Data Types and Abstraction Lists, tuples, and user-defined types Pattern matching and guards Recursive data types (lists, trees) Abstraction and code reuse using higher-order functions			
	1.3 Algebraic Data Types and Polymorphism Product and sum types Defining custom data structures using ADTs Type inference and generic functions Parametric and ad-hoc polymorphism Introduction to type classes and instances			
	1.4 Recursion and Program Efficiency Structural and generative recursion Base cases and recursive cases Tail recursion and accumulator patterns			

	Recursive algorithms on lists and trees Time complexity analysis in functional programs			
Module 2:	2.1 Modules and Abstract Data Types Modular design principles Creating and importing modules Abstract data types (ADTs) and encapsulation Interface vs. implementation separation Reusable functional components	**15**	CO 4 CO 5	K2, K3, K4, K5, K6
	2.2 Lazy Evaluation and Infinite Structures Lazy vs eager evaluation Thunks and deferred computation Streams and infinite lists Advantages and challenges of lazy evaluation Performance implications			
	2.3 Monads and Functional I/O Motivation for monads (handling side effects) Monad laws: unit, bind, and associativity Maybe, List, and IO monads do notation and sequencing computations Monads for state and error handling			
Module 3:	Suggested LAB Assignments 3.1 Installation and set up of the FP environment, explore REPL, write first "Hello World" and simple arithmetic expressions.	**30**	CO 1 CO 2	K1, K2, K3, K4
	3.2 Write higher-order functions; use functions as arguments and return values; practice function composition and currying.			
	3.3 Implement simple recursive functions: factorial, Fibonacci, sum/product			

	of list elements Implement tail-recursive functions; practice accumulator-based recursion; compare recursion vs iteration			
Module 4:	4.1 Define lists, tuples, records; implement recursive structures like trees and linked lists; pattern matching exercises	**30**	CO 2 CO 3 CO 4 CO 5	K3, K4, K5, K6
	4.2 Create ADTs for shapes, expressions; define polymorphic functions for generic lists; practice type classes			
	4.3 Create modules with reusable functions; implement abstract data types (ADTs); import and use modules.			
	4.4 Implement infinite sequences; use take, drop, filter on lazy lists; observe memory and evaluation behavior			
	4.5 Implement Maybe and List monads; write simple programs using IO monad; sequence computations with do notation			
	4.6 Write unit tests for pure functions; property-based testing (QuickCheck); verify correctness and edge cases.			
	4.7 Design and implement a small functional application (e.g., text analyzer, expression evaluator, symbolic computation tool); integrate recursion, ADTs, lazy evaluation, and monads.			
Pedagogy:	Lectures/ Tutorials/ Assignments/ Projects/ Case study			
Texts:	1. Pilquist, M., Bjarnason, R., & Chiusano, P. (2023). *Functional programming in Scala* (2nd ed.) [eBook]. ManningPublications.			
References/ Readings:	1. Bhattacharyya, M. (2014). *Functional programming patterns in Scala and Clojure: Write lean programs for the JVM*. Pragmatic Bookshelf. 2. Scalfani, C. (2021). *Functional programming made easier: A step-by-step guide*. Leanpub.			
Web Resources:	1. "Functional Programming Principles in Scala." Coursera, https://www.coursera.org/learn/scala-functional-programming. Accessed 10 Oct. 2025.			

	2. Functional Programming Tutorial. https://www.tutorialspoint.com/functional_programming/index.htm. Accessed 10 Oct. 2025. 3. "Learn Functional Programming in Python [Full Course]." Boot.Dev, https://www.boot.dev/courses/learn-functional-programming-python. Accessed 10 Oct. 2025. 4. "Intro to Functional Programming Basics." freeCodeCamp.Org, 17 Feb. 2020, https://www.freecodecamp.org/news/intro-to-functional-programming-basics/.

[Back to Index]

SEMESTER IV

Generic Elective (GE) Courses

Title of the Course	Personality Development and Professional Ethics
Course Code	CSA-6201
Number of Credits	4
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:	Nil	
Course Objectives:	This course focuses on balancing personal growth, workplace readiness, and ethical considerations in computing so as to enhance holistic development of students and improve their employability skills.	
Course Outcomes:		**Mapped to PSO**
	CO1. To enable the discovery of authentic self and explore ways in which to change, grow, and achieve one's creative potential.	PSO4

	CO2. To develop professionals with emotional intelligence, and effective communication.		PSO4	
	CO3. To prepare students for professional challenges in computing and IT industries.		PSO4	
	CO4. To instill ethical reasoning and decision-making in the context of technology.		PSO6	
	CO5 To foster teamwork, leadership, and responsible digital citizenship		PSO4	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Self-Development & Soft Skills**			
	Self Analysis -Self awareness, SWOT Analysis, Importance of Self Confidence, Self Esteem. Goal Setting -Setting and Achieving Goals, Time Management Creativity - Out of box thinking, Lateral Thinking. Attitude - Factors influencing Attitude, Challenges and lessons from Attitude, Body Language, Etiquette. Personality And Career Choice -Matching your career and personality, Why it matters, Self-efficacy, Changing Your Personality – individuality, adaptability, controlling emotions Motivation - Factors of motivation, Self-talk, Intrinsic & Extrinsic Motivators. Stress Management- Causes of Stress, Impact of Stress, Managing Stress Work-life Balance techniques Communication -Introduction to Communication, Flow of Communication, Listening, Barriers of Communication, how to overcome barriers of communication	**15**	CO1, CO2, CO5	K5,K6
Module 2:	**Professional Skills**			
	Leadership -Introduction to Leadership, Leadership Power, Leadership Styles, Leadership in Administration. Teamwork- Building collaborative skills, problem-solving in groups	**15**	CO3, CO4, CO5	K5,K6

	Critical Thinking- analyze issues, make decisions and solve problems and articulate effectively in a professional setting Workplace Readiness - Resume writing, group discussions, interview techniques Professionalism - Work ethics, punctuality, discipline, accountability Adaptability&Lifelong Learning- Coping with rapid changes in technology Successful Branding - Digital Professionalism**:** Online presence, networking (LinkedIn, GitHub), elevator pitch			
Module 3:	**Professional Growth and Responsibility**	**15**	CO3, CO4, CO5	K5,K6
	Resilient Leadership - cultivating a growth mindset, fostering a supportive environment,maintaining a long-term vision practical strategies for turning setbacks into opportunities, strengthening communication, building strong relationships, and ensuring the well-being of yourself and your team Entrepreneurial Mindset -Startups, innovation, risk-taking, Global & Cultural Awareness -Diversity in tech, cross-cultural communication Corporate Social Responsibility - Tech companies' role in society Sustainability In IT- Green computing, e-waste management, sustainability in software development and usage Remote Work Ethics Effective Feedback Techniques- effective mechanisms to give, take and act on the feedback			
Module 4	**Ethics and Ethical Issues in Computing**			
	Introduction to Professional Ethics: Need and scope in IT Ethical Theories: Utilitarianism, Deontology, Virtue ethics (applied to computing) Professional Codes of Conduct: IEEE, ACM guidelines Workplace Ethics: Harassment, discrimination, confidentiality, whistleblowing	**15**	CO4, CO5	K5,K6

