SRI VASAVI ENGINEERING COLLEGE (**AUTONOMOUS**)
(Sponsored by Sri Vasavi Educational Society)
(Approved by AICTE, New Delhi & Permanently affiliated to JNTUK, Kakinada)
(Accredited by NAAC with 'A' Grade ,Recognized by UGC under section 2(f) & 12(B))
Pedatadepalli, **TADEPALLIGUDEM – 534 101.** W.G.Dist**. (A.P)**
Department of Computer Science & Engineering (Accredited by NBA)
and
Department of Computer Science & Technology

Minutes of the 8th Board of Studies held on 21/07/2025 at 02.00 PM through online mode

Date: 24.07.2025

The 8th Meeting of Board of Studies in Department of Computer Science and Engineering is held at 02:00 PM on 19-07-2025 through online mode using,

https://us06web.zoom.us/j/83489443842

The following members attended the meeting:

S.No.	Name of the Member	Designation	Role
1.	Dr. D Jaya Kumari	Professor, HoD-CSE, SVEC	Chairperson
2.	Dr.Krishna Mohan Ankala	Professor, UCEK, Kakinada	University Nominee
3.	Dr. R.B.V. Subramanyam	Professor, Department of CSE, NIT Warangal	Academic Expert
4.	Sri. Srinivasa Raju Vuppalapati	Senior Consultant, MSR IT Services LLP, Hitech City, Hyderabad.	Industry Expert
5.	Mr.EEdala Rambabu	Amadues, Bangalore	Alumni
6.	Dr. V Venkateswara Rao	Professor	Member
7.	Dr. V S Naresh	Professor	Member
8.	Dr. K Shirin Bhanu	Professor	Member
9.	Dr. A Daveed Raju	Professor	Member
10.	Mr. G Sriram Ganesh	Sr. Assistant Professor	Member
11.	N. Hiranmayee	Sr. Assistant Professor	Member
12.	K Lakshmi Narayana	Assistant Professor	Member
13.	Mr. M.S.Kumar Reddy	Assistant Professor	Member
14.	M Santhi	Assistant Professor	Member
15.	M N V Surekha	Assistant Professor	Member
16.	Mr. P. Ramamohan Rao	Assistant Professor	Member
17.	Mr. T. Anil Kumar Reddy	Assistant Professor	Member
18.	Mr. K. Phanindra Brahmaji	Lecturer	Member
19.	Mr. S. P. Ramesh Varma	Lecturer	Member
20.	Mrs. A Neelima	Lecturer	Member

The following are the Minutes of the Meeting

Item No.1: Welcome note by the Chairperson BOS.

The Chairperson extended a warm welcome to all members and introduced them formally. A brief progress report of the department was also presented.

Item No.2: Review of the Course Structure I to IV Semesters and Approval of Course Structure for V and VI Semesters of B.Tech(CSE) and B.Tech(CST) Programme under V23 Regulation.

The course structures for the I to VI semesters of both B.Tech (CSE) and B.Tech (CST) programmes were reviewed and approved. The details are provided in **Annexure-I**.

Item No.3: Approval of Syllabi for the Proposed Courses offered in V and VI Semesters of B.Tech(CSE) and B.Tech(CST) Programme under V23 Regulation and also Approval of Professional Elective course of B.Tech(ECE) & B.Tech(ECT)

The syllabi for courses in the V and VI semesters were approved with the following modifications and suggestions:

SEM	Course Name	Suggestions	Modifications
V	Finite Languages and Automata Theory (V23CST11)	Merge with Compiler design (VI SEM)	Renamed as *Automata and Compiler Design (ACD)and offered in V SEM*
V	Artificial Intelligence	Include "AI Agent Communication Protocols" in Unit-V under Expert Systems: • Model Context Protocol (MCP) • Agent-to-Agent Protocol (A2A) • Agent Communication Protocol (ACP)	Incorporated as suggested
VI	Machine Learning (V23CST12)	Move from Professional Elective-II to Professional Core Course	Implemented accordingly
		Add Transformers Concept in Machine Learning	It is being added to the Deep Learning Course in VII Semester
VI	Mobile Adhoc Networks	Shift from Professional Elective-IV to Professional Elective-III	Change effected
VI	Cryptography & Network Security Lab	Replace with Machine Learning Lab	Modified as proposed

The details are provided in **Annexure-II**.

Item No. 4: Approval of list of Courses offering under Job Oriented Elective-I to Job Oriented Elective-IV in V to VII Semesters respectively under V23 Regulation for all other branches and the approval of their Syllabi.

The list of **Job Oriented Elective** courses and their syllabi for V to VII semesters were approved. The details are provided in **Annexure-III.**

Item No. 5: Approval of list of Courses offering under Open Elective-I to Open Elective-IV in V to VII Semesters respectively under V23 Regulation for all other branches and the approval of their Syllabi.

The list of Open Elective courses and their syllabi for V to VII semesters were approved. The details are provided in **Annexure-IV.**

Item No. 6: Approval of B.Tech(Hons) & B.Tech(Minors) offered by CSE.

The proposed B.Tech (Honours) and B.Tech (Minors) programs were reviewed and approved. The details are provided in **Annexure-V.**

Item No. 7: Approval of Course Structure and Syllabi for I to IV Semesters of M.Tech(CS) Programme under V25 Regulation.

Approved the Course Structure and Syllabi for I to IV Semesters of M.Tech(CS) Programme under V25 Regulation. Details are given in **Annexure-VI.**

Chairperson of BOS
(Dr.D Jaya Kumari)

Head of the Department
Dept. of Computer Science & Engineering
Sri Vasavi Engineering College
TADEPALLIGUDEM-534 101

Annexure-I

SEMESTER – I

S.No.	Course Code	Name of the Course		L	T	P	C
1	V23CMT01	Basic Civil & Mechanical Engineering	ESC	3	0	0	3
2	V23CHT01	Applied Chemistry	BS&H	3	0	0	3
3	V23MAT01	Linear Algebra & Calculus	BS&H	3	0	0	3
4	V23CST01	Introduction to Programming	ESC	3	0	0	3
5	V23MET01	Engineering Graphics	ESC	1	0	4	3
6	V23CSL02	IT Workshop Lab	ESC	0	0	2	1
7	V23MEL01	Engineering Workshop Lab	ESC	0	0	3	1.5
8	V23CHL01	Applied Chemistry Lab	BS&H	0	0	2	1
9	V23CSL01	Computer Programming Lab	ESC	0	0	3	1.5
10	V23SPT01	NSS/NCC/Scouts & Guides/Community Service	BS&H	-	-	1	0.5
11	-	English Certificate Course	-	2	0	0	0
			Total	**15**	**0**	**15**	**20.5**

SEMESTER – II

S.No.	Course Code	Name of the Course		L	T	P	C
1	V23MAT02	Differential Equations & Vector Calculus	BS&H	3	0	0	3
2	V23PHT01	Engineering Physics	BS&H	3	0	0	3
3	V23ENT01	Communicative English	BS&H	2	0	0	2
4	V23EET01	Basic Electrical and Electronics Engineering	ESC	3	0	0	3
5	V23CST02	Data Structures	PCC	3	0	0	3
6	V23PHL01	Engineering Physics Lab	BS&H	0	0	2	1
7	V23ENL01	Communicative English Lab	BS&H	0	0	2	1
8	V23EEL01	Electrical and Electronics Engineering Workshop Lab	ESC	0	0	3	1.5
9	V23CSL03	Data Structures Lab	PCC	0	0	3	1.5
10	V23SPT02	Health and wellness, Yoga and sports	BS&H	-	-	1	0.5
			Total	14	0	11	19.5

SEMESTER-III

S.No.	Course Code	Name of the Course		L	T	P	C
1	V23MAT05	Discrete Mathematics & Graph Theory	BS&H	3	0	0	3
2	V23MBT51	Managerial Economics and Financial Analysis	Management Course- I	2	0	0	2
3	V23CST03	Digital Logic & Computer Organization	ESC	3	0	0	3
4	V23CST04	Advanced Data Structures & Algorithm Analysis	PCC	3	0	0	3
5	V23CST05	Object Oriented Programming Through Java	PCC	3	0	0	3
6	V23CSL04	Advanced Data Structures and Algorithm Analysis Lab	PCC	0	0	3	1.5
7	V23CSL05	Object Oriented Programming Through Java Lab	PCC	0	0	3	1.5
8	V23CSSE01	Python Programming Lab	SEC	0	1	2	2
9	V23MET09	Design Thinking & Innovation	BS&H	1	0	2	2
10	V23ENT02	Professional Communication Skills - I	Audit Course	2	0	0	0
			Total	**17**	**1**	**10**	**21**

SEMESTER - IV

S.No.	Course Code	Name of the Course		L	T	P	C
1	V23MBT53	Universal Human Values	BS&H	2	1	0	3
2	V23MAT07	Probability & Statistics	BS&H	3	0	0	3
3	V23CST06	Operating Systems	PCC	3	0	0	3
4	V23CST07	Database Management Systems	PCC	3	0	0	3
5	V23CST08	Software Engineering	PCC	2	1	0	3
6	V23CSL06	Operating Systems Lab	PCC	0	0	3	1.5
7	V23CSL07	Database Management Systems Lab	PCC	0	0	3	1.5
8	V23CSSE02	Full Stack Development –I	SEC	0	1	2	2
9	V23CEAC01	Environmental Science	Audit Course	2	0	0	-
10	V23ENT03	Professional Communication Skills - II	Audit Course	0	2	0	0
			Total	**15**	**5**	**8**	**20**
*Mandatory **Community Service Project (V23COSP01)** Internship of 08 weeks duration during summer vacation*							
		Minor Course (Student may select from the specialized minors pool)	MC	3	0	3	
		Honors Course (Student may select from the honors pool)	HC	3	0	3	

SEMESTER-V

S.No.	Course Code	Name of the Course		L	T	P	C
1	V23CST09	Data Warehousing and Data Mining	PCC	3	0	0	3
2	V23CST10	Computer Networks	PCC	3	0	0	3
3	V23CST11	Automata and Compiler Design	PCC	3	0	0	3
4	**Professional Elective-I**		PEC	3	0	0	3
	V23CSTPE01	1. Object Oriented Analysis and Design					
	V23CSTPE02	2. Artificial Intelligence					
	V23CSTPE03	3. Microprocessors & Microcontrollers					
	V23CSTPE04	4. Quantum Computing					
		5. 12 week MOOC Swayam/NPTEL course recommended by the BoS					
5	**Open Elective -I / Job Oriented Elective -I**		OEC / JOE	3	0	0	3
		Offline / 12 week MOOC Swayam/NPTEL course recommended by the BoS					
		Entrepreneurship Development & Venture Creation					
6	V23CSL08	Data Mining Lab	PCC	0	0	3	1.5
7	V23CSL09	Computer Networks Lab	PCC	0	0	3	1.5
8	V23CSSE03	Full Stack development-2	SEC	0	1	2	2
9	**Engineering Science Course**		ESC	0	0	2	1
	V23CSES01	Master Coding and Competitive Programming - Part-1					
	V23CSES02	User Interface Design using Flutter					
		SWAYAM Plus - Android Application Development (with Flutter)					
10	V23CSP01	Evaluation of Community Service Internship	CSI	-	-	-	2
11	V23ENT04	English for Employability(EE)		-	-	-	-
			Total	**15**	**01**	**10**	**23**
		Minor Course (Student may select from the same specialized minors pool)	MC	3	0	0	3
		Minor Course (Student may select from the same specialized minors pool)	MC	3	0	0	3
		Honors Course (Student may select from the same honors pool)	HC	3	0	0	3
		Honors Course (Student may select from the same honors pool)	HC	3	0	0	3

SEMESTER-VI

S.No.	Course Code	Name of the Course		L	T	P	C
1	V23CST12	Machine Learning	PCC	3	0	0	3
2	V23CST13	Cloud Computing	PCC	3	0	0	3
3	V23CST14	Cryptography & Network Security	PCC	3	0	0	3
4	**Professional Elective-II**		PEC	3	0	0	3
	V23CSTPE05	1. Software Testing Methodologies					
	V23CSTPE06	2. Cyber Security					
	V23CSTPE07	3. DevOps					
	V23CSTPE08	4. Mobile Adhoc Networks					
		5. 12 week MOOC Swayam/NPTEL course recommended by the BoS					
5	**Professional Elective-III**		PEC	3	0	0	3
	V23CSTPE09	1. Software Project Management					
	V23CSTPE10	2. Natural Language Processing					
	V23CSTPE11	3. Big Data Analytics					
	V23CSTPE12	4. Distributed Operating System					
		5. 12 week MOOC Swayam/NPTEL course recommended by the BoS					
6	**Open Elective -II / Job Oriented Elective -II**		OEC / JOE	3	0	0	3
		Offline / 12 week MOOC Swayam/NPTEL course recommended by the BoS					
7	V23CSL10	Cloud Computing Lab	PCC	0	0	3	1.5
8	V23CSL11	Machine Learning Lab	PCC	0	0	3	1.5
9	V23CSSE04	Master Coding and Competitive Programming - Part-2	SEC	0	1	2	2
		SWAYAM Plus - 21st Century Soft Skills					
10	V23ENT05	Technical Paper Writing &IPR	Audit Course	2	0	0	-
			Total	**20**	**01**	**08**	**23**
		Minor Course (Student may select from the same specialized minors pool)	MC	3	0	0	3
		Minor Course (Student may select from the same specialized minors pool)	MC	3	0	0	3
		Honors Course (Student may select from the honors pool)	HC	3	0	0	3
		Honors Course (Student may select from the honors pool)	HC	3	0	0	3

* Under Industry Internship interested students can pursue SWAYAM Plus courses viz., Hands-on Masterclass on Data Analytics OR Artificial Intelligence for Real-World Application

Annexure-II

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CST09
Name of the Course	**Data Warehousing & Data Mining**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Discuss the basic concepts and techniques of data warehousing and data mining. **(K2)**

CO2: Demonstrate the types of the data to be mined and apply pre-processing methods on raw data. **(K3)**

CO3: Illustrate various Classification Techniques. **(K3)**

CO4: Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms. **(K3)**

CO5: Use different clustering techniques to cluster data. **(K3)**

UNIT I: Data Warehousing and Online Analytical Processing: Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse, Data Mining and Patten Mining, Technologies, Applications, Major issues, Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

UNIT II: Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

UNIT III: Classification: Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection.

UNIT IV: Association Analysis: Problem Definition, Frequent Item set Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, **Improving the efficiency of Apriori**, Compact Representation of frequent item sets, FP-Growth Algorithm.

UNIT V: Cluster Analysis: Overview, Basics and Importance of Cluster Analysis, Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bi-secting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses.

Text Books:

1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:

1. Data Mining: VikramPudi and P. Radha Krishna, Oxford Publisher.
2. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press,2013.
3. (NPTEL course by Prof.PabitraMitra) http://onlinecourses.nptel.ac.in/noc17_mg24/preview
4. http://www.saedsayad.com/data_mining_map.htm

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CST10
Name of the Course	**Computer Networks**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Illustrate fundamentals of networks, Reference Models and Transmission Media. **(K2)**
CO2: Demonstrate Error Detection & Correction and Data Link Layer protocols. **(K3)**
CO3: Illustrate the protocols of MAC sub layer and Channelization. **(K2)**
CO4: Utilizeconnection and connectionless services and Apply Routing and Congestion Control Algorithms. **(K3)**
CO5: Explain Transport Layer and Application Layer protocols. **(K2)**

UNIT I: Introduction: Network Types, LAN, MAN, WAN, Network Topologies Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models, OSI Vs TCP/IP.
Physical Layer –Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and introduction about unguided media.

UNIT II: Data link layer: Design issues, **Framing**: fixed size framing, variable size framing, flow control, error control, error detection and correction codes, CRC, Checksum: idea, one's complement internet checksum, services provided to Network Layer, **Elementary Data Link Layer protocols**: Simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel. **Sliding window protocol:** One bit, Go back N, Selective Repeat-Stop and wait protocol, Data link layer in HDLC, point to point protocol (PPP)

UNIT III: Media Access Control: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, **Controlled Access:** Reservation, Polling, Token Passing, **Channelization:** frequency division multiple Access (FDMA), time division multiple access (TDMA), code division multiple access (CDMA).
Wired LANs: Ethernet, Ethernet Protocol, Standard Ethernet, Fast Ethernet(100 Mbps), Gigabit Ethernet, 10 Gigabit Ethernet.

UNIT IV: The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks,
Routing Algorithms-The Optimality principle-Shortest path, Flooding, Distance vector, Link state, Hierarchical, Congestion Control algorithms-General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing-

Admission Control-Traffic Throttling-Load Shedding. Traffic Control Algorithm-Leaky bucket & Token bucket.

Internet Working: How networks differ- How networks can be connected- Tunnelling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version 4 protocol-IPV4 Header Format, IP addresses, Class full Addressing, CIDR, Subnets-IP Version 6-The main IPV6 header, Transition from IPV4 to IPV6, Comparison of IPV4 & IPV6.

UNIT V:The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications- Transmission control protocol: TCP services- TCP features- Segment- A TCP connection- windows in TCP- flow control-Error control, Congestion control in TCP.

Application Layer –- World Wide Web: HTTP, Electronic Mail-Architecture- web based mail-email security- TELENET-local versus remote Logging-Domain Name System.

Text Books:

1. Computer Networks, Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
2. Data Communications and Networks, Behrouz A. Forouzan, Fifth Edition TMH.

Reference Books:

1. Data Communications and Networks- Achut S Godbole, AtulKahate, 2nd Edition, McGrawHill
2. Computer Networks, Mayank Dave, 1st Edition, CENGAGE.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CST11
Name of the Course	**Automata and Compiler Design**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Construct Finite Automata and Regular Expressions. **(K3)**
CO2: Describe the Compilation Process and Lexical Analysis. **(K2)**
CO3: Construct Top down and Bottom up Parsing Techniques. **(K3)**
CO4: Produce Intermediate Code Generation and Runtime Environments**.** **(K3)**
CO5: Explain Code Optimization and Code Generation. **(K2)**

UNIT I: Formal Language and Regular Expressions: Alphabet, Strings, Language, Finite Automaton- Design of DFA, Design of NFA, Equivalence between NFA and DFA, Finite Automata with ϵ-Transition, Equivalence between NFA and ϵ-NFA. **Regular Expression**: Regular expressions Equivalence between Regular Expressions and Finite Automata ,Chomsky Hierarchy.

UNIT II: Compiler: Definition, Structure of a compiler. **Lexical Analysis**: The Role of the Lexical Analyzer, Specification of Tokens, Recognition of Tokens and the Lexical-Analyzer Generator-Lex. **Context Free grammars**: Context free grammars, derivation, parse trees, Ambiguous Grammar, Writing a Grammar-Elimination of Left Recursion, Left Factoring.

UNIT III: Top Down Parsing: First and Follow, LL(1) Grammars, **Bottom-Up Parsing**: Bottom Up Parser Classification, Reductions, Handle Pruning, Shift-Reducing, Constructing SLR Parsing Tables, construction of CLR (1), LALR Parsing tables, Comparison of all Bottom Up approaches.

UNIT IV: Semantic Analysis: Syntax Directed Definitions, Evaluation Orders for SDD‘s

Intermediate Code Generation: Variants of Syntax Trees, Three-Address Code, Basic blocks and Flow graphs, Control Flow. **Run-Time Environments:** Storage Organization, Stack Allocation of Space, Heap Management

UNIT V: Code optimization: Machine Independent Optimization. The principle sources of Optimization, optimization of Basic blocks, peep hole Optimization, Introduction to Date flow Analysis.

Code generation: Issues in design of code generation, The target Language, Address in the target code, A Simple Code generation.

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwaniand J.D.Ullman, 3rdEdition, Pearson, 2008.
2. Compilers Principles, Techniques and Tools Aho, Ullman, Ravisethi, Pearson Education.

Reference Books:

1. Louden: “Compiler Construction, Principles & Practice”, 1st Edition, Thomson Press, 2006.
2. Tremblay J P, Sorenson G P: “The Theory & Practice of Compiler writing”, 1st Edition, BSP Publication, 2010.
3. Theory of Computation, V. Kulkarni, Oxford University Press, 2013.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE01
Name of the Course	**Object Oriented Analysis and Design (Professional Elective-I)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Illustrate the inherent complexity and attributes of software systems and Design structured solutions for complex problems by applying principles of system architecture. **(K3)**

CO2: Use the significance and principles of object-oriented modeling and apply UML concepts to represent structural aspects of software systems using appropriate diagrams. **(K3)**

CO3: Develop detailed class and object models and differentiate between basic and advanced structural constructs such as interfaces, packages, and roles for effective software design. **(K3)**

CO4: Construct behavioral models such as use case, interaction, and activity diagrams to illustrate user interactions and system functionalities in real-world applications. **(K3)**

CO5: Design comprehensive behavioral and architectural models using state charts, component, and deployment diagrams to demonstrate system dynamics and implementation strategies. **(K3)**

UNIT I: Introduction: The Structure of Complex systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems. **Case Study:** System Architecture: Satellite-Based Navigation

UNIT II: Introduction to UML: Importance of modeling, principles of modeling, object-oriented modeling, conceptual model of the UML, Architecture, and Software Development Life Cycle. **Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams. **Case Study:** Control System: Traffic Management.

UNIT III: Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams. **Advanced Structural Modeling:** Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. **Case Study:** AI: Cryptanalysis.

UNIT IV: Basic Behavioral Modeling-I: Interactions, Use cases, Use case Diagrams, Interaction diagrams, Activity Diagrams. **Case Study:** Web Application: Vacation Tracking System

UNIT V:Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. **Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams. **Case Study:** Weather Forecasting

Text Books:

1. Grady BOOCH, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston, "Object- Oriented Analysis and Design with Applications",3rd Edition, 2013, PEARSON.
2. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, 1st Edition, Pearson Education.

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, 1st Edition, PearsonEducation.
2. Pascal Roques: Modeling Software Systems Using UML2, 2004, WILEY- Dreamtech IndiaPvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, 2004, The McGraw-Hill Companies.
4. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, 2nd Edition, Craig Larman, Pearson Education.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE02
Name of the Course	**Artificial Intelligence (Professional Elective-I)**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Understand the fundamental concepts of Artificial Intelligence. **(K2)**
CO2: Employ **v**arious search strategies and game-playing algorithms to solve AI problems. **(K3)**
CO3: Applyknowledge representation techniques and probabilistic reasoning methods to model AI systems. **(K3)**
CO4: Use logic based inference techniques and learning methods for intelligent decision- making. **(K3)**
CO5: Demonstrate the principles of expert system architecture and knowledge acquisition in AI applications. **(K3)**

UNIT I: Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT II : Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT III: Representation of Knowledge: Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes' probabilistic interferences and dempstershafer theory.

UNIT IV: Logic concepts: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT V: Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells. **AI Agent Communication Protocols:** Model Context Protocol (MCP), Agent-to-Agent Protocol (A2A) / Inter-agent Protocol and Agent Communication Protocol (ACP).

Textbooks:

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach",SecondEdition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", McGraw Hill

Reference Books:

1. David Poole, Alan Mackworth, Randy Goebel,"Computational Intelligence: a logical approach", Oxford University Press.
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problemsolving", Fourth Edition, Pearson Education.
3. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

Online Learning Resources:

1. https://ai.google/

2. https://swayam.gov.in/nd1_noc19_me71/preview

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE03
Name of the Course	**Microprocessors & Microcontrollers(Professional Elective-I)**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Understand the fundamental concepts of Artificial Intelligence. **(K2)**
CO2: Employ various search strategies and game-playing algorithms to solve AI problems. **(K3)**
CO3: Applyknowledge representation techniques and probabilistic reasoning methods to model AI systems. **(K3)**
CO4: Use logic based inference techniques and learning methods for intelligent decision- making. **(K3)**
CO5: Demonstrate the principles of expert system architecture and knowledge acquisition in AI applications. **(K3)**

UNIT I: 8086 Architecture: Main features, pin diagram/description, 8086 microprocessor family, internal architecture, bus interfacing unit, execution unit, interrupts and interrupt response, 8086 system timing, minimum mode and maximum mode configuration.

UNIT II: 8086 Programming: Program development steps, instructions, addressing modes, assembler directives, writing simple programs with an assembler, assembly language program development tools.

UNIT III: 8086 Interfacing: Semiconductor memories interfacing (RAM, ROM), Intel 8255 programmable peripheral interface, Interfacing switches and LEDS, Interfacing seven segment displays, software and hardware interrupt applications, Intel 8251 USART architecture and interfacing, Intel 8237a DMA controller, stepper motor, A/D and D/A converters, Need for 8259 programmable interrupt controllers.

UNIT IV: Microcontroller, Architecture of 8051, Special Function Registers(SFRs), I/O Pins Ports and Circuits, Instruction set, Addressing modes, Assembly language programming.

UNIT V: Interfacing Microcontroller, Programming 8051 Timers, Serial Port Programming, Interrupts Programming, LCD & Keyboard Interfacing, ADC, DAC & Sensor Interfacing, External Memory Interface, Stepper Motor and Waveform generation, Comparison of Microprocessor, Microcontroller, PIC and ARM processors

Textbooks:

1. Microprocessors and Interfacing – Programming and Hardware by Douglas V Hall, SSSP Rao, Tata McGraw Hill Education Private Limited, 3rd Edition,1994.
2. K M Bhurchandi, A K Ray, Advanced Microprocessors and Peripherals, 3rd edition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2nd edition, Pearson, 2012.

Reference Books:

1. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 6th edition, Penram International Publishing, 2013.
2. Kenneth J. Ayala, The 8051 Microcontroller, 3rd edition, Cengage Learning, 2004.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE04
Name of the Course	**Quantum Computing(Professional Elective-I)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Understand the foundational principles of quantum mechanics relevant to computing**.(K2)**
CO2: Explain quantum computing or related interdisciplinary fields(mathematics, physics) . **(K2)**
CO3: Construct various physical quantum units of information. **(K3)**
CO4: Demonstrate the various quantum Algorithms. **(K3)**
CO5: Illustrate the role of noise in quantum systems and the necessity for error correction.**(K3)**

UNIT I: History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT II: Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements. Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (Central Dogma)

UNIT III: Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT IV: Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm.

UNIT V: Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation

Textbooks:

1. Quantum Computation and Quantum Information, Nielsen M. A.,Cambridge
2. Programming Quantum Computers, Essential Algorithms and Code Samples, Eric R Johnson, Nic Harrigan, Mercedes Ginemo, Segovia, Oreilly

Reference Books:

1. Quantum Computing for Computer Scientists, Noson S. Yanofsk, Mirco A. Mannucci
2. Principles of Quantum Computation and Information,Benenti G., Casati G. and Strini G., Vol.I: Basic Concepts, Vol II
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	3	1.5	V23CSL08
Name of the Course	**Data Mining Lab**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment.(**K2)**

CO2: Develop a data warehouse or data mart to present information needed by management in a form that is usable. **(K3)**

CO3: Practice hands-on experience working with all real data sets. **(K3)**

CO4: Compute real data sets using popular data mining tools such as WEKA, Python Libraries. **(K3)**

CO5: Develop ability to design various algorithms based on data mining tools. **(K3)**

List of Experiments

(**Software Requirements:** The tasks should be implementing by using WEKA Tool/Python/R-Tool/Rapid Tool/Oracle Data mining)

1**. Illustrate the Creation of a Data Warehouse.**

- Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects,etc.,)
- Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc).
- Develop ETL scripts and implement using data warehouse tools.
- Perform Various OLAP operations such slice, dice, roll up, drill up and pivot

2. **Describe the machine learning tool "WEKA"**

- Explore WEKA Data Mining/Machine Learning Toolkit.
    - Downloading and/or installation of WEKA data mining toolkit.
    - Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
    - Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
- Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weather dataset, Iris dataset, etc.)
- Load each dataset and observe the following:
  1. List the attribute names and they types
  2. Number of records in each dataset
  3. Identify the class attribute (if any)
  4. Plot Histogram
  5. Determine the number of records for each class.
  6. Visualize the data in various dimensions.

3. **Demonstrate data preprocessing tasks and also performing association rule mining on data sets using WEKA/Python.**

- Explore various options available in Weka for preprocessing data and apply Unsupervised filters like Discretization, Resample filter, etc. on each dataset
- Load weather. nominal, Iris, Glass datasets into Weka and run Apriori Algorithm with different support and confidence values.
- Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated.
- Derive interesting insights and observe the effect of discretization in the rule generation process.

4. **Demonstrate performing classification on data sets**

- Load each dataset and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
- Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.
- Load each dataset into Weka/Python and perform k-Nearest Neighbour classification. Interpret the results obtained.
- Plot RoC Curves
- Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

5. Demonstrate performing clustering of data sets

- Load each dataset into Weka/R and run simple k-means clustering algorithm with different values of k (number of desired clusters).
- Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
- Explore other clustering techniques available in Weka/R.
- Explore visualization features of Weka/R to visualize the clusters. Derive interesting insights and explain.

6. **Demonstrate knowledge flow application on data sets into Weka/R**

- Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms
- Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm
- Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree.

7. Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations
8. Develop a java program to prepare a simulated data set with unique instances.
9. Develop a Python program to generate frequent item sets / association rules using Apriori algorithm
10. Develop a program to calculate chi-square value using Python/R. Report your observation.
11. Develop a program of Naive Bayesian classification using Python/R programming language.

12. Implement a Java/R program to perform Apriori algorithm
13. Develop a R program to cluster your choice of data using simple k-means algorithm using JDK
14. Develop a program of cluster analysis using simple k-means algorithm Python/R programming language.
15. Develop a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python
16. Visualize the datasets using matplotlib in python/R.(Histogram, Box plot, Bar chart, Pie chart etc.,)

Textbooks:

1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:

1. Data Mining: Practical Machine Learning Tools and Techniques, IanH.Witten, Eibe Frank, Mark A.Hall,3rdEdition,MorganKaufmannPublishers.
2. Data Mining Concepts and Techniques,JiaweiHan,MichelineKamber,JianPei,3rd Edition, Morgan Kaufmann Publishers.
3. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach,VipinKumar,1stEdition,Pearson Education Inc

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	3	1.5	V23CSL09
Name of the Course	**Computer Networks Lab**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Objectives: After Successful completion of the Course, the student will be able to:

CO 1: Develop programs on data link layer framing methods and checksum. **(K3)**

CO 2: Develop programs to implement Error detection technique and Sliding window protocol. **(K3)**

CO 3: Develop programs to implement Routing and Congestion Control Algorithms. **(K3)**

CO 4: Demonstrate various networking tools like Wireshark, NS2 and Nmap. **(K3)**

List of Experiments

1. Study of Network devices in detail and connect the computers in Local Area Network.
2. Develop a Program to implement the data link layer framing methods such as
   i) Character stuffing ii) bit stuffing.
3. Develop a Program to implement data link layer farming method checksum.
4. Develop a program for Hamming Code generation for error detection and correction.
5. Develop a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
6. Develop a Program to implement Sliding window protocol for Goback N.
7. Develop a Program to implement Sliding window protocol for Selective repeat.
8. Develop a Program to implement Stop and Wait Protocol.
9. Develop a program for congestion control using leaky bucket algorithm
10. Develop a Program to implement Dijkstra‘s algorithm to compute the Shortest path through a graph.
11. Develop a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).
12. Develop a Program to implement Broadcast tree by taking subnet of hosts.
13. Wireshark
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
14. Discover active hosts, open ports and running services in a network using Nmap
15. Find Operating System in a host using Nmap
16. Do the following using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
    v. Simulate to Compare Data Rate& Throughput.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	1	2	2	V23CSSE03
Name of the Course	**Full Stack Development-2 (Skill Enhancement Course)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the key concepts of ExpressJS such as routing, HTTP methods, middleware, templating, and handling form data. **(K2)**

CO2: Apply ExpressJS to implement cookies, sessions, authentication, database operations, and RESTful APIs in server-side applications. **(K3)**

CO3: Develop ReactJS components using JSX, props, state, and styling to handle events and user interactions**.(K3)**

CO4: Analyse ReactJS techniques such as conditional rendering, forms, routing, hooks, and screen updates to manage dynamic content. **(K4)**

CO5: Apply MongoDB CRUD operations to create and manage databases, collections, and documents for full-stack applications. **(K3)**

Experiments covering the Topics:

- ExpressJS – Routing, HTTP Methods, Middleware, Templating, Form Data
- ExpressJS – Cookies, Sessions, Authentication, Database, RESTful APIs
- ReactJS – Render HTML, JSX, Components – function & Class, Props and States, Styles, Respond to Events
- ReactJS – Conditional Rendering, Rendering Lists, React Forms, React Router, Updating the Screen
- ReactJS – Hooks, Sharing data between Components, Applications – To-do list and Quiz
- MongoDB – Installation, Configuration, CRUD operations, Databases, Collections and Records

Sample Experiments:

1. ExpressJS – Routing, HTTP Methods, Middleware.

a. Develop a program to define a route, Handling Routes, Route Parameters, Query Parameters and URL building.
b. Develop a program to accept data, retrieve data and delete a specified resource using http methods.
c. Develop a program to show the working of middleware.

2. ExpressJS – Templating, Form Data

a. Develop a program using templating engine.
b. Develop a program to work with form data.

3. ExpressJS – Cookies, Sessions, Authentication

a. Develop a program for session management using cookies and sessions.
b. Develop a program for user authentication.

4. ExpressJS – Database, RESTful APIs

a. Develop a program to connect MongoDB database using Mangoose and perform CRUD operations.
b. Develop a program for single page application using RESTful APIs.

5. ReactJS – Render HTML, JSX, Components – function & Class

a. Develop a program to render HTML to a web page.
b. Develop a program for writing markup with JSX.
c. Develop a program for creating and nesting components (function and class).

6. ReactJS – Props and States, Styles, Respond to Events

a. Develop a program to work with props and states.
b. Develop a program to add styles (CSS & Sass Styling) and display data.
c. Develop a program for responding to events.

7. ReactJS – Conditional Rendering, Rendering Lists, React Forms

a. Develop a program for conditional rendering.
b. Develop a program for rendering lists.
c. Develop a program for working with different form fields using react forms.

8. ReactJS – React Router, Updating the Screen

a. Develop a program for routing to different pages using react router.
b. Develop a program for updating the screen.

9. ReactJS – Hooks, Sharing data between Components

a. Develop a program to understand the importance of using hooks.
b. Develop a program for sharing data between components.

10. MongoDB – Installation, Configuration, CRUD operations

a. Install MongoDB and configure ATLAS
b. Develop MongoDB queries to perform CRUD operations on document using insert(), find(), update(), remove()

11. MongoDB – Databases, Collections and Records

a. Develop MongoDB queries to Create and drop databases and collections.
b. Develop MongoDB queries to work with records using find(), limit(), sort(), createIndex(), aggregate().

12. Augmented Programs: (Any 2 must be completed)

a. Design a to-do list application using NodeJS and ExpressJS.
b. Design a Quiz app using ReactJS.
c. Complete the MongoDB certification from MongoDB University website.

Text Books:

1. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.
2. Node.Js in Action, Mike Cantelon, Mark Harter, T.J. Holowaychuk, Nathan Rajlich,Manning Publications. (Chapters 1-11)
3. React Quickly, AzatMardan,Manning Publications(Chapters 1-8,12-14)

Web Links:

1. ExpressJS - https://www.tutorialspoint.com/expressjs
2. ReactJS - https://www.w3schools.com/REACT (and) https://react.dev/learn#
3. MongoDB - https://learn.mongodb.com/learning-paths/introduction-to-mongodb

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	2	1	V23CSES01
Name of the Course	**Master Coding and Competitive Programming - Part-1**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Apply Mathematical reasoning and number theory to solve real world problems in linear time. **(K3)**
CO2: Use of modular arithmetic, to solve complex problems in linear time , logarithmic. **(K3)**
CO3: Use of Prime Factorization and complex solve problems. **(K3)**
CO4: Analyse different techniques including sieve to find prime numbers and evaluate efficiency of these methods. **(K4)**
CO5: Experiment with Hashing and searching techniques to solve problems on Arrays in Linear time. **(K3)**

List of Experiments

1. Develop Programs to solve problems based on Mathematical logic, Reasoning and number theory
2. Develop programs using different techniques to find prime number
3. Develop programs using Sieve method and optimize Complexity of finding prime number
4. Develop Programs based on series, patterns
5. Develop programs on concept of Fibonacci series
6. Develop programs on strings including palindrome and anagram concepts
7. Develop programs to search pattern in a string
8. Develop programs for String Processing.

Text Books:

1. Java The Complete Reference - Eleventh Edition, Herbert Schildt, Oracle
2. Guide to Competitive Programming by Antti Laaksonen
3. Programming challenges by Steven S Skiena

Tools:

1. practice.geeksfoegeeks.com
2. leetcode.com
3. codingninjas.com
4. Hackerrank.com
5. Interviewbit.com

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	2	1	V23CSES02
Name of the Course	**User Interface Design Using Flutter**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO 1: Apply Dart and Flutter setup procedures to build basic mobile application components. **(K3)**
CO 2: Develop responsive user interfaces using layout widgets and media queries. **(K3)**
CO 3: Apply navigation techniques and state management strategies to enhance functionality in Flutter applications. **(K3)**
CO 4: Construct functional mobile app components using API integration, animations, custom widgets, and debugging tools. **(K3)**

List of Experiments

Students need to implement the following experiments

1. a) Show the Installation process of Flutter and Dart SDK.
   b) Develop a simple Dart program to understand the language basics.
2. a) Practice various Flutter widgets (Text, Image, Container, etc.).
   b) Develop different layout structures using Row, Column, and Stack widgets.
3. a) Develop a responsive UI that adapts to different screen sizes.
   b) Develop media queries and breakpoints for responsiveness.
4. a) Illustrate the navigation Set up process between different screens using Navigator.
   b) Develop navigation with named routes.
5. a) Expalin about stateful and stateless widgets.
   b) Develop state management using set State and Provider.
6. a) Develop custom widgets for specific UI elements.
   b) Apply styling using themes and custom styles.
7. a) Develop a form with various input fields.
   b) Develop form validation and error handling.
8. a) Apply animations to UI elements using Flutter's animation framework.
   b) Experiment with different types of animations (fade, slide, etc.).
9. a) Demonstrate data Fetching from a REST API.
   b) Show the fetched data in a meaningful way in the UI.
10. a) Develop unit tests for UI components.
    b) Use Flutter's debugging tools to identify and fix issues.

Text Books:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.
2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st Edition, Apres
3. Richard Rose, Flutter & Dart Cookbook, Developing Full stack Applications for the Cloud, Oreilly.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	2	0	0	**V23ENT04**
Name of the Course	**English for Employability**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Demonstrate proficiency in verbal ability, including word meanings, synonyms, antonyms, and contextual usage, to enhance their communication skills and excel in competitive exams. **(K3)**
CO2: Effectively apply various reading techniques and comprehension skills, including literal, inferential, critical, evaluative, and appreciative reading, to accurately understand and interpret complex texts. **(K3)**
CO3: Apply knowledge and skills to participate effectively in group discussions on various topics, showcasing critical thinking, communication, and teamwork abilities. **(K3)**
CO4: Construct a well-organized and compelling resume tailored to specific job descriptions and industries. **(K3)**
CO5: Analyse the skills required to excel in an interview and identify areas for improvement. **(K1)**

UNIT I: VERBAL ABILITY-Words; Meanings; Synonyms; Antonyms; Contextual Usage; Various MNC mock test papers practice

UNIT II: READING COMPREHENSION-Reading Techniques; Types of RC; Literal; Inferential; Critical; Evaluative; Appreciative

UNIT III: GROUP DISCUSSION-Dos & Don'ts; Types of GD; Technical Topics; Case Studies; Abstract Topics; Current Affairs

UNIT IV:RESUME BUILDING-Tailoring resume as per job description; Making Video Resume (Visume)

UNIT V:MOCK INTERVIEWS-Tailored to Specific job roles; Dos & Don'ts; Types of Mock Interviews; Traditional-Behavioural-Technical

Reference Books:

1. Lewis, N. (1978). Word Power Made Easy. Pocket Books.
2. Wren, P. C., & Martin, H. (1990). High School English Grammar & Composition. S. Chand.
3. Krannich, C. R. (2004). 101 Dynamite Answers to Interview Questions. Impact Publications.
4. Jackson, T. (2009). Resume Magic: Trade Secrets of a Professional Resume Writer. JIST Publishing.
5. Lisa McGrimmon, The Resume Writing Guide: A Step-by-Step Workbook for Creating a Winning Resume (2013). CareerChoiceGuide; 2nd edition.
6. Puri, G. (2018). Verbal Ability and Reading Comprehension for CAT and Other MBA Entrance Examinations. Upkar Prakashan.

Semester	**V**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	2	0	0	
Name of the Course	**Aptitude**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Analyze appropriate methods of logical thinking on Ratio and Proportion, Partnership, LCM and HCF, Number System, Areas & Volumes. (K2)

CO2: Demonstrate problem solving skills through the concepts of Percentages, Profit and loss, Simple Interest & Compound Interest and Allegation. (K3)

CO3: Calculate the end results of Cubes, Dice and Data Analysis, Time & Work, Time & Distance, Race & Games. (K4)

UNIT I: Ratio and Proportion, Partnership, LCM and HCF, Number System, Areas & Volumes.

UNIT II: Percentages, Profit and loss, Simple Interest & Compound Interest and Allegation.

UNIT III: Cubes, Dice and Data Analysis, Time & Work, Time & Distance, Race & Games.

Text Book: WORK BOOK ON APTITUDE

Reference Books:

- Training & Placement cell, 2025, Workbook -2 on Aptitude, Sri Vasavi Engineering College.
- M Tyra, 2013, Magical Book on Quicker maths, BSC Publications.

Hyperlinks

1. https://www.indiabix.com/
2. https://www.campusgate.co.in/
3. https://www.instacks.com/

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CST12
Name of the Course	**Machine Learning**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain machine learning paradigms, data preparation, and model evaluation**. (K2)**

CO2: Apply distance-based techniques and algorithms to classify and regress data using proximity measures. **(K3)**

CO3: Construct decision tree and Bayesian models to perform classification and regression. **(K3)**

CO4: Demonstrate linear discriminants, SVMs, and neural networks to solve classification and regression problems in machine learning. **.(K3**)

CO5: Apply ensemble techniques and graphical model-based approaches to develop predictive and sequential models for real-world problems. **(K3)**

UNIT I: Introduction to Machine Learning: Evolution of Machine Learning, Paradigms for ML, Learning by Rate, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Prediction, Search and Learning, Data Sets.

UNIT II: Nearest Neighbour-Based Models: Introduction to Proximity Measures, Distance Measures, Non-Metric Similarity Functions, Proximity Between Binary Patterns, Different Classification Algorithms Based on the Distance Measures ,K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression, Performance of Classifiers, Performance of Regression Algorithms.

UNIT III: Models Based on Decision Trees: Decision Trees for Classification, Impurity Measures, Properties, Regression Based on Decision Trees, Bias–Variance Trade-off, Random Forests for Classification and Regression. The Bayes Classifier: Introduction to the Bayes Classifier, Bayes' Rule and Inference, The Bayes Classifier and its Optimality, Multi-Class Classification, Class Conditional Independence and Naive Bayes Classifier (NBC)

UNIT IV: Linear Discriminants for Machine Learning: Introduction to Linear Discriminants, Linear Discriminants for Classification, Perceptron Classifier, Perceptron Learning Algorithm, Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick, Logistic Regression, Linear Regression, Multi-Layer Perceptrons (MLPs), Backpropagation for Training an MLP.