	Social Impact of Computing: Automation, AI, privacy, cybersecurity Intellectual Property Rights: Software piracy, plagiarism, open-source ethics Data Ethics: Data ownership, consent, ethical AI, algorithmic bias Cyber Ethics: Hacking, phishing, responsible use of internet Emerging Technologies: Ethics in AI, robotics, blockchain, genetic computing Famous ethical dilemmas in IT industry - Issues like personal privacy, access rights, harmful actions, patents, copyrights, trade secrets, liability, and piracy,personal privacy Case Studies: Facebook/Cambridge Analytica (data privacy), Google/Project Nightingale (data ethics vs. user privacy), Volkswagen's emissions scandal (corporate responsibility), and dilemmas in autonomous vehicles (AI decision-making in accidents) and AI bias			
Pedagogy:	**Classes need a multi-faceted approach to personal development combining theory, personal experience, and self-reflection** ● Lectures/ tutorials/assignments/class presentations and debates/peer reviews, Group discussions, presentations, role-play ● **Project/Portfolio:** Personal development plan, ethical issue research project Weekly Planner To do list, Reflection journals. Reflection at the end of semester - writing about your reactions to these experiences will be an important part in the pedagogy.			
Texts:	1) Goleman, D. (1995). Emotional intelligence. Bantam Books. 2) Govindarajan, M., Natarajan, S., & Senthil Kumar, V. S. (2013). Professional ethics and human values. PHI Learning.			
References/ Readings:	1) Martin, M. W., & Schinzinger, R. (2005). *Ethics in engineering* (4th ed.). McGraw-Hill. 2) Gotterbarn, D., & Miller, K. W. (1997). Software engineering ethics. Prentice Hall. 3) Covey, S. R. (1989). The 7 habits of highly effective people. Simon & Schuster. 4) Harris, C. E., Pritchard, M. S., & Rabins, M. J. (2018). Engineering ethics: Concepts and cases (6th ed.). Cengage Learning.			

	5) Johnson, D. G. (2009). Computer ethics (4th ed.). Pearson Education. 6) Baase, S., & Henry, T. (2018). A gift of fire: Social, legal, and ethical issues for computing and the internet (5th ed.). Pearson. 7) Ghillyer, A. (2023). Business ethics: Now and the future (5th ed.). McGraw-Hill Education. 8) Batra, R. (2017). Personal growth and training & development. Deep & Deep Publications. 9) Sanghi, S. (2012). Improve your communication skills (2nd ed.). Response Books. 10) Burnard, P. (2005). Interpersonal skills training: A source book of activities. Routledge. 11) Alex, K. (2010). Soft skills: Know yourself & know the world. S. Chand Publishing. 12) Sunitha, V. (2016). Personality development & communicative English. Neelkamal Publications. 13) Maxwell, J. C. (1993). Developing the leader within you. Thomas Nelson.

[Back to Index]

Title of the Course	Computer Security
Course Code	CSA-6202
Number of Credits	(4T)
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:	Nil	
Course Objectives:	To gain understanding of risk and threats faced by Computer Systems, learn how vital, indispensable business data and information can be compromised, lost, corrupted or be prone to unauthorized access. Understand controls and safeguards to be implemented to protect data and loss of privacy.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand the various threats, vulnerabilities, and attacks.	PSO1, PSO2
	CO2. Analyze security risks, conduct risk assessment.	PSO1, PSO2
	CO3. Implement security policies and security controls	PSO1, PSO2, PSO6
	CO4. Understand ethical and legal implications.	PSO1, PSO3, PSO6

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	Information Systems, Type of Information Systems, Definition of Computer Security, The Challenges of Computer Security, Model for Computer Security, Threats and Attacks, Security Functional Requirements, OSI Security Architecture: Security Attacks, Security Services, Security Mechanism. Computer Security Strategy. Basic Cryptographic Concepts: Symmetric and Public Key Encryption, Confidentiality using symmetric encryption, Message Authentication, Digital Signatures & Non Repudiation, Digital Certificates, Importance of Key Management**.** User Authentication: Password based User Authentication, Password Selection and Management, Token Based and Biometric Authentication, Security issues for Password Authentication. Types of Malicious Software (Malware), Propagation**n**, Infected Content, Viruses, Worms, SPAM E-mail, Trojans, Attack Agent—Zombie, Bots, Keyloggers, Phishing, Spyware, Backdoors, Rootkits, Countermeasures	**15**	CO1, CO2	K1, K2, K3
Module 2:	Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Application-Based Bandwidth Attacks, Reflector and Amplifier Attacks, Defences Against Denial-of-Service Attacks, Responding to a Denial-of-Service Attack Intrusion Detection and Prevention Systems: Intruder, Host based Intrusion Detection, Distributed Host-Based Intrusion Detection, Network-Based Intrusion Detection, Distributed Adaptive Intrusion Detection, Honeypots,	**15**	CO1, CO2	K1, K2, K3

	Firewalls, The Need for Firewalls, Firewall Characteristics, Types of Firewalls, Firewall Basing, Firewall Location and Configurations Intrusion Prevention Systems Access Control, Access Control Principles: Authentication, Authorization, Audit; Access Control Policies: Discretionary Access Control, Mandatory Access Control, Role Based Access Control.			
Module 3:	Buffer Overflow, Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow Attacks Trusted Computing and Multilevel Security, The Bell LaPadula Model, Other Formal Models for Computer Security, The Concept of Trusted Systems, Application of Multilevel Security, Trusted Computing and the Trusted Platform Module, Common Criteria for Information Technology Security Evaluation, Assurance and Evaluation IT Security Management, Organizational Context and Security Policy, Security Risk Assessment, Detailed Security Risk Analysis IT Security Controls, Plans, and Procedures, IT Security Management Implementation, Security Controls or Safeguards, IT Security Plan, Implementation of Controls, Implementation Follow-up.	**15**	CO1, CO2 CO3	K1, K2, K3, K4, K5

Module 4:	Physical Security Threats, Physical Security Prevention and Mitigation Measures, Recovery from Security Breaches Human Factors, Security Awareness, Training and Education, Organizational Security Policy, Employment Practices and Policies, Email and Internet use policies Security Audits, Security Audit Architecture, Audit Trail, Logging Functions, Audit Trail Analysis Legal and Ethical Aspects, Cybercrime and Computer Crime, Intellectual Property, Privacy, Ethical Issues	**15**	CO1, CO2, CO3 CO4	K1, K2, K3, K4
Pedagogy:	Lectures/tutorials/ assignments/ presentations/ case studies/ class discussions.			
Texts:	1. Stallings, W., & Brown, L. (2010). Computer security: Principles and practice. Pearson Education.			
References/ Readings:	1. Easttom, C. (2014). Network defenses and countermeasures: Principles and practices. Pearson Education. 2. Forouzan, B. A. (2006). Data communication and networking. Tata McGraw-Hill Education. 3. Forouzan, B. A. (2010). Cryptography & network security. Tata McGraw-Hill Education.			

[Back to Index]

Title of the Course	Ethical Use of AI
Course Code	CSA-6203
Number of Credits	4
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 5009	
Course Objectives:	To equip students with the critical analytical skills necessary to identify, articulate, and propose solutions for the ethical, social, and governance challenges presented by the design, deployment, and regulation of Artificial Intelligence (AI) systems across various domains.	
Course Outcomes:		**Mapped to PSO**
	CO1. Critically analyze and apply fundamental ethical theories and contemporary AI ethical principles to real-world AI systems and case studies.	PSO3, PSO4
	CO2. Identify the sources of algorithmic bias and unfairness across the AI lifecycle and propose technical and policy interventions for mitigation.	PSO1, PSO2, PSO3
	CO3. Evaluate the broader social, political, and economic impacts of AI systems, including issues of surveillance capitalism, labor displacement, environmental costs, and the	PSO3, PSO6

	concentration of power.			
	CO4. Contrast and critique different national and international approaches to AI governance, including major regulatory efforts and proposals for accountability and liability in autonomous systems.		PSO3, PSO6	
	CO5. Discern the unique ethical and legal challenges posed by AI in high-stakes sectors and articulate principles for responsible deployment in these domains.		PSO1, PSO3	
	CO6. Analyze the ethical implications of emerging AI technologies, such as Generative AI and deepfakes, and the long-term philosophical challenges related to AI safety, control, and alignment of machine goals with human values.		PSO3, PSO5, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Defining AI and Classical Ethics Defining AI/ML, The nature of moral problems in technology, Utilitarianism, Deontology (Kantian imperatives), Virtue Ethics. 1.2 Applied Ethical Principles & Alignment Core principles: Fairness, Accountability, Transparency (FAT), Non-Maleficence, The problem of Value Alignment, Human control and autonomy. 1.3 The Nature of Algorithmic Decision-Making Introduction to predictive vs. generative models, The concept of the "Black Box," Opacity and justification in high-stakes systems. 1.4 Ethics of Generative Content: Deepfakes & AI Art Deepfakes: Non-consensual imagery, political manipulation, erosion of trust in media. AI Art: Copyright, intellectual property, and the definition of authorship/creativity.	**15**	CO1, CO3	K2, K3