UNIT V: Ensemble Learning: Boosting, Bagging, Random Forest, Different Ways to Combine Classifiers. **Graphical Models:**Bayesian Networks, Markov Random Fields, Hidden Markov Models, Tracking Methods.

Textbooks:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024
2. "Machine Learning: An Algorithmic Perspective", Stephen Marsland, 2nd Ed., CRC Press.

Reference Books:

1. "Machine Learning", Tom M. Mitchell, McGraw-Hill Publication, 2017
2. "Machine Learning in Action",Peter Harrington, DreamTech
3. "Introduction to Data Mining", Pang-Ning Tan, Michel Stenbach, Vipin Kumar, 7th Edition, 2019.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CST13
Name of the Course	**Cloud Computing**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the evolving utility computing model called cloud computing. **(K2)**
CO2: Explain the different service models of cloud computing, such as IaaS, PaaS, and SaaS. **(K2)**
CO3: Explain the core concepts of cloud technologies like distributed computing, service-oriented architecture, and virtualization. **(K2)**
CO4: Analyze the security and other challenges in cloud computing. **(K4)**
CO5: Apply advanced concepts such as containers, serverless computing, and cloud-centric Internet of Things. **(K3)**

UNIT I: Introduction to Cloud Computing Fundamentals: Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services (IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google AppEngine).

UNIT II: Cloud Enabling Technologies: Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures for parallel computing (SISD, SIMD, MISD, MIMD), elements of distributed computing, Inter-process communication, technologies for distributed computing, remote procedure calls (RPC), service-oriented architecture (SOA), Web services, virtualization.

UNIT III: Virtualization and Containers: Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud Computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM (e.g. Amazon EC2) and container (e.g. Amazon Elastic Container Service) offerings.

UNIT IV: Cloud computing challenges : Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.

UNIT V: Advanced concepts in Cloud Computing: Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g. AWS Lambda) and open-source (e.g. OpenFaaS) serverless platforms, Internet of Things (IoT), applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.

Textbooks

1. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, ShivanandaPoojara, Satish N. Srirama, 2nd Edition, Mc Graw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra,Elsevier, 2012.

Reference Books:

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
2. Essentials of cloud Computing, K. Chandrasekhran, CRC press, 2014.
3. Online documentation and tutorials from cloud service providers (e.g., AWS, Azure, GCP).

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CST14
Name of the Course	**Cryptography & Network Security**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain fundamentals of security concepts and mathematical support of Cryptography. **(K2)**
CO2: Discuss Symmetric cryptosystems. **(K2)**
CO3: Discuss Asymmetric cryptosystems. **(K2)**
CO4: Apply Hash functions & Digital Signatures to provide authentication and integrity. **(K3)**
CO5: Demonstrate various methods of Mutual trust and mail security Network Security Scenarios. **(K3)**

UNIT I: Basic Principles: Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography- integer arithmetic, modular arithmetic, matrices, linear conguence.

UNIT II: Symmetric Encryption: Mathematics of Symmetric Key Cryptography-algebraic structures, GF(2^n) Fields, Introduction to Modern Symmetric Key Ciphers-modern block ciphers, modern stream ciphers, Data Encryption Standard- DES structure, DES analysis, Security of DES, Multiple DES, Advanced Encryption Standard-transformations, key expansions, AES ciphers, Analysis of AES.

UNIT III: Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography-primes, primality testing, factorization, CRT, Asymmetric Key Cryptography- RSA crypto system Rabin cryptosystem, Elgamal Crypto system, ECC

UNIT IV: Data Integrity, Digital Signature Schemes & Key Management: Message Integrity and Message Authentication-message integrity, Random Oracle model, Message authentication, Cryptographic Hash Functions-whirlpool, SHA-512, Digital Signature- process, services, attacks, schemes, applications, Key Management-symmetric key distribution, Kerberos.

UNIT V: Network Security-I: Security at application layer: PGP and S/MIME, Security at the Transport Layer: SSL and TLS, **Network Security-II:** Security at the Network Layer: IPSec-two modes, two security protocols, security association, IKE, ISAKMP, System Security-users, trust, trusted systems, buffer overflow, malicious software, worms, viruses, IDS, Firewalls.

Textbooks:

1. Cryptography and Network Security, 3rd Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015
2. Cryptography and Network Security,4th Edition, William Stallings, (6e) Pearson,2006
3. Everyday Cryptography, 1st Edition, Keith M.Martin, Oxford,2016

Reference Books:

1. Network Security and Cryptography, Bernard Meneges 1st Edition, Cengage Learning,2018

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE05
Name of the Course	**Software Testing Methodologies(Professional Elective-II)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe Path Testing using flowgraphs. **(K2)**
CO2: Demonstrate Transaction Flow, Dataflow and Domain Testing Techniques. **(K3)**
CO3: Apply Path Expressions and Logic-Based Testing. **(K3)**
CO4: Demonstrate State Transition Testing Using State Graphs. **(K3)**
CO5: Implement Graph Matrices and Testing Tools for Test Case Generation, Automation. **(K3)**

UNIT I: Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing. **(Textbook-1)**

UNIT II: Transaction Flow Testing: transaction flows, transaction flow testing techniques.
Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.
Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability. **(Textbook-1)**

UNIT III: Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications. **(Textbook-1)**

UNIT IV: State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips. **(Textbook-1)**

UNIT V: Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon). **(Textbook-2)**

Textbooks:
1. Software Testing techniques - BarisBeizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

Reference Books:
1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE06
Name of the Course	**Cyber Security (Professional Elective-II)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the fundamental concepts of cybercrime, its origins, types, and the challenges posed by modern devices and technologies. **(K2)**
CO2: Demonstrate various tools and techniques used in committing cybercrimes, including phishing, malware, spoofing, session hijacking, and social engineering. **(K2)**
CO3: Demonstrate knowledge of cybercrime investigation processes including digital evidence collection, eDiscovery, email/IP tracking, and the use of encryption/decryption tools. **(K2)**
CO4: Apply computer forensic techniques and use current tools to investigate and analyze digital evidence on different platforms such as Windows, Linux, and mobile devices. **(K3)**
CO5: Demonstrate the legal framework related to cybercrime, with a focus on the Indian IT Act, its challenges, amendments, and the global legal scenario. **(K2)**

UNIT I: Introduction to Cybercrime: Introduction, Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Cybercriminals, Classifications of Cybercrime, Cyberstalking, Cybercafe and Cybercrimes, Botnets. Attack Vector, Proliferation of Mobile and Wireless Devices, Security Challenges Posed by Mobile Devices, Attacks on Mobile/Cell Phones, Network and Computer Attacks.

UNIT II: Tools and Methods: Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, Sniffers, Spoofing, Session Hijacking Buffer over flow, DoS and DDoS Attacks, Attacks on Wireless Networks, Identity Theft (ID Theft), Foot Printing and Social Engineering, Port Scanning, Enumeration.

UNIT III: Cyber Crime Investigation: Introduction, Investigation Tools, eDiscovery, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Hands on Case Studies. Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

UNIT IV: Computer Forensics and Investigations: Understanding Computer Forensics, Preparing for Computer Investigations. Current Computer Forensics Tools: Evaluating Computer Forensics Tools, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Graphics and Network Forensics, E-mail Investigations, Cell Phone and Mobile Device Forensics.

UNIT V: Cyber Crime Legal Perspectives: Introduction, Cybercrime and the Legal Landscape around the World, The Indian IT Act, Challenges to Indian Law and Cybercrime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Amendments to the Indian IT Act, Cybercrime and Punishment, Cyber Law, Technology and Students: Indian Scenario.

Textbooks

1. Sunit Belapure Nina Godbole "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", WILEY, 2011.
2. Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi, 2009.

Reference Books:

1. Michael T. Simpson, Kent Backman and James E. Corley, "Hands on Ethical Hacking and Network Defence", Cengage, 2019.
2. Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi, 2009.
3. Alfred Basta, Nadine Basta,Mary Brown and Ravinder Kumar "Cyber Security and Cyber Laws" , Cengage,2018.

E-Resources:

1. CERT-In Guidelines- http://www.cert-in.org.in/
2. https://www.coursera.org/learn/introduction-cybersecurity-cyber-attacks [Online Course]
3. https://computersecurity.stanford.edu/free-online-videos [Free Online Videos]
4. Nickolai Zeldovich. 6.858 Computer Systems Security. Fall 2014. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu License:Creative CommonsBY-NC-SA.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE07
Name of the Course	**DevOps(Professional Elective-II)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the architecture, principles, and lifecycle of DevOps and how it integrates with Agile practices and automation tools.. **(K2)**

CO2: Apply GIT commands and workflows to manage source code collaboratively and analyze code quality using SonarQube. **(K3)**

CO3: Develop and manage continuous integration pipelines using Jenkins to automate build and testing processes. **(K3)**

CO4: Implement containerized applications using Docker and integrate testing tools for continuous delivery. **(K3)**

CO5: Use Ansible and OpenShift to automate application deployment and manage container orchestration in CI/CD pipelines. **(K3)**

UNIT I: Introduction to DevOps: Introduction to SDLC, Agile Model. Introduction to Devops. DevOps Features, DevOps Architecture, DevOps Lifecycle, Understanding Workflow and principles, Introduction to DevOps tools, Build Automation, Delivery Automation, Understanding Code Quality, Automation of CI/ CD. Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples.

UNIT II: Source Code Management (GIT): The need for source code control, The history of source code management, Roles and code, source code management system and migrations. What is Version Control and GIT, GIT Installation, GIT features, GIT workflow, working with remote repository, GIT commands, GIT branching, GIT staging and collaboration. UNIT TESTING - CODE COVERAGE: Junit, nUnit & Code Coverage with Sonar Qube, SonarQube - Code Quality Analysis.

UNIT III :Build Automation - Continuous Integration (CI): Build Automation, What is CI Why Cl is Required, CI tools, Introduction to Jenkins (With Architecture), jenkins workflow, jenkins master slave architecture, Jenkins Pipelines, PIPELINE BASICS - Jenkins Master, Node, Agent, and Executor Freestyle Projects & Pipelines, Jenkins for Continuous Integration, Create and Manage Builds, User Management in Jenkins Schedule Builds, Launch Builds on Slave Nodes.

UNIT IV: Continuous Delivery (CD): Importance of Continuous Delivery, CONTINUOUS DEPLOYMENT CD Flow, Containerization with Docker: Introduction to Docker, Docker installation, Docker commands, Images & Containers, DockerFile, Running containers, Working with containers and publish to Docker Hub.
Testing Tools: Introduction to Selenium and its features, JavaScript testing.

UNIT V: Configuration Management - ANSIBLE: Introduction to Ansible, Ansible tasks, Roles, Jinja templating, Vaults, Deployments using Ansible.
CONTAINERIZATION USING KUBERNETES(OPENSHIFT): Introduction to Kubernetes Namespace & Resources, CI/CD - On OCP, BC, DC & ConfigMaps, Deploying Apps on Openshift Container Pods. Introduction to Puppet master and Chef.

Textbooks:

1. Joyner, Joseph., Devops for Beginners: Devops Software Development Method Guide for Software Developers and It Professionals, 1st Edition Mihails Konoplows, 2015.
2. Alisson Machado de Menezes., Hands-on DevOps with Linux,1st Edition, BPB Publications, India, 2021.

Reference Books:

1. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley; ISBN-10
2. Gene Kim Je Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona, Joakim Practical DevOps, 1st Edition, Packt Publishing, 2016.
4. Joakim Verona. Practical Devops, Ingram short title; 2nd edition (2018). ISBN10: 1788392574
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications. ISBN: 9788126579952.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE08
Name of the Course	**Mobile Adhoc Networks (Professional Elective-III)**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1 :Explain MANETs, their uses, challenges, and types of MAC protocols. **(K2)**
CO2 : Apply suitable routing and transport protocols in ad hoc wireless networks. **(K3)**
CO3 : Describe common security threats and basic protection methods in MANETs. **(K2)**
CO4 : Use proper methods to collect and send data in wireless sensor networks. **(K3)**
CO5 : Operate tools and software to build secure wireless sensor network systems. **(K3)**

UNIT I: Introduction to Ad Hoc Wireless Networks- Cellular and Ad Hoc Wireless Networks, Characteristics of MANETs, Applications of MANETs, Issues and Challenges of MANETs, Ad Hoc Wireless Internet, MAC protocols for Ad hoc Wireless Networks-Issues, Design Goals and Classifications of the MAC Protocols.

UNIT II: Routing Protocols for Ad Hoc Wireless Networks- Issues in Designing a Routing Protocol, Classifications of Routing Protocols, Topology-based versus Position-based Approaches, Issues and design goals of a Transport layer protocol, Classification of Transport layer solutions, TCP over Ad hoc Wireless Networks, Solutions for TCP over Ad Hoc Wireless Networks, Other Transport layer protocols.

UNIT III: Security protocols for Ad hoc Wireless Networks- Security in Ad hoc Wireless Networks, Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management, Secure Routing in Ad hoc Wireless Networks, Cooperation in MANETs, Intrusion Detection Systems.

UNIT IV: Basics of Wireless Sensors and Applications- The Mica Mote, Sensing and Communication Range, Design Issues, Energy Consumption, Clustering of Sensors, Applications, Data Retrieval in Sensor Networks-Classification of WSNs, MAC layer, Routing layer, Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

UNIT V: Security in WSNs- Security in WSNs, Key Management in WSNs, Secure Data Aggregation in WSNs, Sensor Network Hardware-Components of Sensor Mote, Sensor Network Operating Systems–TinyOS, LA-TinyOS, SOS, RETOS, Imperative Language-nesC, **Dataflow Style Language-**TinyGALS, Node-Level Simulators, NS-2 and its sensor network extension, TOSSIM.

Text Books:

1. Ad Hoc Wireless Networks – Architectures and Protocols, 1st edition, C. Siva Ram Murthy, B. S. Murthy, Pearson Education, 2004
2. Ad Hoc and Sensor Networks – Theory and Applications, 2nd edition *Carlos Corderio Dharma P.Aggarwal,* World Scientific Publications / Cambridge University Press, March 2006

Reference Books:

1. Wireless Sensor Networks: An Information Processing Approach, 1st edition*, Feng Zhao, Leonidas Guibas*, Elsevier Science imprint, Morgan Kauffman Publishers, 2005, rp2009
2. Wireless Ad hoc Mobile Wireless Networks – Principles, Protocols and Applications, 1st edition*,* Subir Kumar Sarkar, et al., Auerbach Publications, Taylor & Francis Group, 2008
3. Ad hoc Networking, 1st edition*,Charles E.Perkins*, Pearson Education, 2001
4. Wireless Ad hoc Networking, 1st edition*, Shih-Lin Wu, Yu-Chee Tseng,* Auerbach Publications, Taylor & Francis Group, 2007
5. Wireless Sensor Networks – Principles and Practice, 1st edition, Fei Hu, Xiaojun Cao, An Auerbach book, CRC Press, Taylor & Francis Group, 2010

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE09
Name of the Course	**Software Project Management (Professional Elective-III)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain conventional and agile software methods with their effect on software economics. **(K2)**
CO2: Explain modern software practices and identify life cycle phases and artifacts. **(K2)**
CO3: Describe software architectures, workflows, and checkpoints in iterative development. **(K2)**
CO4: Illustrate planning techniques and identify project structures and automation elements. **(K3)**
CO5: Describe software metrics and capability models to improve quality. **(K2)**

UNIT I: Conventional Software Management: The waterfall model, conventional software, Agile Methodology, Agiling capabilities, Management performance.
Evolution of Software Economics: Software Economics, pragmatic software cost estimation.
Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.
The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

UNIT II: Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.
Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

UNIT III: Model based software architectures: A Management perspective and technical perspective.
Work Flows of the process: Software process workflows, Iteration workflows.
Checkpoints of the process: Majormile stones, Minor Milestones, Periodic status assessments.
Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

UNIT IV: Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations. **Process Automation:** Automation Building blocks, The Project Environment.

UNIT V: Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation. **Process Capability Models:** Capability Maturity Model, Enhancing software Quality.

Textbooks:

1. Software Project Management, Walker Royce, PEA, 2005.
2. Software Engineering: A Practitioner's Approach by Roger S. Pressman and Bruce R. Maxim

Reference Books:

1. Software Project Management, Bob Hughes,3/e, Mike Cotterell, TMH
2. Software Project Management, Joel Henry, PEA
3. Software Project Management in practice, Pankaj Jalote, PEA, 2005,
4. Effective Software Project Management, Robert K.Wysocki, Wiley,2006.
5. Project Management in IT, Kathy Schwalbe, Cengage .

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE10
Name of the Course	**Natural Language Processing (Professional Elective-III)**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the fundamental concepts of language modeling, English morphology, and spelling correction techniques using regular expressions and finite-state automata. **(K2)**

CO2: Apply N-gram models and part-of-speech tagging techniques using stochastic and rule-based approaches to perform word-level language analysis. **(K3)**

CO3: Construct syntactic parsers using context-free grammars and apply probabilistic parsing methods such as PCFG and CYK for syntactic analysis. **(K3)**

CO4: Describe semantic representation techniques and methods for word sense disambiguation using supervise and knowledge-based approaches. **(K2)**

CO5: Apply discourse segmentation and reference resolution techniques using lexical resources like WordNet, Penn Treebank, and Brill's Tagger for effective text analysis. **(K3)**

UNIT I: Introduction: Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance.

UNIT II:Word Level Analysis: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part- of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

UNIT III: Syntactic Analysis: Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures

UNIT IV:Semantics And Pragmatics: Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

UNIT V:Discourse Analysis And Lexical Resources: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC).

Textbooks:

1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2nd Edition, Daniel Jurafsky, James H. Martin -Pearson Publication,2014.
2. Natural Language Processing with Python, First Edition, Steven Bird, Ewan Klein and Edward Loper, OReilly Media,2009.

Reference Books:

1. Language Processing with Java and Ling Pipe Cookbook, 1st Edition, Breck Baldwin, Atlantic Publisher, 2015.
2. Natural Language Processing with Java, 2nd Edition, Richard M Reese, OReilly Media,2015.
3. Handbook of Natural Language Processing, Second, NitinIndurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.Edition
4. Natural Language Processing and Information Retrieval, 3rd Edition, TanveerSiddiqui, U.S. Tiwary, Oxford University Press,2008.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE11
Name of the Course	**Big Data Analytics(Professional Elective-III)**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe big data concepts, applications, and related technologies. **(K2)**

CO2: Use NoSQL models and perform basic operations in Cassandra. **(K3)**

CO3: Develop data processing tasks using Hadoop and Hive. **(K3)**

CO4: Build data analytics workflows using Apache Spark. **(K3)**

CO5: Implement streaming and tuning techniques in Spark. **(K3)**

UNIT I: Introduction: Big data, convergence of key trends, unstructured data; **Industry examples of big data**: web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data; **Big data technologies**: Introduction to Hadoop, open source technologies, cloud and big data, mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.

UNIT II: Introduction to NoSQL; **Aggregate Data Models**: aggregates, key-value and document data models; **More Details on Data Models:** relationships, graph databases, schema less databases, materialized views; **Distribution models**: sharding, master-slave replication, peer- peer replication, combining sharding and replication; **Consistency**: relaxing consistency; **Version Stamps**; **Working with Cassandra**: Table creation, loading and reading data.

UNIT III: MapReduce: Data formats, analyzing data with Hadoop, scaling out, Data locality, Map Reduce Architecture, Process flow; **Hadoop distributed file system (HDFS)**: Architecture, fault tolerance with data replication, High availability, Java interface, data flow; **Hadoop I/O**: data integrity, compression, serialization; **Hive:** Introduction, data types and file formats, HiveQL data definition, HiveQL data manipulation, Logical joins, Window functions, Optimization, Table partitioning, Bucketing, Indexing, Join strategies.

UNIT IV: Apache spark- Advantages over Hadoop, lazy evaluation, In memory processing, DAG, Spark context, Spark Session, RDD, Transformations- Narrow and Wide, Actions, Data frames, RDD to Data frames, Catalyst optimizer, Data Frame Transformations, Working with Dates and Timestamps, Working with Nulls in Data, Working with Complex Types, Working with JSON, Grouping, Window Functions, Joins, Data Sources, Broadcast Variables, Accumulators, Deploying Spark- On-Premises Cluster Deployments, Cluster Managers- Standalone Mode, Spark on YARN , Spark Logs, The Spark UI-Spark UI History Server, Debugging and Spark First Aid.

UNIT V: Spark-Performance Tuning, Stream Processing Fundamentals, Event-Time and State full Processing - Event Time, State full Processing, Windows on Event Time- Tumbling Windows, Handling Late Data with Watermarks, Dropping Duplicates in a Stream, Structured Streaming Basics - Core Concepts, Structured Streaming in Action, Transformations on Streams, Input and Output.

Textbooks:

1. Big Data, Big Analytics: Emerging, Michael Minnelli, Michelle Chambers, and AmbigaDhiraj, 1st edition ,2013
2. SPARK: The Definitive Guide, Bill Chambers &MateiZaharia, O'Reilley, 2018-first Edition.
3. Business Intelligence and Analytic Trends for Today's Businesses", Wiley, First edition-2013.
4. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World Polyglot Persistence", Addison-Wesley Professional, 2012
5. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012

Reference Books:

1. Hadoop Operations, Eric Sammer, 1st Edition, O'Reilly
2. Programming Hive, E. Capriolo, D. Wampler, J. Rutherglen, O'Reilly
3. HBase: The Definitive Guide, Lars George, 1st Edition, O'Reilly
4. Cassandra: The Definitive Guide, Eben Hewitt, O'Reilly
5. Programming Pig, Alan Gates, 1st Edition, O'Reilly

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE12
Name of the Course	**Distributed Operating Systems(Professional Elective-III)**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe the basic concepts of distributed operating systems. **(K2)**
CO2: Understand the Communication using RPC. **(K2)**
CO3: Demonstrate the concept of shared memory. **(K3)**
CO4: Explain resource management techniques. **(K2)**
CO5: Illustrate Distributed file system and fault tolerance performance. **(K2)**

UNIT I:Fundamentals: What is Distributed Computing Systems? Evolution of Distributed Computing System; Distributed Computing System Models; What is Distributed Operating System? Issues in Designing a Distributed Operating System; Introduction to Distributed ComputingEnvironment(DCE).
Message Passing: Introduction, Desirable features of a Good Message Passing System, Issues in PC by Message Passing, Synchronization, Buffering, Multi-datagram Messages, Encoding and Decoding of Message Data, Process Addressing, Failure Handling, Group Communication, Case Study: 4.3 BSD UNIX IPC Mechanism.

UNIT II: Remote Procedure Calls: Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC

UNIT III: Distributed Shared Memory: Introduction, General Architecture of DSM systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM. Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms

UNIT IV:Resource Management: Introduction, Desirable Features of a Good Global Scheduling Algorithm, Task Assignment Approach, Load – Balancing Approach, Load – Sharing Approach Process Management: Introduction, Process Migration, Threads.

UNIT V: Distributed File Systems: Introduction, Desirable Features of a Good Distributed File System, File models, File–Accessing Models, File – Sharing Semantics, File – Caching Schemes, File Replication, Fault Tolerance, Atomic Transactions and Design Principles.

Textbooks:

1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.

Reference Books:

1. Andrew S. Tanenbaum: Distributed Operating Systems, Pearson Education, 2013.
2. Ajay D. Kshem kalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2008
3. Sunita Mahajan, Seema Shan, " Distributed Computing", Oxford University Press,2015.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	3	1.5	V23CSL10
Name of the Course	**Cloud Computing Lab**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Demonstrate various service types, delivery models and technologies of a cloud computing environment. **(K3)**

CO2: Classify cloud services based on virtual machines and containers in the cloud offerings. **(K2)**

CO3: Identify and outline key challenges associated with a cloud-based application. **(K2)**

CO4: Illustrate advanced cloud concepts such as server less computing and cloud simulation. **(K3)**

CO5: Apply suitable programming paradigms to solve real world and scientific problems using cloud services. **(K3)**

List of Experiments

1. Develop and deploy basic web servicesss
2. Develop IPC mechanisms and message communication using publish/subscribe model.
3. Demonstrate Configuration and Installation of VirtualBox/VMware Workstation with different flavors of Linux or windows OS on top of windows8 or above.
4. Demonstrate Installation of C compiler in the virtual machine created using VirtualBox and execute Simple Programs.
5. Demonstrate Creation of an Amazon EC2 instance and set up a webserver on the instance and associate an IP address with the instance. In the process, create a security group allowing access to port 80 on the instance.

OR

6. Demonstrate EC2 setup on OpenStack environment (*Hint: perform experiment 5 using OpenStack)*
7. Demonstrate Installation of Google App Engine. Create a Hello World app and other simple web applications using python/java.
8. Operate a Docker container and set up a webserver (e.g. apache2 or Python based Flask micro web framework) on the instance. Map the host directory as a data volume for the container.
9. Demonstrate file transfer from one virtual machine to another virtual machine. Similarly, from one container to another container.
10. Operate and verify virtual machine on Trystack. (Online Openstack Demo Version)
11. Demonstrate Installation of Hadoop single node cluster and run simple applications like word count.
12. Employ OpenFaaS – Serverless computing framework and demonstrate basic event driven function invocation.
13. Demonstrate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in Cloud Sim.

Textbooks:

1. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, 2nd Edition, McGraw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Reference Books:

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
2. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
3. Online documentation and tutorials from cloud service providers (e.g. AWS, Google App Engine)
4. Docker, Reference documentation, https://docs.docker.com/reference/
5. OpenFaaS, Serverless Functions Made Simple, https://docs.openfaas.com/

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	3	1.5	V23CSL11
Name of the Course	**Machine Learning Lab using Python**					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After successful completion of the course, the student will be able to:

CO1: Identify various Python libraries used in Machine Learning. **(K2)**
CO2: Construct non-probabilistic classifiers using Python Programming. **(K3)**
CO3: Implement probabilistic classifiers using Python Programming. **(K3)**
CO4: Apply ensemble methods for classification tasks. **(K3)**
CO5: Demonstrate neural and sequential models using Python. **(K3)**

LIST OF EXPERIMENTS

1. Introduction to required python libraries such as Numpy, Pandas, Scipy, Matplotliband Scikit-learn.
2. Import, preprocess, and split the datasets using scikit-learn.
3. Implement the KNN algorithm for classification and demonstrate the process of finding out optimal "K" value using Python Programming.
4. Demonstrate the process of the Decision Tree construction for classification problems using python programming.
5. Construct a classification model using the Bayes classifier using Python Programming.
6. Implement a Logistic Regression algorithm for binary classification using Python Programming.
7. Construct an SVM classifier using python programming.
8. Demonstrate the working of Multi-layer perceptron with MLPClassifier() using Python programming.
9. Implement an Ensemble Learner using Random Forest Algorithm using python programming.
10. Implement an Ensemble Learner using Adaboost Algorithm using Python programming.
11. Construct and Analyze a Bayesian Network using pgmpy for Probabilistic Inference.
12. Implement and Evaluate a Hidden Markov Model using hmmlearn for Sequential Pattern Recognition.

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024
2. "Machine Learning: An Algorithmic Perspective, Stephen Marsland, 2nd Ed., CRC Press.
3. "Introduction to Machine Learning with Python", Andreas C. Muller and Sarah Guido, First Edition, O'Reilly.

Reference Books:

1. "Practical Machine Learning with Python", Dipanjan Sarkar, Raghav Bali and Tushar Sharma, First Edition, A Press.

Semester	**VI**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	1	2	2	V23CSSE04
Name of the Course	**Master Coding and Competitive Programming - Part-2**					
Branch	Common to CSE, CST, CAI, AIM and CDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Apply Divide and Conquer algorithm technique to solve complex in logarithmic time. **(K3)**

CO2: Apply Greedy method to solve Optimization and decision making problems. **(K3)**

CO3: Apply Backtracking Algorithm technique to find combinatorial problems. **(K4)**

CO4: Experiment with Dynamic Programming Algorithm technique to solve Problems that uses Optimal substructures. **(K3)**

CO5: Develop programs using Linked List Graphs, DFS and BFS techniques. **(K3)**

List of Experiments

1. Develop Programs to solve problems based on Divide and Conquer Algorithm Technique.
2. Develop programs using two pointer and sliding window algorithms.
3. Illustrate Problem Solving using Greedy Algorithm technique.
4. Illustrate Problem Solving using Backtracking.
5. Develop programs using Dynamic Programming and Kadane Algorithm.
6. Develop programs using Linked List and its applications.
7. Develop programs using Graphs and Graph Searching Techniques.

Textbooks:

1. Introduction to Algorithms, Second Edition, Thomas H. Cormen Charles E. Leiserson.

2. Data Structures and Algorithms Made Easy: Narasimha Karumanchi .

3. The Algorithm Design Manual, Springer series, Steven Skiena.

Tools:

1. practice.geeksfoegeeks.com
2. leetcode.com
3. codingninjas.com
4. Hackerrank.com
5. Interviewbit.com

Semester	**V to VII SEM**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTPE13
Name of the Course	Computer Organization and Architecture **(Professional Elective-I)**					
Branch	Common to ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the basic structure of digital computers, Von Neumann architecture, and data representation techniques. **(K2)**

CO2: Describe the instruction cycle, basic computer organization, and programming using machine and assembly language. **(K2)**

CO3: Understand the working of the CPU, instruction formats, addressing modes, and computer arithmetic operations. **(K2)**

CO4: Explain input-output organization, including data transfer methods and interrupt mechanisms. **(K2)**

CO5: Describe different types of memory and their organization, including cache, associative, and virtual memory. **(K2)**

UNIT-I : Introduction: Digital Computers, Von Neumann computers, Basic organization of a computer, **Data Representation:** Data types, Complements, Fixed-point representation, Conversion of fractions, Floating-point representation.