Module 2:	2.1 Sources and Types of Algorithmic Bias Pre-existing bias (Societal, historical data), Technical bias (Sampling, measurement), Emergent bias (Context of use). 2.2 Formal Definitions of Fairness Group Fairness (Demographic Parity), Individual Fairness, Inherent conflicts and trade-offs between different metrics. 2.3 AI and Systemic Injustice Intersectionality and AI, Amplification of existing inequalities, Normative critique in criminal justice, hiring, and welfare. 2.4 Fairness-Aware Design and Auditing Mitigation strategies: re-sampling, post-processing, Inherent limitations of purely technical solutions, The need for social and technical algorithmic audits.	**15**	CO2, CO4	K3, K4
Module 3:	3.1 Explainable AI (XAI) Theory The philosophical importance of explanation and justification, Distinguishing Interpretability (model understanding) from Explainability (human communication). 3.2 The Problem of Responsibility and Liability The "Responsibility Gap" in autonomous systems, Moral agency and personhood in AI, Distributing liability (designer vs. user vs. system). 3.3 Case Study: Consequences of AI in Medicine Consequences of AI in Medicine: Diagnostic errors, patient harm, legal liability for misdiagnosis, data privacy of sensitive health records, algorithmic biases in treatment recommendations . 3.4 Accountability Frameworks and Governance Conceptual models of accountability, Auditing, traceability, and the role of	**15**	CO5, CO4	K3

	documentation (Model Cards, Datasheets) in development. 3.5 AI Safety, Robustness, and Security Defining AI Safety, Adversarial examples and attacks, The theory of robustness against unexpected inputs, Data poisoning attacks.			
Module 4	4.1 Data Ethics and Privacy Theory Concepts of privacy (contextual integrity), Differential Privacy as a technical safeguard, Data ownership and data monopolies. 4.2 AI and Surveillance Capitalism Shoshana Zuboff's framework, The ethics of pervasive tracking, Algorithmic manipulation and effects on human autonomy and agency. 4.3 The Dead Internet Theory and Digital Trust Dead Internet Theory, AI-generated content, and algorithmic curation. Ethical consequence: Erosion of epistemic trust, information whiteout, inability to distinguish human from synthetic interaction, and potential for scaled manipulation. 4.4 AI and Socio-Economic Impacts Automation and technological unemployment debates, Ethical theories of labor rights, The problem of concentrating power in tech giants and financial institutions. 4.5 The Ethics of Autonomous Systems Moral dilemmas for autonomous vehicles (e.g., the Trolley Problem), Ethics of Lethal Autonomous Weapon Systems (LAWS): responsibility and the laws of war.	**15**	CO3, CO6	K3, K5

	4.6 Global AI Governance and Future Ethics Comparison of global regulatory frameworks (e.g., EU AI Act), Existential Risk (x-risk) from Advanced AI, The long-term ethics of superintelligence.			
Pedagogy:	Lectures/tutorials/ assignments/ debates/ PPT presentations/ case studies/ class discussions.			
Texts:	1. Barocas, S., Hardt, M., & Narayanan, A. (2023). Fairness and machine learning: Limitations and opportunities. The MIT Press. 2. Bullock, J. B., Chen, Y.-C., Himmelreich, J., Hudson, V. M., Korinek, A., Young, M. M., & Zhang, B. (Eds.). (2024). The Oxford handbook of AI governance. Oxford University Press. 3. Coeckelbergh, M. (2020). AI ethics. The MIT Press. 4. Zuboff, S. (2019). The age of surveillance capitalism: The fight for a human future at the new frontier of power. Public Affairs.			
References/ Readings:	1. Bender, Emily M., et al. (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT). 2. Mittelstadt, B., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. Big Data & Society, 3(2), 1–21. 3. Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, 4. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 5. Awad, E., Dsouza, S., Kim, R., Schulz, J., Henrich, J., Shariff, A., Bonnefon, J.-F., & Rahwan, I. (2018). The Moral Machine experiment. Nature, 563(7729).			
Web Resources:	1. Algorithmic Justice League (AJL). (n.d.). Algorithmic Justice League. Retrieved October 8, 2025, from https://www.ajl.org/ 2. Future of Life Institute (FLI). (n.d.). Future of Life Institute. Retrieved October 8, 2025, from https://futureoflife.org/ 3. Google. (2018). AI principles. Retrieved October 8, 2025, from https://ai.google/principles/			

	4. Stanford University Human-Centered AI Institute (HAI). (n.d.). Stanford HAI. Retrieved October 8, 2025, from https://hai.stanford.edu/ 5. The Alan Turing Institute. (n.d.). The Alan Turing Institute. Retrieved October 8, 2025, from https://www.turing.ac.uk/ 6. United Nations Educational, Scientific and Cultural Organization (UNESCO). (2021). Recommendation on the ethics of artificial intelligence. Retrieved October 8, 2025, from https://www.unesco.org/en/artificial-intelligence/recommendation-ethics

[Back to Index]

Title of the Course	Prompt Engineering
Course Code	CSA-6204
Number of Credits	4
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:		
Course Objectives:	To equip learners with the knowledge and skills to design, optimize, and defend prompts for Generative AI applications, enabling effective task-specific generation, reasoning, and multimodal output while ensuring responsible AI usage.	
Course Outcomes:		**Mapped to PSO**
	CO1: Explain the principles, structures, and methodologies of prompt engineering for Generative AI.	PSO1,PSO3,PSO4
	CO2: Develop and refine effective prompts for text, image, and multimodal AI generation tasks.	PSO1,PSO2,PSO3,PSO5, PSO6
	CO3: Apply advanced reasoning-based prompting techniques such as Chain-of-Thought and Tree-of-Thought for complex problem-solving.	PSO1,PSO2,PSO3, PSO5,PSO6

	CO4: Evaluate, defend, and optimize prompt strategies to ensure robustness, reproducibility, and secure AI outputs.		PSO1-PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Prompt Engineering for Generative AI** Introduction to prompt engineering and its significance – Understanding prompt structure and formulation – Basic ingredients of a prompt – Potential complexity of prompts – System prompt and user prompt – Instruction-based prompting – In-context learning: zero-shot and few-shot prompting – Context length and context efficiency – Exploring AI prompting interfaces and tools – Designing effective and context-aware prompts – Persona-based prompting patterns – Principles and best practices for crafting prompts – Industry perspectives on effective prompt design – Emerging applications and future use cases of prompt engineering.	**15**	CO1, CO2	K2, K3
Module 2	**Prompt Engineering – Techniques and Approaches** Overview of prompting methodologies – Text-to-text prompt strategies – Interview pattern approach – Chain-of-Thought technique for logical reasoning – Tree-of-Thought approach for structured problem solving – Self-consistency in reasoning and response generation – Comparative analysis of Chain-of-Thought and Tree-of-Thought approaches – Selecting suitable prompting techniques for different tasks – Design considerations for prompt optimization – Developing and refining effective prompts for Generative AI applications.	**15**	CO2, CO3	K2, K3
Module 3	**Text-to-Image Prompting** Fundamentals of text-to-image prompting – Understanding the relationship between textual description and visual generation – Key elements of effective image prompts (subject, style, context, and constraints) – Techniques for creating accurate and expressive text-to-image prompts – Balancing creativity and control in visual prompt design – Evaluating and refining generated	**15**	CO2	K3,K4

	outputs – Integrating prompt engineering principles into multimodal AI systems.			
Module 4	**Prompt Engineering Best Practices and Defensive Prompting** **Prompt Engineering Best Practices:** Write clear and explicit instructions – Provide sufficient context – Break complex tasks into simpler subtasks – Give the model time to think – Iterate and refine your prompts – Evaluate prompt engineering tools – Organize and version prompts for reproducibility and efficiency. **Defensive Prompt Engineering:** Understanding proprietary prompts and reverse prompt engineering – Jailbreaking and prompt injection – Information extraction risks – Defenses against prompt attacks – Consolidation of key concepts in prompt engineering – Application of best practices in prompt formulation – Designing comprehensive prompts for complex AI tasks.	**15**	CO4	K5,K6
Pedagogy:	Teaching/lab hands on/ presentation /Mini project			
Texts:	1.Alammar, J. (2023). Hands-on large language models. O'Reilly Media. 2. Rothman, D. (2021). Transformers for natural language processing. Packt Publishing.			
References/ Readings:	1. Huyen, C. (2024). AI engineering: Building applications with foundation models. O'Reilly Media			
Web Resources:	1. OpenAI Platform. https://platform.openai.com. Accessed 14 Oct. 2025. 2. Introduction | 🦜 🔗 LangChain. https://python.langchain.com/docs/introduction/. Accessed 14 Oct. 2025. 3. Alammar, Jay. The Illustrated Transformer. https://jalammar.github.io/illustrated-transformer/. Accessed 14 Oct. 2025..			

[Back to Index]

Title of the Course	Spatial Data Applications
Course Code	CSA-6205
Number of Credits	4
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:	Nil	
Course Objectives:	To develop expertise in spatial data concepts, GIS, and geospatial analytics by acquiring, storing, processing, and visualizing spatial data; applying spatial analysis techniques to solve real-world problems; and integrating geospatial tools and APIs to create interactive applications.	
Course Outcomes:		**Mapped to PSO**
	CO1. Explain spatial data types, models, GIS components	PSO1, PSO2,
	CO2. Acquire, preprocess, visualize spatial data	PSO1, PSO2,
	CO3. Execute spatial queries & geoprocessing	PSO2
	CO4. Develop interactive spatial applications	PSO2, PSO5
	CO5. Evaluate spatial data applications for usability & societal relevance	PSO3, PSO6

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	1.1 Introduction: Raster/vector, coordinate systems, GIS components 1.2Spatial Data Acquisition: GPS, remote sensing, metadata	**15**	CO1, CO2	K2,K3
Module 2:	2.1 Spatial Data Management: PostGIS, spatial indexing, topology 2.2 Spatial Analysis: Overlay, buffer, network analysis	**15**	CO3	K3,K4
Module 3:	3.1 Cartography & Visualization: Thematic maps, dashboards 3.2 Web Mapping & APIs: Leaflet.js, Mapbox, GeoJSON	**15**	CO2, CO4	K3, K4,
Module 4:	4.1 Applications & Case Studies: Smart cities, environment, Bhuvan 4.2 Emerging Trends: AI/ML, drones, ethics, privacy	**15**	CO5	K6
Pedagogy:	Interactive lectures, hands-on exercises, mini-projects, Case studies			
Texts:	1. Burrough, P. A., & McDonnell, R. A. (2015). Principles of Geographical Information Systems (4th ed.). Oxford University Press. 2. Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). Geographical Information Systems and Science (4th ed.). Wiley. 3. Chang, K. T. (2016). Introduction to Geographic Information Systems (8th ed.). McGraw-Hill Education. 4. Obe, R. O., & Hsu, L. S. (2015). PostGIS in Action (2nd ed.). Manning Publications.			
References/ Readings:	1. Tomlinson, R. F. (2013). Thinking About GIS: Geographic Information System Planning for Managers (2nd ed.). ESRI Press. 2. Indian Space Research Organisation (ISRO) & National Remote Sensing Centre (NRSC). (2020). National Geospatial Data Infrastructure and Applications. ISRO & NRSC Publications.			
Web Resources:	1. https://www.qgistutorials.com/. Accessed 14 Oct. 2025. 2. OpenStreetMap Wiki. https://wiki.openstreetmap.org/. Accessed 14 Oct. 2025. 3. Mapbox | Maps, Navigation, Search, and Data. https://www.mapbox.com. Accessed 14 Oct. 2025. 4. Earth Engine. https://earthengine.google.com. Accessed 14 Oct. 2025.			

[Back to Index]

Title of the Course	Educational Technology
Course Code	CSA-6206
Number of Credits	4
Theory/Practical	Theory
Level	500
Effective from AY	2025 -26
New Course	Yes
Bridge Course/Value-added Course	No
Course for advanced learners	Yes

Pre-requisites for the Course:	Basic knowledge of computer applications, programming, and instructional design concepts	
Course Objectives:	The students will be able to: 1. Understand the theoretical foundations of educational technology and instructional design. 2. Analyze and evaluate different digital learning environments and tools. 3. Apply learning theories to design and develop technology-supported instructional materials. 4. Implement emerging technologies like AI, AR/VR, and mobile learning in educational contexts. 5. Conduct research and evaluation studies on the effectiveness of EdTech systems.	
Course Outcomes:		**Mapped to PSO**
	CO1. Explain key concepts, models, and theories in educational technology	PSO1, PSO6
	CO2. Analyze instructional design models and apply them to digital learning	PSO1, PSO2

	CO3. Develop an interactive digital learning application or module		PSO1, PSO2, PSO5	
	CO4. Evaluate educational tools using usability and learning analytics		PSO1, PSO2, PSO6	
	CO5. Explore the role of AI, gamification, and immersive technologies in learning		PSO3, PSO4	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Foundations of Educational Technology** ● Definition, scope, and evolution of educational technology ● Systems approach to education and learning ● Learning theories: Behaviorism, Cognitivism, Constructivism, and Connectivism ● Pedagogical, andragogical, and heutagogical models of learning ● ICT in education and its societal impact	**15**	CO1, CO3 CO4,	K1, K2, K3, K4
	Instructional Design and Learning Models ● ADDIE, ASSURE, and Dick & Carey models ● Learning objectives and outcomes formulation (Bloom's Taxonomy) ● Multimedia learning principles (Mayer's Principles) ● Designing digital content: text, audio, video, and interactivity ● Storyboarding and scripting for e-learning			
Module 2:	**Digital Learning Environments and Tools** ● Learning Management Systems (LMS): Moodle, Canvas, Google	**15**	CO2, CO3	K2, K3, K4, K5

	Classroom ● MOOCs and open educational resources (OERs) ● Virtual classrooms, webinars, and collaborative tools ● Mobile learning (m-learning) and microlearning ● Accessibility, usability, and inclusivity in learning technology			
Module 3:	**Emerging Technologies in Education** ● Artificial Intelligence and Machine Learning in Education ● Adaptive and personalized learning systems ● Augmented Reality (AR) and Virtual Reality (VR) applications ● Gamification and game-based learning ● Learning analytics and data-driven education	**15**	CO3, CO6	K2, K3, K4, K5
Module 4:	**Evaluation, Research, and Ethics in Educational Technology** ● Evaluation of instructional effectiveness ● Usability testing and learner analytics ● Research methodologies in educational technology ● Ethical, privacy, and legal issues in EdTech ● Future trends and challenges	**15**	CO3, CO6	K2, K3, K4, K5
Pedagogy:	Lectures/tutorials/ assignments/ mini-project/presentations/ case studies/ class discussions. **Assignments/Project-work recommended for Practice** Students will complete **one major project** and **two minor lab exercises**, such as: 1. Design and develop a small e-learning module using an LMS or authoring tool.			