Register Transfer and Micro operations: Register transfer language, Register transfer, Bus and Memory transfers, Arithmetic Micro operations, Logic Micro operations, Shift Micro operations, Arithmetic Logic Shift Unit.

UNIT-II: Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference instructions, Input-Output and Interrupt, Complete Computer Description, Design of Basic computer

Programming the Basic Computer: Introduction, Machine Language, Assembly language, The Assembler, Program Loops, Programming Arithmetic and Logic Operations

Micro programmed Control: Control Memory, Address Sequencing, Microprogram Example, Design of Control Unit (**Preferably from Reference Book 2**)

UNIT-III: Central Processing Unit: Introduction, General Register Organization, Stack organization, Instruction Formats, Addressing Modes, Data transfer and Manipulation, Program Control, Reduced Instruction Set Computer.

Computer Arithmetic:Introduction, Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating-Point Arithmetic Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations.

UNIT-IV: Input-Output organization : Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), Input-Output Processor (IOP), Serial Communication.

UNIT-V: Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware.

Text Book:

1. M.Morris Mano," Computer System Architecture," Pearson Publishers, Revised Third Edition

Reference Books

1. John P Hayes, "Computer Architecture and Organization,"Mc-Graw Hill Publishers, Third Edition
2. Carl Hamacher, "Computer Organization," Tata Mc-Graw Hill Publishers, Fifth Edition.

Annexure-III

List of Job Oriented Elective Courses

List of Job Oriented Elective Courses through offline				List of Job Oriented Elective Courses through MOOCS		
S.No.	**Course Code**	**Course Name**	**Credits**	**Course Name** (offered by NPTEL MOOCS)	**Number of Weeks**	**Credits**
1.	V23CSTJE01	Python for Data Science	3	Data Science Using Python	12	3
2.	V23CSTJE02	Ethical Hacking	3	Ethical Hacking	12	3
3.	V23CSTJE03	Java Full Stack Technologies	3	-	-	-
4.	V23CSTJE04	Shell and System Programming	3	-	-	-
5.	V23CSTJE05	Agentic AI	3	-	-	-

NOTE: All the Job oriented can be theory / Lab Course

Semester	**V to VII**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V20CSTJE01
Name of the Course	Python for Data Science					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Apply NumPy for array manipulation and numerical operations. **(K3)**
CO2: Use Pandas for data cleaning and analysis. **(K3)**
CO3: Demonstrate data visualization techniques using Python matplotlib. **(K3)**
CO4: Prepare Exploratory Data Analysis (EDA) to derive insights. **(K3)**

List of Experiments

1. NumPy for Numerical Computing

- Create and manipulate NumPy arrays:
  - Array creation, slicing, indexing, reshaping.
  - Statistical operations (mean, median, std).
- Perform matrix operations using NumPy:
  - Matrix addition, multiplication, transpose, inverse.
- Generate random numbers and perform element-wise operations.

2. Pandas for Data Analysis

- Create and manipulate **Series** and **DataFrames**.
- Import and explore real-world data (CSV/Excel).
- Perform:
  - Data cleaning (handling missing values).
  - Filtering, grouping, merging, and concatenation.
- Perform descriptive statistics and correlation analysis using Pandas.

3.Data Visualization with Matplotlib & Seaborn

- Plot different types of charts using **Matplotlib**:
  - Line, Bar, Histogram, Scatter, Pie.
- Visualize datasets using **Seaborn**:
  - Pairplot, Heatmap, Boxplot, Countplot.
- Customize plots (titles, labels, legends, grid, style).

4.Exploratory Data Analysis (EDA) Case Study

- Perform a complete **EDA** on a given dataset (e.g., Titanic, Iris, or Sales Data):
  - Load data using Pandas.
  - Clean and preprocess data.
  - Analyze patterns using visualization.
  - Compute insights and summarize results.

Case Study 1: Prediction of Student Academic Performance using Data Science Techniques.

Problem Statement: Educational institutions often face challenges in identifying students who are likely to underperform. Using data science, we can analyze patterns in student behavior and academic data to predict their performance early — enabling timely intervention and support.

Dataset Description:

Column Name	Description
Hours_Studied	Average daily study hours
Attendance	Percentage of attendance
Previous_Score	Average score in past tests
Extra_Curricular	Participation in activities (Yes/No)
Final_Score	Final exam score (target variable)

Sample Data (CSV format):

Hours_Studied	Attendance	Previous_Score	Extra_curricular	Final_Score
2.5	60	45	No	50
5.0	80	78	Yes	82
3.2	75	60	No	65
6.0	90	88	Yes	90
1.5	55	40	No	45

Case Study 2: Sales Forecasting using Data Science

Problem Statement: Retail stores often struggle to forecast demand accurately, leading to problems such as overstocking or stockouts. Using data science techniques, we aim to analyze past sales data and predict future sales to help improve inventory management, marketing decisions, and revenue planning.

Dataset Description:

Column Name	Description
Date	Sales date
Store_ID	Unique identifier for each store
Items_Sold	Number of items sold
Customers	Number of customers visiting the store
Promo	Whether a promotion was active (1=Yes, 0=No)
Holiday	Whether the day was a holiday (1=Yes, 0=No)
Sales	Total daily sales amount (Target variable)

Example Data (CSV):

	Store_ID	Items_Sold	Customers	Promo	Holiday	Sales
2023-01-01	101	120	85	1	0	15000
2023-01-02	101	95	70	0	0	12000
2023-01-03	101	140	90	1	0	17500
2023-01-04	101	60	45	0	1	9000
2023-01-05	101	130	88	1	0	16800

Case Study 3: Customer Segmentation using K-Means Clustering

Problem Statement: Businesses often treat all customers the same way, leading to ineffective marketing and poor customer retention. Using data science and clustering techniques, we can segment customers based on features like spending score, income, and age — helping companies to understand their customer base better.

Dataset Description (Mall Customers Dataset)

Column Name	Description
CustomerID	Unique ID for each customer
Gender	Male/Female
Age	Age of the customer
Annual Income (k$)	Annual income in thousand dollars
Spending Score (1–100)	Score assigned based on spending habits

Example Data:

Customer ID	Gender	Age	Annual Income (k$)	Spending Score (1–100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

Reference Books:

1. Martin C. Brown,Python: The Complete Reference, 4th Edition, McGraw-Hill Education, 2018.
2. Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, 2022.
3. Mukhiya, S. K., & Ahmed, U. (2020). Hands-on exploratory data analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data. Packt Publishing.
4. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow,3rd Edition, Packt Publishing, 2019.

Semester	**V to VII**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V20CSTJE02
Name of the Course	Ethical Hacking					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe ethical hacking concepts, phases, and legal aspects. **(K1)**
CO2: Explain reconnaissance, scanning, and enumeration techniques. **(K2)**
CO3: Apply open-source tools for vulnerability assessment and password analysis. **(K3)**
CO4: Demonstrate web application testing using OWASP tools in a controlled lab. **(K3)**
CO5: Apply basic malware analysis and prepare a penetration testing report. **(K3)**

UNIT I: Introduction to ethical hacking: concepts, need, and legality, Phases of hacking and ethical hacker roles, setting up a virtual lab using VirtualBox and Kali Linux (open-source), Network configuration (NAT and Host-only), **Information gathering:** OSINT, DNS, WHOIS, Shodan, Google Dorks, **Tools:** Nmap, Recon-ng, theHarvester, Maltego CE.

UNIT II: Scanning techniques: port scanning, service detection, version detection, **Enumeration:** SMB, SNMP, NetBIOS, Packet capturing and traffic monitoring using Wireshark and tcpdump, Introduction to intrusion detection systems (theory). **Tools:** Nmap, Zenmap, Wireshark, tcpdump.

UNIT III: Vulnerability Assessment and Password Security: Basics of vulnerability scanning and reporting, OpenVAS (Greenbone Community Edition) for vulnerability assessment, Password policies and hashing mechanisms, Applying Hashcat and John the Ripper for password strength analysis, Basics of Denial-of-Service (theory only).

UNIT IV: Web Application Security and OWASP Top 10: Overview of OWASP Top 10 vulnerabilities, Web application testing using OWASP ZAP and Burp Suite Community Edition, SQL Injection, XSS, CSRF, Authentication flaws (on DVWA, Juice Shop), Safe testing and mitigation techniques. **Tools:** OWASP ZAP, Burp Suite Community, DVWA, Juice Shop, sqlmap.

UNIT V: Post-Exploitation, Malware Analysis, and Reporting : Post-exploitation concepts (privilege escalation, maintaining access — theory only), Introduction to malware behavior and sandboxing, Basic malware analysis using **Ghidra** (open-source), Introduction to digital forensics using **Autopsy/Sleuth Kit,** Report writing and remediation documentation.

Textbooks:

1. Georgia Weidman, *Penetration Testing: A Hands-On Introduction to Hacking*, No Starch Press.
2.Dafydd Stuttard& Marcus Pinto, *The Web Application Hacker's Handbook*, Wiley.

Reference Books:

1. Peter Kim. *The Hacker Playbook 3: Practical Guide to Penetration Testing.* 3rd edition. ISBN 978-1980901754.
2. Jon "Smibbs" Erickson. *Hacking: The Art of Exploitation (2nd Edition).* No Starch Press. ISBN 978-1593271442.

Semester	**V to VII**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V20CSTJE03
Name of the Course	**Java Full Stack Technologies**					
Branch	Common to CSE and CST					

Syllabus Details

After Successful completion of the Course, the students will be able to:

CO1: Develop programs using Servlets. **(K3)**
CO2: Illustrate MVC architecture. **(K3)**
CO3: Demonstrate applications of Hibernate. **(K3)**
CO4: Illustrate Spring MVC Framework. **(K3)**

Exercise 1: Understanding about Servlets: Create Example programs Using the below concepts

- Introduction to Servlets.
- Develop Servlet application to print current date & time.
- Develop Servlet program to link Html & Servlet Communication.
- Develop Servlet program to Auto refresh a page.
- Demonstrate session tracking.
- Develop Servlet program to insert/delete/update the record into database.
- Develop Servlet program to add cookie to selected value

Exercise 2: Understanding about Model View Controller : Create Example programs Using the below concepts

- Introduction to MVC in java.
- Develop sample program on Model Layer in MVC Using Java.
- Develop sample program on View Layer in MVC Using Java.
- Develop sample program on Controller Layer in MVC Using Java.
- Demonstrate MVC deployment in java.
- Rules for MVC Mapping in Server Side.
- How to use Web Server for MVC Deployment.

Exercise 3: Understanding about Hibernate : Create Example programs Using the below concepts

- Introduction to Hibernate.
- What is ORM?Demonstrate the components of Hibernate
- How to persist objects using Hibernate and how to use map using XML and Annotations
- Working with relationship between entities-association
- Transactions in Hibernate
- Querying with HQL (Hibernate Query Language)
- Various other forms of querying - Criteria, QBE etc.
- Understanding Some Debugging Tools in Java NetBeans, Eclipse, IntelliJ IDEA, Visual Studio Code.

Exercise 4: Develop SpringBoot applications for the following case studies

- **Case Study A: Online Course Management System (OCMS):**Build a scalable Online Course Management System (OCMS) where administrators can upload courses, and students can register, browse, enroll, and track their learning progress.

Core Modules

- **Student Module:** Register and update personal details (name, email, contact),Enroll in courses, Track course progress and view completed courses
- **Course Module:** Create and manage courses (title, description, category, instructor),Assign content like videos, PDFs, quizzes, List all available courses for students
- **Instructor Module:** Register as an instructor, Add and manage courses they're Teaching, View enrolled students and track their performance.
- **Enrollment Module:** Links students to courses, Stores date of enrollment and status (ongoing, completed),Provides history of courses taken by the student.
- **Notification Module:** Sends alerts for new course additions or updates, Reminds students of pending or upcoming coursework, Notifies instructors of newly enrolled students
- **User Authentication & Role Module:** Handles login/signup for both students and instructors, Defines role-based access (e.g. only instructors can create courses),Secures API with Spring Security.

- **Case Study B: Smart Health Monitoring System (SHMS):** Develop a Smart Health Monitoring System for tracking patient vitals (like heart rate, blood pressure, oxygen levels, etc.), enabling doctors to monitor patients remotely in real time.

Core Modules

- **Patient Module:** Register and update personal and medical details
- **Vital Stats Module:** Stores real-time body parameters: blood Pressure, oxygen Level, temp, etc.
- **Alert Module:** Raises alerts if parameters fall outside safe range
- **Doctor Module:** Assigns doctors to patients and shows dashboard

Reference Books:

1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006.
2. Murach's Java Servlets and JSP, 3rd Edition by (Murach: Training & Reference) 3rd Edition.
3. Spring and Hibernate Paperback – 1 July 2017 by K. Santosh Kumar.
4. Full Stack Java Development with Spring MVC, Hibernate, jQuery, and Bootstrap by Mayur Ramgir,Wiley.

Semester	**V to VII**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V20CSTJE04
Name of the Course	Shell and System Programming					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Demonstrate the basic knowledge of Linux commands and utilities by using Linux shell environment. **(K3)**

CO2: Experiment with the Concept of shell Programming on Files and Directories. **(K3)**

CO3: Develop C programs using system calls to perform file handling operations, directory manipulation, and execution of UNIX commands. **(K3)**

CO4: Demonstrate concepts of inter-process communication and process management using pipes and system calls. **(K3)**

CO5: Demonstrate understanding of process states by creating and analyzing zombie and orphan processes, and develop programs involving time and concurrent execution. **(K3)**

LIST OF EXPERIMENTS

1. Experiment the following Unix Commands:
   a) **General Purpose Utilities:** cal, date,man,who,whoami,echo,bc,clear, uname, passwd.
   b) **Directory Handling Commands:** pwd,cd,mkdir,rmdir.
   c) **File Handling Utilities:** cat,touch,cp,ls,rm,mv,nl,wc,tar
   d) **File Permissions:** Chmod, umask
   e) **Displaying Commands:** more, head, tail
   f) **Filters:** cmp, comm., ulink, diff, head, tail, find, cut, paste,sort, uniq, tr
   g) **Disk Utilities:** du,df
   h) **Process Utilities**–ps, kill.
   i) **Networking Utilities**– ping, telnet, rlogin, ftp, finger
2. Develop a shell script that that copies multiple files to a directory.
3. Develop a shell script that counts the number of lines and words present in a given file.
4. Develop a shell script that displays the list of all files in the given directory.
5. Develop a shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two division options: one returns the quotient and the other returns reminder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add(-a), subtract(-s),multiply (-m), quotient (-c) and reminder (-r).
6. Develop a C program that counts the number of blanks in a text file.
   a) Using standard I/O b)Using system calls.
7. Develop in C the following Unix commands using system calls.
   a) cat b) ls c)mv
8. Develop a program that takes one or more file/directory names as command line input and reports the following information on the file:
   a) File type.
   b) Number of links.
   c) Time of last access.
   d) Read, Write and Execute permissions.
9. Develop a C program that illustrates uses of the mkdir, opendir, readdir, closedir, and rmdir APIs.
10. Develop a C program that illustrates how to execute two commands concurrently with a command pipe.