	2. Evaluate an existing educational app using usability and learning analytics metrics. 3. Develop a prototype AI-powered or gamified learning system. 4. Present a research paper review on a current EdTech trend.
Texts:	1. Heinich, R., Molenda, M., Russell, J. D., & Smaldino, S. E. (2002). Instructional media and technologies for learning (7th ed.). Pearson. 2. Reiser, R. A., & Dempsey, J. V. (2018). Trends and issues in instructional design and technology (4th ed.). Pearson. 3. Bates, A. W. (2019). Teaching in a digital age: Guidelines for designing teaching and learning (2nd ed.). Tony Bates Associates. https://pressbooks.bccampus.ca/teachinginadigitalagev2
References/ Readings:	1. Clark, R. C., & Mayer, R. E. (2016). E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning (4th ed.). Wiley. 2. Horton, W. (2011). Designing web-based training: How to teach anyone anything anywhere anytime (2nd ed.). Wiley. 3. Ally, M. (Ed.). (2009). Mobile learning: Transforming the delivery of education and training. Athabasca University Press. 4. Siemens, G. (2005). Connectivism: A learning theory for the digital age. International Journal of Instructional Technology and Distance Learning, 2(1), 3–10. http://www.itdl.org/Journal/Jan_05/article01.htm
Web Resources:	1. Educational Technology - Course. https://onlinecourses.swayam2.ac.in/cec19_ed08/preview. Accessed 14 Oct. 2025. 2. "Best Educational Technology Courses & Certificates [2025] | Coursera Learn Online." Coursera, https://www.coursera.org/courses?query=educational%20technology. Accessed 14 Oct. 2025. **Software Tools/Platforms Recommended for Practice** ● Moodle / Google Classroom ● Articulate Storyline / Adobe Captivate ● Scratch / Unity (for gamified learning)

	• Power BI / Tableau (for learning analytics) • ChatGPT / Google Gemini / OpenAI API (for AI-integrated learning)

[Back to Index]

Title of the Course	Visual Modeling and 3-D Printing
Course Code	CSA-6207
Number of Credits	4
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:	Nil	
Course Objectives:	To introduce students to additive manufacturing concepts, 3D modeling, and printing technologies, enabling them to design, prepare, and produce 3D objects using CAD tools and various 3D printing processes while understanding materials, processes, and post-processing techniques.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand additive manufacturing concepts, CAD modeling, and reverse engineering for 3D printing	PSO1, PSO2
	CO2. Prepare 3D printing data, including STL file creation, repair, slicing, support generation, and toolpath planning.	PSO1, PSO2
	CO3. Operate and utilize 3D printing software such as Fusion 360, SolidWorks, Tinkercad, Ultimaker Cura, and Repetier Host.	PSO1, PSO4

	CO4 Identify and select appropriate materials for 3D printing, understanding their properties, preparation, and applications.		PSO1, PSO2, PSO3	
	CO5 Analyze and evaluate solid, liquid, and powder-based 3D printing processes, including FDM, SLA, SLS, DED, and Binder Jetting.		PSO1, PSO5, PSO6	
	CO6 Design and implement small-scale 3D printing projects, applying knowledge of modeling, process selection, and post-processing while considering ethical and practical aspects.		PSO1, PSO4, PSO5, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	1.1 Introduction to Additive Manufacturing and CAD Additive vs. Conventional Manufacturing processes, Overview of 3D printing processes, Computer Aided Design and file formats, Process chain for 3D Printing, Reverse Engineering for 3D Printing.	**15**	CO1	K2, K3
Module 2:	2.1 Data Preparation and 3D Printing Software 3D Printing interfaces, STL interface specification, Creating and repairing STL files, STL data manipulation, Part orientation, support generation, Model slicing, Direct and adaptive slicing, Tool path generation. 3D printers overview, Accuracy, Precision, Tolerance, 3D printing software (Fusion 360, SolidWorks, Onshape, Tinkercad, Ultimaker Cura, MeshLab, Simplify3D, Repetier Host) – operation and usage of any one.	**15**	CO2, CO3	K2, K3, K4
Module 3:	3.1 3D Printing Materials and Solid/Liquid/Powder-based Processes Forms of 3D printing raw materials: Liquid, Solid, Wire, Powder. Popular materials for FDM, SLA, SLS, Binder Jetting, Material Jetting, Direct Energy Deposition. Polymers, Metals, Non-Metals, Ceramics. Principles, construction, working, post-processing, troubleshooting, applications, advanced concepts, and future trends of Solid, Liquid, and Powder-based 3D printing processes (FDM, SLA, SLS, DED, Binder Jetting).	**15**	CO4, CO5	K2, K3, K4, K5

Module 4:	4.1 Applications, Case Studies and Project Work Case studies of industrial and research 3D printing applications, Comparative analysis of 3D printing technologies, Mini-project implementation integrating design, material selection, printing process, and post-processing. Discussion on ethical and sustainability considerations.	**15**	CO5, CO6	K3, K4, K5
Pedagogy:	Lectures/ tutorials/assignments/class presentations and debates.			
Texts:	1. Soloman, S. (2018). 3D printing and design. Khanna Publishing House.			
References/ Readings:	1. Gebhardt, A. (2011). Understanding additive manufacturing: Rapid prototyping, rapid tooling, rapid manufacturing. Hanser Publisher. 2. Chua, C. K., & Leong, K. F. (2017). 3D printing and rapid prototyping: Principles and applications. World Scientific. 3. Kloski, L. W., & Kloski, N. (2021). Getting started with 3D printing: A hands-on guide to the hardware, software, and services behind the new manufacturing revolution (2nd ed.). Make Community, LLC. 4. Pradhan, S. K., Nayak, A., Thakur, S. S., Francis, V., & Nagargoje, A. (2025). Fundamentals and applications of additive manufacturing: Hardware, software, methods, materials and future trends (1st ed.). CRC Press.			
Web Resources:	1. Onshape. (n.d.). Onshape: Cloud-based CAD platform for collaborative 3D modeling and product design. PTC. https://www.onshape.com/en/ (Accessed October 13, 2025) 2. Tinkercad. (n.d.). Tinkercad: Free online 3D modeling and CAD tool for beginners. https://www.tinkercad.com/ (Accessed October 13, 2025) 3. Ultimaker. (n.d.). Ultimaker Cura: Free slicing software for 3D printing. https://ultimaker.com/software/ultimaker-cura/ (Accessed October 13, 2025) 4. MeshLab. (n.d.). MeshLab: Open-source software for processing and editing 3D triangular meshes. https://www.meshlab.net/ (Accessed October 13, 2025) 5. Simplify3D. (n.d.). Simplify3D: Professional 3D printing slicing software with advanced control over print settings. https://www.simplify3d.com/ (Accessed October 13, 2025) 6. Repetier Software. (n.d.). Repetier Host: Free 3D printer host software for controlling printers, slicing, and managing print jobs. https://www.repetier.com/ (Accessed October 13, 2025)			

Title of the Course	Digital Transformation
Course Code	CSA-6208
Number of Credits	4
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:	Nil	
Course Objectives:	• Understand the concepts, scope, and importance of Digital Transformation. • Explore emerging technologies such as Cloud Computing, Digital Twin, and IoT. • Analyze how digital connectivity and computing trends shape industries. • Apply digital strategies to enable innovation and organizational growth. • Evaluate real-world digital transformation case studies across domains.	
Course Outcomes:		**Mapped to PSO**
	CO1. Understand the fundamental concepts, needs, and factors driving Digital Transformation.	PSO1, PSO5
	CO2. Understand the role of computing, cloud, and visualization technologies in digital transformation.	PSO1, PSO2

	CO3. Understand the importance of digital connectivity, IoT, and 5G technologies.		PSO1, PSO2	
	CO4. Analyze the significance of the Digital Twin and digital strategy in innovation.		PSO2, PSO4	
	CO5. Evaluate digital maturity and industry trends through case studies.		PSO3, PSO5, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1	**Introduction to Digital Transformation** Introduction - Concept, Need, and Scope, Drivers and Benefits of Digital Transformation, Key Success Factors and Strategy, Digital Maturity and Assessment Models, Industry Case Studies – Automotive, Healthcare, Education, Introduction to the Digital Twin	**15**	CO1, CO4, CO5	K1, K2, K3, K4
Module 2	**Computing and Visualization Technologies** Role of Computing in Digital Transformation, Cloud Computing – Concepts, Models, and Benefits, Edge Computing Overview, Visualization Technologies – AR, VR, MR, Cloud and Visual Computing Use Cases	**15**	CO2, CO4	K2, K3, K4
Module 3	**Connectivity and IoT** Digital Connectivity – Wired and Wireless, Evolution of Cellular Networks (4G, 5G), Internet of Things – Concepts and Applications, Connectivity and Digital Transformation Integration, Industry Examples – Smart Cities, Manufacturing, Retail	**15**	CO3, CO4, CO5	K2, K3, K4

Module 4	**Digital Strategy and Innovation** Digital Transformation Challenges and Best Practices, Developing a Digital Transformation Strategy, Innovation Through Digital Technologies, Measuring Digital Success – KPIs and Metrics, Case Studies: Digital Leaders (Amazon, Tesla, Siemens), Group Activity – Propose a Digital Strategy	**15**	CO4, CO5	K3, K4, K5
Pedagogy:	Lectures, case studies, tutorials, group discussions, mini-projects, and multimedia presentations.			
Texts:	1. Siebel, T. M. (2019). Digital transformation: survive and thrive in an era of mass extinction. RosettaBooks. 2. Westerman, G., Bonnet, D., & McAfee, A. (2014). Leading digital: Turning technology into business transformation. Harvard Business Press.			
References/ Readings:	1. Herbert, L. (2017). Digital transformation: Build your organization's future for the innovation age. Bloomsbury Publishing. 2. Skilton, M., & Hovsepian, F. (2018). The 4th industrial revolution. Springer Nature. 3. Greenway, A., Terrett, B., & Bracken, M. (2021). Digital transformation at scale: Why the strategy is delivery. Do Sustainability. 4. Tao, F., Zhang, M., & Nee, A. Y. C. (2019). Digital twin driven smart manufacturing. Academic press.			
Web Resources:	1. O'Brien, K., Downie, A., & Scapicchio, M. (2025, September 19). What is Digital Transformation?. IBM.https://www.ibm.com/think/topics/digital-transformation Accessed 13 Oct. 2025. 2. "Travelling through Time with the Digital Twin." Inspiring.Siemens,https://www.siemens.com/global/en/company/stories/industry/2025/digital-twin-digital-enterprise-time-travel.html. Accessed 13 Oct. 2025. 3. Michelman, George Westerman and Paul. "Webinar: Digital Transformation After the Pandemic." MIT Sloan Management Review, 11 Jun. 2021, https://sloanreview.mit.edu/video/digital-transformation-after-the-pandemic/.			

[Back to Index]

Title of the Course	eGovernance: Service Design & Delivery
Course Code	CSA-6209
Number of Credits	4
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:	Nil	
Course Objectives:	To understand the importance of E-Governance, to know E-Governance initiatives and understanding E-Governance technologies.	
Course Outcomes:	**CO1:** To understand the concept of EGovernance, its infrastructure and use in public administration	PSO1, PSO2,
	CO2: To apply the concept of eGovernance and technologies in creating a better place for administration	PSO1, PSO2
	CO3: To analyze the impact of different technologies in eGovernance	PSO3, PSO5

	CO4: To design an architecture for eGovernance for better administration		PSO3, PSO4, PSO6	
Content:		**No. of hours**	**Mapped to CO**	**Cognitive level**
Module 1	**EGovernance Introduction** ● Introduction to E-Government and E-Governance ● Difference between E-Government and E-Governance ● E-Government as Information System; ● Benefits of E-Government; ● EGovernment Life Cycle; ● Online Service Delivery and Electronic Service Delivery; ● Evolution, Scope and Content of E-Governance; ● Present Global Trends of Growth in E-Governance; ● Models of E-Governance; ● Introduction Model of Digital Governance: Broadcasting / Wider Dissemination Model, Critical Flow Model, Comparative Analysis Model, Mobilization and Lobbying Model, Interactive – Service Model / Government-to-Citizen-toGovernment Model (G2C2G)	**15**	CO1 CO3	K1 K2
Module 2	**Infrastructure of E Governance and legal initiative** ● Evolution in E-Governance and Maturity Models: Five Maturity Levels; ● Characteristics of Maturity Levels; ● Towards Good Governance through E-Governance Models ; ● National e-Governance Services Delivery Assessment (NeSDA) ● E-Government Infrastructure Development, Network Infrastructure; ● Computing Infrastructure; ● Information Society and Community Empowerment. ● IT Acts and National E-Governance Plan. ● E-Governance Initiatives in India	**15**	CO1 CO2	K2 K3 K4
Module 3	**eGovernance Framework**	**15**	CO2	K1

	● Data centers; ● E-Government Architecture; ● Interoperability Framework ● Cloud Governance; ● E-readiness; ● Data System Infrastructure; ● Legal Infrastructural Preparedness Unit ● Ethics of Law and Technology. ● Data Security and Privacy Concerns. ● Smart Devices, ● Processes and Services		CO3	K2 K3 K5
Module 4	**Methods of EGovernance and EGovernance in public office** ● Institutional Infrastructural Preparedness; ● Human Infrastructural Preparedness; Technological Infrastructural Preparedness ● Security for e-Government, ● Challenges and Approach of E-government Security; ● Security Management Model; ● E- Government Security Architecture ● Database and Human Development. ● National Informatics Centre (NIC) ● Back Office Operations and Front Office Delivery. Business Process Reengineering (BPR).	**15**	CO2 CO3 CO4	K1 K3 K5 K6
Pedagogy:	Lectures/ tutorials /assignments/ class presentations and debates/ peer reviews.			
Texts:	1. Garson, G. D. (2006). Public information technology and e-governance: Managing the virtual state (1. ed). Jones and Bartlett.. 2. Yaren, H. (2020). Implementing blockchain technology in the customs environment to support the safe framework of standards. World Customs Journal, 14(1). https://doi.org/10.55596/001c.116314 3. Park, J. J., & Park, J. J. J. H. (with the International Conference on Information Technology Convergence and Services). (2013). Information technology convergence: Security, robotics, automations and communication.			

	Springer. 4. Das, R. K., & Patra, M. R. (2013). A service oriented design approach for e-governance systems. International Journal of Information Technology Convergence and Services, 3(3), 1–11. https://doi.org/10.5121/ijitcs.2013.3301 5. Tomura, A. (2009). Inaugural cambridge chapter of 85 broads, judge business school, university of cambridge, uk, 23 may 2009. Strategic HR Review, 8(6). https://doi.org/10.1108/shr.2009.37208fab.010 6. Sinha, R. P. (2006). E-governance in India: Initiatives and issues (1. publ). Concept Publ.