11. Develop a C programs that illustrates the following:
    a) Two-way communication with unidirectional pipes.
    b) Two-way communication with bidirectional pipes
12. Develop a C program that illustrates the creation of child process using fork system call.
13. Develop a C program to create a Zombie process.
14. Develop a C program that illustrates how an orphan is created
15. Develop a C program that displays the real time of a day every 60 seconds.

Text Books:

1. UNIX and Shell Programming: A Textbook, Behrouz A. Forouzan | Richard F. Gilberg, Cengage Learning.
2. UNIX: Concepts and Applications, Sumithaba Das, 4th Edition, Tata McGrawHill.
3. Unix & Shell Programming, M.G.Venkatesh Murthy, Pearson Education.
4. W. Richard Stevens, "Advanced programming in the unix environment", 3rd Edition, Pearson education

Semester	**V to VII**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V20CSTJE05
Name of the Course	Agentic AI					
Branch	Common to CSE and CST					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the principles of agentic AI covering agent loops, tool selection, and basic routing. **(K2)**

CO2: Apply function-calling and tool integration to build single-agent workflows for campus and helper tasks. **(K3)**

CO3: Develop retrieval-based agents using LangChain and stateful correction loops with LangGraph. **(K3)**

CO4: Construct agentic RAG pipelines that ingest multi-document corpora and generate grounded answers with citations.**(K3)**

CO5: Operate API-enabled and web-served agents with basic logging and evaluation for task success, latency, and cost.**(K3)**

Exercise 1: Introduction to Agentic AI + Query Router Agent

Topics: What is an Agent vs Chatbot, Agent loop (observe–decide–act), Tool/function calling concept, Intent classification, Prompts for routing, Action selection logic, Simple agent flow in Python, Campus automation domain framing, Console interaction design, Limitations of basic agents.

Practice: Build a Campus Query Router Agent that classifies a question as Academic, Exam, or General and triggers the respective stub function output.

Assignment: Implement a Python agent that:

- Takes user query
- Uses LLM to classify intent
- Calls the appropriate function (academic/exam/general)
- Prints tool response

Exercise 2: Basic Tool Use & Function Calling for Agents

Topics: System and task prompts, Function-calling concept in agents, Designing and wrapping tools in Python, Reason-to-action execution flow, Classifying user requests into tool categories, Command-line agent interaction, Handling basic math/code/grammar queries, Introducing human-in-loop confirmation logic, Error-safe tool invocation, Understanding limits of rule-based tool routing vs LLM-driven routing

Practice: Build a Student Helper Agent that can understand whether a user needs math help, Python syntax support, or English grammar correction, and then call the appropriate tool function to produce a relevant response.

Assignment: Implement a Python agent that:

- Takes a user query
- Uses an LLM to decide which tool to invoke
- Calls one of the tool functions (math, python_help, english_grammar)
- Prints the selected tool's response to the user

Exercise 3: LangChain Basics & Simple Retrieval Agent

Topics: LangChain introduction, Chains and tools, PromptTemplate basics, Loading text data, Text splitting concepts, Embeddings fundamentals, Vector databases overview (Chroma intro), Document retrieval flow, Retrieval + response pipeline design, Integrating LLM with Python functions, Simple evaluation of retrieved results

Practice: Build a simple Retrieval Agent using LangChain that loads a text document (ex: syllabus or academic guidelines), splits it, embeds it, and answers student questions from that content using a basic retrieval chain.

Assignment: Implement a LangChain-based Python agent that:

- Loads a PDF/text document (campus syllabus or rules)
- Converts it into searchable chunks
- Uses vector search to retrieve the most relevant chunk
- Generates an answer for a student query using the retrieved text

Exercise 4: LangGraph Basics & Self-Correction Loop

Topics: Introduction to LangGraph, State and workflow node concepts, Graph-based agent control, Reflection and retry pattern, Human-approved execution flow, Message/state trimming, Controlled agent pipeline vs direct generation, Designing self-correction cycles, Principles of safe iterative improvement

Practice: Build a Self-Correction Coding Agent using LangGraph that takes a simple coding task, generates code, checks if the output contains an error message, and then revises its response in a second attempt (simulating an autonomous correction loop).

Assignment: Implement a LangGraph agent that:

- Accepts a small programming question (e.g., Python list reverse, factorial, file reading)
- Generates the code answer
- Detects basic errors in its own output (simple text check like "Traceback" or "Error")
- Attempts to correct the response and present the improved code to the user

Exercise 5: Agentic RAG With Structured Retrieval

Topics: RAG vs Agentic RAG, Document chunking strategies, Embedding selection basics, Context-controlled retrieval, Query rewriting behavior, Multi-step retrieval prompts, Preventing irrelevant recall, Citing retrieved sources, Answer confidence estimation, Limitations of naïve RAG and need for agent orchestration

Practice: Build a Campus Information RAG Agent that retrieves content from multiple documents (such as academic handbook, exam rules, and department brochures) and generates context-grounded answers with supporting text snippets.

Assignment: Implement a multi-document RAG system that:

- Loads two or more campus-related documents
- Embeds and stores them in a vector store
- Retrieves context for a student query
- Answers using retrieved text and displays reference snippets

Exercise 6: Multi-Agent Collaboration Basics

Topics: Multi-agent design principles, Agent roles and task delegation, Coordinator vs worker agents, Message-passing fundamentals, Agent role prompting, Task pipeline creation, Handling conflicting outputs, Multi-agent chat flows, Simple evaluation of cooperative agents, Practical multi-agent scenarios in campus settings

Practice: Build a two-agent system where one agent acts as a "Research Agent" to gather bullet points from a topic and a second agent acts as a "Refiner Agent" to turn those points into a well-structured short explanation.

Assignment: Implement a two-agent pipeline that:

- Takes a user topic
- Generates bullet points using the first agent
- Converts them into a short explanatory answer using the second agent
- Delivers the refined output to the user

Exercise 7: API-Enabled Agent for External Automation

Topics: API calls from agents, Python tool execution on demand, Web data extraction basics, JSON response parsing, Email/sheets automation concept, Scheduling simple actions, Safety in executing commands, User approval before execution, Integrating external knowledge sources

Practice: Build a Web-Lookup Agent that fetches simple information from a public API (such as weather, news headline, or university database endpoint) and returns structured results to the user with a short natural-language explanation.

Assignment: Implement an API-powered agent that:

- Receives a query requiring external information
- Calls a public API to retrieve the relevant data
- Parses the response and formats a readable answer
- Displays both raw and explanatory outputs

Exercise 8: Deploying and Evaluating an Agent

Topics: Basic FastAPI integration, Converting agent logic into an API endpoint, HTTP request/response design for agents, Basic logging and trace printing, Response formatting, Monitoring agent behavior, Simple evaluation metrics (accuracy, relevance, latency), Deployment considerations for student projects

Practice: Convert one previous week's agent into a simple web API using FastAPI, invoke it from a browser or command-line client, and observe the structured JSON output along with logs.

Assignment: Deploy an agent as an API that:

- Defines a /query endpoint
- Accepts a user question and routes to agent logic
- Returns the answer in structured JSON format
- Prints logs for each request and response

Textbooks:

1. "An Introduction to MultiAgent Systems", Michael Wooldridge, Wiley, 2nd Ed, 2009.
2. "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations", Yoav Shoham, Kevin Leyton-Brown, Cambridge University Press, 1st Ed, 2009.
3. "Multiagent Systems", Gerhard Weiss (Ed.), MIT Press, 2nd Ed, 2013.
4. "Designing Data-Intensive Applications", Martin Kleppmann, O'Reilly Media, 1st Ed, 2017.
5. "Generative AI with LangChain: Build production-ready LLM applications and advanced agents usingPython, LangChain, and LangGraph", Ben Auffarth, Leonid Kuligin, Packt Publishing, 2nd Ed, 2025.

References:

1. "LangChain Documentation", https://docs.langchain.com/
2. "LangChain Python Reference", https://reference.langchain.com/python/
3. "LangGraph Documentation", https://langchain-ai.github.io/langgraph/
4. "OpenAI API – Function (Tool) Calling", https://platform.openai.com/docs/guides/function-calling
5. "OpenAI Assistants – Overview", https://platform.openai.com/docs/assistants/overview
6. "CrewAI Documentation", https://docs.crewai.com/
7. "FastAPI – Official Documentation", https://fastapi.tiangolo.com/

Annexure-IV

Open Electives

The following courses are offered to the students of other departments.

List of Open Elective Courses through offline				List of Open Elective Courses through MOOCS		
S.No.	**Course Code**	**Course Name**	**Credits**	**Course Name** (offered by NPTEL MOOCS)	**Number of Weeks**	**Credits**
1.	V23CSTOE01	Python Programming Lab	3	The Joy of Computing using Python	12	3
2.	V23CSTOE02	Object Oriented Programming Through Java Lab	3	Programming In Java	12	3
3.	V23CSTOE03	Database Management Systems Lab	3	Introduction to Database Systems	12	3
4.	V23CSTOE04	Cryptography and Network Security	3	Cryptography and Network Security	12	3
5.	V23CSTOE05	Operating Systems	3	Operating System Fundamentals	12	3
6.	V23CSTOE06	Computer Networks	3	Computer Networks and Internet Protocol	12	3
7.	V23CSTOE07	Data Mining	3	Data Mining	12	3
8.	V23CSTOE08	Software Engineering	3	Software Engineering	12	3
9.	V23CSTOE09	Cyber Security	3	Cyber Security & Privacy / Privacy and Security in Online Social Media	12	3
10.	V23CSTOE10	Human Computer Interaction	3	Human-Computer Interaction / Design & Implementation of Human Computer Interfaces	12	3

Semester	**III**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V23CSTOE01
Name of the Course	**Python Programming Lab**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Illustrate basic concepts and control structures in python Programming. **(K3)**
CO2: Demonstrate functions and packages. **(K3)**
CO3: Construct python programs using structured data types. **(K3)**
CO4: Develop programs on Files, Exception handling and OOPs Concepts. **(K3)**
CO5: Construct programs for Data Analysis using Num Py and Pandas. **(K3)**

UNIT–I: History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter Notebook.

Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language.

Control Flow Statements: if statement, if-else statement, if...elif…else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement.

Sample Experiments:

1. Develop a program to find the largest element among three Numbers.
2. Develop a Program to display all prime numbers within an interval
3. Develop a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.
   i) Arithmetic Operators ii) Relational Operators iii) Assignment Operators iv) Logical Operators v) Bit wise Operators vi) Ternary Operator vii) Membership Operators viii) Identity Operators
5. Develop a program to add and multiply complex numbers
6. Develop a program to print multiplication table of a given number.

UNIT–II: Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.

Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.

Sample Experiments:

1. Develop a program to define a function with multiple return values.
2. Develop a program to define a function using default arguments.
3. Develop a program to find the length of the string without using any library functions.
4. Develop a program to check if the substring is present in a given string or not.

5. Develop a program to perform the given operations on a list:
   i. addition ii. Insertion iii. slicing
6. Develop a program to perform any 5 built-in functions by taking any list.

UNIT–III: Dictionaries: Creating Dictionary, Accessing and Modifying key: value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement.

Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozenset.

Sample Experiments:

1. Develop a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
2. Develop a program to count the number of vowels in a string (No control flow allowed).
3. Develop a program to check if a given key exists in a dictionary or not.
4. Develop a program to add a new key-value pair to an existing dictionary.
5. Develop a program to sum all the items in a given dictionary.

UNIT–IV: Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.

Sample Experiments:

1. Develop a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper-case words from source must be lowered.
2. Develop a Python program to print each line of a file in reverse order.
3. Develop a Python program to compute the number of characters, words and lines in a file.
4. Develop a program to create, display, append, insert and reverse the order of the items in the array.
5. Develop a program to add, transpose and multiply two matrices.
6. Develop a Python program to create a class that represents a shape. Include methods tocalculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.

UNIT–V: Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas.

Sample Experiments:

1. Develop a Python program to check whether a JSON string contains complex object or not.
2. Demonstrate NumPy arrays creation using array () function.
3. Demonstrate use of ndim, shape, size, dtype.
4. Demonstrate basic slicing, integer and Boolean indexing.
5. Develop a Python program to find min, max, sum, cumulative sum of array
6. Prepare a dictionary with at least five keys and each key represent value as a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows:
    a) Apply head () function to the pandas data frame
    b) Perform various data selection operations on Data Frame
7. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attribute with scatter and plot operations in matplotlib.

Reference Books:

1. Gowrishankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2ndEdition, Pearson, 2024
3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Online Learning Resources:

1. https://www.coursera.org/learn/python-for-applied-data-science-ai
   https://www.coursera.org/learn/python?specialization=python#syllabus.

Semester	**V to VII SEM**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V23CSTOE02
Name of the Course	Object Oriented Programming Through Java Lab(**Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Use code editors and JDK tools to write, compile, and run Java programs. **(K3)**
CO2: Use control statements and arrays while programming. **(K3)**
CO3: Develop programs using classes and objects. **(K3)**
CO4: Use inheritance, interfaces and packages while developing programs in Java. **(K3)**
CO5: Develop exception-handling and multithreaded programs. **(K3)**

Syllabus:

CYCLE–I:Overview of Object-oriented Programming: Introduction to Object-oriented Programming, Principles of Object-oriented Programming Languages, and Applications of OOP.

Introduction to Java: History of Java, Java Features, Java Virtual Machine, Java Program Structure, Literals, Identifiers, Primitive Data types, Variables, Operators and Expressions, Operator Precedence and Associativity, Type Conversion and Casting.

Exercises

a) Develop a Java program to display the default values of all primitive data types of Java.
b) Construct a Java program that calculates the area of a triangle, given the lengths of all three sides.

Area =$\sqrt{(S\,(S-a)\,(S-b)\,(S-c))}$, where S = (a+b+c)/2.