[Back to Index]

Title of the Course	Digital Storytelling & Gamification
Course Code	CSA-6210
Number of Credits	4
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:	NIL	
Course Objectives:	To develop skills in digital storytelling, interactive narrative design, gamification application, usability evaluation, and the integration of storytelling with gamification for impactful digital experiences.	
Course Outcomes:		**Mapped to PSO**
	CO1. Explain principles, elements, types of digital storytelling.	PSO1, PSO2
	CO2. Design interactive narratives	PSO1, PSO2
	CO3. Apply gamification frameworks	PSO1, PSO2, PSO3
	CO4. Evaluate solutions for usability, engagement, social impact	PSO3, PSO5
	CO5. Integrate storytelling & gamification	PSO2, PSO5

Content:		No of hours	Mapped to CO	Cognitive Level
Module 1:	Introduction Story Elements: narrative, storyboarding, scripting	**15**	CO 1, CO 2	K2, K3
Module 2:	Tools: Twine, Powtoon, StoryMapJS Gamification Fundamentals	**15**	CO 2, CO 3	K2, K3
Module 3:	Psychology of Engagement: Flow, SDT, player types Designing Gamified Experiences	**15**	CO 3	K4
Module 4:	Integration: Narrative-driven gamification, AR/VR Evaluation & Ethics	**15**	CO 4, CO 5	K5
Pedagogy:	Interactive lectures, hands-on exercises, mini-projects, Case studies			
Texts:	1. Robin, B. R. (2008). *The educational uses of digital storytelling*. Society for Information Technology & Teacher Education (SITE).			
References/ Readings:	1. Werbach, K., & Hunter, D. (2012). *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press. 2. McGonigal, J. (2011). *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. Penguin Press. 3. Zichermann, G., & Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media.			
Web Resources:	1. Twine. Twine – interactive storytelling tool. Retrieved October 13, 2025, from https://twinery.org/ 2. Knight Lab. StoryMapJS – map-based storytelling tool. Retrieved October 13, 2025, from https://storymap.knightlab.com/ 3. Gameful. Gameful – gamification design resources. Retrieved October 13, 2025, from https://gameful.com/ 4. Coursera. Coursera – online courses, tutorials on storytelling and gamification. Retrieved October 13, 2025, from https://www.coursera.org/			

Title of the Course	Natural Language Processing			
Course Code	CSA-6211			
Number of Credits	4			
Theory/Practical	Theory			
Level	500			
Effective from AY	2025-26			
New Course	Yes			
Bridge Course/ Value added Course	No			
Course for advanced learners	Yes			

Pre-requisites for the Course:	Nil			
Course Objectives:	To understand the fundamentals of Natural Language Processing (NLP).			
Course Outcomes:			**Mapped to PSO**	
	CO1: Remember core NLP terminologies.		PSO1, PSO2, PSO3	
	CO2: Understand NLP tasks and processing steps.		PSO1, PSO3, PSO5	
	CO3: Apply NLP techniques in real-world applications.		PSO2, PSO5, PSO6	
	CO4: Evaluate outcomes of NLP-based systems.		PSO2, PSO3, PSO4	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1	1.1 Introduction**:** Definition, Natural Language Understanding, Natural Language Generation, Three generations of NLP, NLP trinity, Corpora and their	**15**	CO1, CO2	K1, K2

	construction, concordance, collocation, regular expressions, Issues and Challenges, NLP applications. 1.2 Word Sense Disambiguation: Lexical knowledge networks, Princeton WordNet, Indian language wordnet, WordNet relations, WordNet applications, Idioms and Metaphors. 1.3 Computational Morphology: Definition, Agglutination, Types of Morphology.			
Module 2	2.1 Shallow Parsing: POS tagging, Chunking, Multi-word expressions, Named entity recognition – techniques, challenges, and applications. 2.2 Deep parsing: Constituency parsing, Statistical parsing, Dependency parsing, Scope ambiguity, Attachment ambiguity, rule-based parsing, and statistical parsing.	**15**	CO1, CO2	K1, K2, K3
Module 3:	3.1 Sentiment Analysis: Ambiguity – lexical, syntactic, semantic, discourse, pragmatic; Lexicons – manual creation, automatic creation; Rule-based – word level, sentence level, document level; Statistical – Naïve Bayes, Support Vector Machine. 3.2 Neural networks for NLP: Review of neural networks basics (Perceptron, Feed forward networks, Back-propagation algorithm). 3.3 Word embeddings: Word2vec, Glove, FastText.	**15**	CO2, CO3	K1, K2, K3, K4
Module 4:	4.1 Modern Trends in NLP: Transformer Architectures: Evolution from RNNs to Transformers; Self-Attention Mechanism; Encoder–Decoder Framework. Pre-trained Language Models (PLMs): Overview of BERT, GPT, RoBERTa, and T5; Transfer Learning and Fine-Tuning for NLP tasks. Prompt-based Learning and Instruction Tuning: Few-shot and zero-shot capabilities in large language models (LLMs).	**15**	CO4	K1, K2, K3, K4, K5, K6

	Applications of Generative NLP: Chatbots, text summarization, machine translation, and creative text generation. 4.2 Suggested Tutorials: ● Tokenization – word, sentence, character, sub-word, using stop words as delimiter ● Stop word removal, Punctuation removal ● Use of Stemmer and Lemmatizer ● Extracting all nouns in a text ● Finding cosine similarity between two texts 4.3 Suggested Mini projects: ● Develop a POS tagger using a statistical technique. ● Implement a morphological analyzer. ● Implement a model to analyze the sentiment of a given text. ● Generate a summary for a given document. ● Implement a Language Detection system for any 4 languages of your choice. ● Implement a Named Entity Recognition system to identify the named entities from a given text. ● Implement a model to identify the multi-word expressions in a given text. ● Implement a model to identify if the given phrase is used in an idiomatic sense or a regular sense.			
Pedagogy:	Lectures/ Assignments/ Flipped Classroom			
Texts:	1. Bhattacharyya, P., & Joshi, A. (2023). Natural language processing. Wiley.			

References/ Readings:	1. Allen, J. (1995). Natural language understanding (2nd ed.). Pearson Education. 2. Jurafsky, D., & Martin, J. H. (n.d.). Speech and language processing (3rd ed., draft). Retrieved October 14, 2025, from https://web.stanford.edu/~jurafsky/slp3/
Web Resources:	1. Bhattacharyya, P. (n.d.). Natural language processing [Course]. NPTEL, IIT Bombay. Retrieved May 16, 2025, from https://nptel.ac.in/courses/106101007 2. IIT Madras. (n.d.). Introduction to natural language processing (i-NLP) [Course]. Retrieved May 16, 2025, from https://study.iitm.ac.in/ds/course_pages/BSCS5002.html 3. Goayal, P. (n.d.). Natural language processing [Course]. NPTEL, IIT Kharagpur. Retrieved May 16, 2025, from https://onlinecourses.nptel.ac.in/noc19_cs56/preview

[Back to Index]

Title of the Course	Computational Linguistics
Course Code	CSA-6212
Number of Credits	4
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:	Proficiency in Python and some exposure to Algorithms and Data structures. Familiarity with basic linguistic principles is helpful.	
Course Objectives:	This course provides a comprehensive introduction to computational linguistics and natural language processing (NLP), the interdisciplinary field that uses computational methods to analyze and generate human language.	
Course Outcomes:	Upon successful completion of this course, students will be able to:	**Mapped to PSO**
	CO1: Explain the core linguistic concepts of phonology, morphology, syntax, and semantics from a computational perspective.	PSO1,PSO3,PSO4
	CO2: Develop programming skills in Python to process and analyze textual data using libraries like NLTK or spaCy.	PSO1,PSO2,PSO3,PSO5,PSO6
	CO3: Apply fundamental algorithms and statistical methods to common NLP problems, such as text classification, part-of-speech tagging, and parsing.	PSO1,PSO2,PSO3,PSO5,PSO6

	● **CO4:** Design and execute a mini project applying computational linguistics methods to a linguistic problem or real-world application.		PSO1-PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Foundations of language and computation** ● Introduction to computational linguistics and NLP: What is CL/NLP? History and major applications (e.g., machine translation, chatbots). Overview of linguistic levels. Review of foundational linguistics**:** Phonetics, phonology, morphology, syntax, and semantics. The Chomsky hierarchy and formal languages. ● Python for computational linguistics**:** Review of Python essentials, data structures, and file handling. Introduction to the Natural Language Toolkit (NLTK).	**15**	CO1, CO2	K2, K3
Module 2	**Text processing and statistical methods** ● Regular expressions and text normalization**:** Tokenization, stemming, and lemmatization. Regular expressions for pattern matching. ● Corpus linguistics**:** Introduction to linguistic corpora. Frequency distributions, collocations, and concordances. ● Language modeling and probability**:** Events and probability. N-gram models and smoothing techniques. ● Text classification and supervised learning**:** Introduction to machine learning. Features for text classification, sentiment analysis.	**15**	CO2, CO3	K2, K3
Module 3	**Syntactic and Semantic Analysis** ● Parts of speech (POS) tagging**:** Rule-based tagging, hidden Markov Models (HMMs), and evaluation. ● Syntactic parsing**:** Context-free grammars, parsing algorithms (e.g., chart parsing).	**15**	CO2	K3,K4

	• Computational semantics**:** Word meaning, word sense disambiguation, and lexical resources like WordNet. • Distributional semantics**:** Semantic similarity, word embeddings (Word2Vec), and neural language models.			
Module 4	**Advanced topics and applications** • Deep learning for NLP**:** Introduction to neural networks, recurrent neural networks (RNNs), and transformers. • Advanced NLP applications**:** Machine translation, speech recognition, and question answering systems. • Ethics and bias in NLP**:** Societal impact of language technology, bias in data and algorithms, fairness.	**15**	CO4	K5,K6
Pedagogy:	Teaching/lab hands on/ presentation /Mini project			
Texts:	• 1. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media. • 2. Bhattacharyya, P. (2022). Natural Language Processing: A Paninian Perspective, PHI			
References/ Readings:	1. Jurafsky, D., & Martin, J. H. (2023). Speech and language processing (3rd ed.). Pearson. 2. Akmajian, A. (2001). Linguistics: An Introduction to Language and Communication. United Kingdom: MIT Press.			
Web Resources:	1. ACL Member Portal | The Association for Computational Linguistics Member Portal. https://www.aclweb.org/portal/. Accessed 14 Oct. 2025. 2.CMU School of Computer Science. 13 Oct. 2025, https://www.cs.cmu.edu/index. 3.Start[Linguisticsweb.Org]. https://linguisticsweb.org/doku.php. Accessed 14 Oct. 2025.			

[Back to Index]

Title of the Course	Software Project Management
Course Code	CSA-6213
Number of Credits	4
Theory/Practical	Theory
Level	500
Effective from AY	2025 -26
New Course	Yes
Bridge Course/Value-added Course	No
Course for advanced learner	Yes

Pre-requisites for the Course:	NIL	
Course Objectives:	By the end of the course, students will be able to: 1. Introduce principles, processes, and challenges of managing software projects. 2. Familiarize students with project estimation, scheduling, risk management, and quality assurance techniques. 3. Develop understanding of project management methodologies (traditional and agile). 4. Equip students with knowledge of tools and metrics for monitoring and controlling software projects. 5. Prepare students for leadership and decision-making roles in software development environments.	
Course Outcomes:		**Mapped to PSO**
	CO1. Explain the principles, lifecycle, and frameworks of software project management.	PSO1, PSO6
	CO2. Apply project estimation and planning techniques to real-world projects.	PSO1, PSO2, PSO6

	CO3. Develop and manage schedules, resources, and risks effectively.		PSO1, PSO2, PSO5, PSO6	
	CO4. Evaluate software quality, productivity, and performance metrics.		PSO1, PSO2, PSO6	
	CO5. Analyze and apply agile and modern project management methodologies.		PSO3, PSO4, PSO6	
Content:		**No of hours**	**Mapped to CO**	**Cognitive Level**
Module 1:	**Introduction to Software Project Management, Project Planning and Estimation** ● Definition, characteristics, and objectives of software project management ● Software vs. conventional project management ● Project life cycle and process models (Waterfall, Iterative, Spiral, Agile) ● Stakeholders and organizational structures in software projects ● Role and responsibilities of a software project manager ● Overview of PMBOK knowledge areas and SDLC alignment ● *Project planning objectives and activities* ● *Software project estimation: size, effort, cost, and schedule* ● *Estimation techniques: Function Point Analysis, COCOMO models, Use Case Points* ● *Work Breakdown Structure (WBS) and activity definition*	**15**	CO1, CO4	K1, K2, K3, K4

	● *Resource allocation and team organization* ● *Project budgeting and cost management* **Learning Outcome:** Understand the scope, roles, and context of managing software projects. Gain the ability to estimate and plan project activities, timelines, and costs systematically.			
Module 2:	**Scheduling, Tracking, and Risk Management** ● Project scheduling using Gantt Charts, PERT, and CPM ● Network diagrams and critical path analysis ● Resource leveling and allocation optimization ● Project tracking and performance measurement (Earned Value Analysis) ● Risk management process: identification, assessment, mitigation, and contingency planning ● Change management and configuration control **Learning Outcome:** Apply scheduling, monitoring, and risk management techniques for effective control.	**15**	CO2, CO3	K2, K3, K4, K5
Module 3:	**Software Quality Management and Metrics** ● Software quality assurance and standards (ISO 9001, CMMI, Six Sigma) ● Software configuration management (SCM) and version control ● Defect prevention and detection techniques ● Project metrics: size, cost, effort, productivity, defect density, and	**15**	CO3, CO6	K2, K3, K4, K5

	reliability ● Process improvement models: PSP, TSP, and QIP ● Reviews, audits, and testing as quality management tools **Learning Outcome:** Understand how to ensure and measure quality throughout the project lifecycle.			
Module 4:	**Agile, Leadership, and Emerging Practices** ● Agile methodologies: Scrum, XP, Kanban, Lean Software Development ● Agile roles, ceremonies, and artifacts ● Hybrid project management and DevOps integration ● Leadership styles and team motivation in software projects ● Communication management and stakeholder engagement ● Project closure and post-implementation review ● Emerging trends: AI-assisted project management, cloud collaboration tools **Learning Outcome:** Acquire knowledge of modern project management approaches and leadership skills for dynamic environments.	**15**	CO3, CO6	K2, K3, K4, K5
Pedagogy:	Lectures, case studies, and group discussions Analysis of real-world project case studies Demonstration of project management tools (MS Project, Jira, Trello, or Asana) Guest lectures from industry professionals			
Texts:	1. Hughes, B., & Cotterell, M. (2017). *Software Project Management*. McGraw-Hill Education. 2. Pressman, R. S. (2020). *Software Engineering: A Practitioner's Approach*. McGraw-Hill. 3. Schwalbe, K. (2022). *Information Technology Project Management*. Cengage Learning.			
References/ Readings:	1. Sommerville, I. (2016). *Software Engineering*. Pearson. 2. Jalote, P. (2012). *Software Project Management in Practice*. Pearson. 3. Royce, W. (1998). *Software Project Management: A Unified Framework*. Addison-Wesley.			

	4. Highsmith, J. (2010). *Agile Project Management: Creating Innovative Products*. Addison-Wesley. 5. PMBOK Guide (Latest Edition) – Project Management Institute (PMI).
Web Resources:	1. "Software Project Management (SPM) - Software Engineering." GeeksforGeeks, 18 Apr. 2019, https://www.geeksforgeeks.org/software-engineering/software-engineering-software-project-management-spm/ 2. *Sof*tware Project Management. https://www.tutorialspoint.com/software_engineering/software_project_management.htm. Accessed 15 Oct. 2025. 3. Software Project Management - Course by Swayam https://onlinecourses.nptel.ac.in/noc19_cs70/preview

[Back to Index]