CYCLE– II: Control Statements: Conditional Statements - if, switch; Iteration Statements - while, do-while, for, for-each version of for; Jump Statements - break, continue, return.

Arrays: Introduction to Arrays, Array Declaration and Initialization, One-Dimensional Arrays, Multi-Dimensional Arrays, Basic String Handling.

Exercises

a) Develop a Java program that displays
   i) The roots of a quadratic equation $ax^2+bx+c=0$
   ii) The nature of roots by calculating the discriminate D.
b) N bikers compete in a race such that they drive at a constant speed, which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all N racers. Take as input, the speed of each racer and print back the speed of qualifying racers.
c) Develop a Java program that displays the name of the day, based on the value of day, using the switch statement.
d) Develop a Java program to search for an element in a given list of elements using Linear Search.
e) Develop a Java program to perform multiplication of two matrices.
f) Develop a Java program using StringBuffer to perform various operations on a string.

CYCLE– III: Introduction to Classes and Objects: General Form of a Class, Methods, Declaring Objects using new, Constructors, this Keyword, Understanding static, Method and Constructor Overloading, Using Command-Line Arguments, Garbage Collection.

Exercises

a) Construct a Java program to demonstrate class mechanism - Create a class that contains variables, methods, constructors and invoke those methods inside main().
b) Develop a Java program demonstrating the use of static variables, methods.
c) Develop a Java program demonstrating the use of this keyword.
d) Develop a Java program that implements method overloading.
e) Develop a Java program that implements constructor overloading.
f) Develop a Java program demonstrating the use of command-line arguments.

CYCLE– IV: Inheritance: Access Control, Introduction to Inheritance, Types of Inheritance, Using super, Method Overriding and Dynamic Method Dispatch, Using final, Abstract Classes.
Interfaces: Defining and Implementing Interfaces. **Packages:** Creating Packages, Importing Packages, Importance of CLASSPATH.

Exercises

a) Construct a Java program to demonstrate single inheritance.
b) Construct a Java program to demonstrate multi-level inheritance.
c) Construct a Java program that illustrates the use of super.
d) Develop a Java program that illustrates runtime polymorphism.
e) Develop a Java program that uses an abstract class to find areas of different shapes.
f) Develop a Java program using interfaces. In addition, use interfaces to achieve multiple inheritance.
g) Construct a Java program that creates a user-defined package. Use the package by importing it in another Java program.

CYCLE– V: Exception Handling: Exception-Handling Fundamentals, Using try and catch, Using throw, Using throws and finally, User-defined Exceptions.

Exercises

a) Develop a Java program to demonstrate exception-handling mechanism using try/catch. Use multiple catch clauses.
b) Construct a Java program for illustrating the use of throw.
c) Construct a Java program for illustrating the use of finally.
d) Construct a java program for demonstrating the creation and use of user-defined exceptions.

CYCLE– VI: Multithreading: Introduction to Multithreading, Creation of Threads, Thread Life Cycle, isAlive() and join(), Thread Synchronization, and Interthread Communication.

Exercises

a) Construct a Java program that creates threads by extending Thread class. The first thread displays “Good Morning” every 1 second, the second thread displays “Hello” every 2 seconds and thethird displays “Welcome” every 3 seconds.
b) Use Runnable to develop a Java program for the above problem.
c) Construct a java program illustrating isAlive() and join().
d) Develop a Java program to solve producer consumer problem using thread synchronization.

Text Books:

1. Java: The Complete Reference; 8th edition; Herbert Schildt; TMH.
2. Programming in Java;2nd edition; SachinMalhotra, SaurabhChoudhary; Oxford University Press.
3. Core JAVA, An Integrated Approach; Dr. R. Nageswara Rao; Dreamtech Press.

Semester	**V to VII SEM**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	0	0	6	3	V23CSTOE03
Name of the Course	Database Management Systems Lab **(Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1:	Construct SQL queries to perform different database operations.	(K3)
CO2:	Experiment with various constraints and Database Indexing Techniques.	(K3)
CO3:	Construct PL/SQL Cursors and Exceptions.	(K3)
CO4:	Develop PL/SQL Functions and Procedures.	(K3)
CO5:	Develop PL/SQL Packages.	(K3)

LIST OF EXPERIMENTS

1. Construct SQL queries to facilitate acquaintance of Built-In Functions, String Functions, Numeric Functions, Date Functions and Conversion Functions.
2. Construct SQL queries using Operators.
3. Construct SQL queries to Retrieve and Change Data: Select, Insert, Delete and Update
4. Construct SQL queries using Group By, Order By and Having Clauses.
5. Construct SQL queries on Controlling data: commit, rollback and savepoint
6. Construct report using SQL*PLUS
7. Construct SQL queries for Creating, Dropping and Altering Tables, Views and Constraints
8. Construct SQL queries on Joins and Correlated Subqueries
9. Demonstrate Index, Sequence and Synonym.

PL/SQL

11. DemonstrateBasicVariables,AnchoredDeclarations,andUsageofAssignmentOperation Using PL SQL block
12. Demonstrate Bind and Substitution Variables using PL SQL block
13. Demonstrate Control Structures in PL SQL
14. Demonstrate Cursors, Exception and Composite Data Types in PL SQL.
15. Demonstrate Procedures, Functions, and Packages in PLSQL.

Textbooks:

1. OracleDatabase11gTheCompleteReferencebyOraclePress,Kevin Loney
2. DatabaseSystemsUsingOracle,NileshShah,2ndEdition, PHI.
3. Introduction to SQL, Rick FVanderLans, 4thEdition, Pearson Education.

Reference Books:

1. OraclePL/SQLInteractiveWorkbook,B.RosenzweigandE.Silvestrova,2ndEdition,Pearsoneducation.
2. SQL&PL/SQLforOracle10 g, Black Book, Dr.P.S.Deshpande, DreamTech.

Semester	V to VII SEM	L	T	P	C	COURSE CODE
Regulation	V23	3	0	0	3	V23CSTOE04
Name of the Course	Cryptography and Network Security **(Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Discuss fundamentals and mathematical support of Cryptography and Network Security. **(K2)**
CO2: Discuss symmetric and asymmetric cryptosystems. **(K2)**
CO3 : Discuss about HASH functions & Digital Signatures to provide authentication and integrity. **(K2)**
CO4: Demonstrate various methods of Mutual trust and mail security**.** **(K3)**
CO5: Review the Network& Internet Security Scenarios. **(K2)**

UNIT-I: Overview: Security attacks, Services, Mechanisms, A model for network security, Symmetric cipher model. **Classical encryption techniques**: Substitution Techniques, Transposition Techniques. **Number Theory**: Prime numbers, Fermat's theorem, Euler's Theorem, the Chinese Remainder Theorem.

UNIT-II: Block Cipher: Principles, DES, Strength of DES, AES, Block cipher Modes of Operations. **Public Key Cryptography:** Principles, Public Key Crypto system, RSA Algorithm, Diffie Hellman Key Exchange.

UNIT-III: Cryptographic Hash Functions: Application of Cryptographic Hash Functions, Requirements & Security, SHA-512, Message Authentication Functions, Requirements, HMAC. **Digital Signatures**: Properties, Attacks and Forgeries, Requirements, Digital Signature Standards, NIST Digital Signature Algorithm.

UNIT-IV: Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Asymmetric Key Distribution Using Symmetric Encryption, Distribution of Public Keys, X.509 Certificates. **User Authentication**: Remote User Authentication Principles, Kerberos. **Electronic Mail Security:** Pretty Good Privacy (PGP) And S/MIME.

UNIT-V: IP Security: Two modes, two security protocols Authentication Header, Encapsulating Security Payload. **Transport Level Security:** Secure Socket Layer (SSL) and Transport Layer Security (TLS). **HTTPS:** Connection Initiation Connection Closure.

Text Books:

1. William Stallings, "Cryptography and Network Security, Principles and Practices", Pearson Education, Sixth Edition.
2. Cryptography and Network Security, Behrouz A Forouzan, Debdeep Mukhopadhyay, (3e) Mc Graw Hill.

Reference Books:

1. Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security – Private Communication in a Public World" Pearson/PHI.

Semester	V to VII SEM	L	T	P	C	COURSE CODE
Regulation	V23	3	0	0	3	V23CSTOE05
Name of the Course	Operating Systems **(Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe Operating System Services and System Calls. **(K2)**
CO2: Illustrate Process Management Concepts and CPU Scheduling Algorithms. **(K3)**
CO3: Demonstrate Process Synchronization primitives and Process Deadlocks. **(K3)**
CO4: Illustrate Memory Management Techniques and Page Replacement Algorithms. **(K3)**
CO5: Describe File System Concepts and Mass Storage Structures. **(K2)**

UNIT-I: Introduction: Operating-System Structure, Operating-System Services, User and Operating System Interface, System Calls, Types of System Calls.

UNIT-II: Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication. **Threads:** Overview, Multithreading Models
CPU Scheduling: Basic Concepts**,** Scheduling Criteria**,** Scheduling Algorithms.

UNIT-III: Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors. **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

UNIT-IV: Memory Management: Main Memory: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.
Virtual Memory: Introduction, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

UNIT-V: Storage Management: Overview of Mass-Storage Structure, Disk Scheduling, File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Allocation Methods.

Text Book:

1. Operating System Concepts**,** Abraham Silberschatz, ,Peter Baer Galvin,Greg Gagne, 9th Edition, John Wiley and Sons Inc., 2012.

Reference Books:

1. Operating Systems – Internals and Design Principles, William Stallings, 7th Edition, Prentice Hall, 2012 .
2. Modern Operating Systems, Andrew S. Tanenbaum, Third Edition, Addison Wesley,2007.

Semester	V to VII SEM	L	T	P	C	COURSE CODE
Regulation	V23	3	0	0	3	V23CSTOE06
Name of the Course	Computer Networks **(Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Discuss fundamentals of network concepts and Reference Models. **(K2)**
CO2: Discuss Communication media and switching techniques. **(K2)**
CO3: Demonstrate Error control and Data link layer protocols. **(K3)**
CO4: Apply Routing algorithms and congestion control algorithms. **(K3)**
CO5: Discuss Transport layer protocols and Application layer protocols. **(K2)**

UNIT-I: Introduction: Reference models: The OSI Reference Model- the TCP/IP Reference Model, Examples ofNetworks: Novell Networks, Arpanet, Internet, Network Topologies WAN, LAN, MAN.

UNIT-II: Physical Layer: Transmission Media, Multiplexing: FDM, WDM and TDM- LAN Technologies, introduction to switching: Circuit Switched Networks, Datagram Networks, and Virtual Circuit Networks.

UNIT–III: Data link layer: Design issues, Framing, Flow control, error control, error detection - Parity bit, CRC, Checksum, error correction- Hamming code. MAC: ALOHA, CSMA. Elementary Data Link Layer protocols: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel. Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, HDLC, point to point protocol (PPP).Piggybacking.

UNIT-IV: Network Layer : Network layer design issues- Algorithm shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical routing, Broad cast, Multi cast Routing algorithms-Congestion control and algorithms, Internet Protocol (IP) Addresses, Subnet masking. Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

UNIT–V:Transport Layer: Services, Primitives and sockets, Elements of transport protocols, Internet Transport protocols(TCP,UDP,RPC,RTTP/RTP,RTCP) Segment headers, Primitives, Control, Congestion control.
Application layer: DNS, SMTP, POP,FTP HTTP Presentation formatting. Network security: Cryptography, DES Public key and RSA private key cryptography Algorithms.

Text Books:
1. Computer Networks — Andrew S Tanenbaum, 4th Edition. Pearson Education/PHI.
2. Data Communications and Networks – Behrouz A. Forouzan.Third Edition TMH.

Reference Books:
1. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson.

Semester	**V to VII SEM**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTOE07
Name of the Course	Data Mining (**Open Elective**)					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the concept of Data Mining and its functionalities. **(K2)**
CO2: Discuss various Data Preprocessing Techniques. **(K3)**
CO3: Demonstrate Association Analysis Techniques. **(K3)**
CO4: Illustrate various Classification Techniques. **(K3)**
CO5: Use different Clustering techniques to cluster data. **(K3)**

UNIT-I: Introduction: Need for Data Mining, Knowledge Discovery from Data, Kinds of Data mined, Kinds of Patterns mined, Technologies used, Kinds of Applications targeted, Major Issues in Data Mining, Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Measuring Data Similarity and Dissimilarity.

UNIT-II: Data Preprocessing: Overview of Data Preprocessing, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

UNIT-III: Mining Frequent Patterns, Associations, and Correlations: Basic Concepts, Frequent Itemset Mining Methods- Apriori Algorithm: Finding Frequent Itemsets by Confined Candidate Generation, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, Pattern-Growth Approach for Mining Frequent Itemsets.

UNIT-IV: Classification: Basic Concepts, Decision Tree Induction, Attribute Selection Measures, Tree Pruning. **Bayes Classification Methods:** Bayes' Theorem, Naive Bayesian Classification.
Bayesian Belief Networks: Concepts and Mechanisms.

UNIT-V: Cluster Analysis: Basic Concepts and Methods, Partitioning Methods, Hierarchical Methods, Density Based Method-DBSCAN.

Text Books:

1. Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei,3rd Edition, Morgan Kaufmann Publishers.

.

Reference Books:

1. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, 1st Edition, Pearson Education Inc.
2. Data Mining and Analysis,Mohammed J Zaki, Wagner Meira JR, 1st Edition ,Cambridge University Press.

Semester	**V to VII SEM**	**L**	**T**	**P**	**C**	**COURSE CODE**
Regulation	V23	3	0	0	3	V23CSTOE08
Name of the Course	Software Engineering **(Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Demonstrate the Software Development life cycle Models. **(K3)**
CO2: Illustrate the Requirements engineering process and SRS document. **(K3)**
CO3: Develop the Software Architecture and Design Modeling. **(K3)**
CO4: Apply the Coding & Testing techniques and Risk management strategies. **(K3)**
CO5: Describe Project estimation techniques and Quality Management& Metrics. **(K2)**

UNIT-I: Software and Software Engineering: The Nature of Software, Software Engineering, Software Process, Software Engineering Practice, Software Myths. **Software process models:** Waterfall model, Prototyping, Iterative development, Unified process, RAD model, Spiral model, and agile process.

UNIT-II: Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, SRS document. **Requirements engineering process:** Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

UNIT-III: Software Architecture: Role of software architecture, Architecture views, components and connector view, Cohesion and Coupling, documenting architecture design. **Design:** Design concepts, Function-oriented design, object-oriented design, UML diagrams, and Data flow diagram.

UNIT-IV: Coding and Testing: Programming principles and guidelines, incrementally developing code. Testing concepts, testing process, Black-box & White-box testing.**Risk management**: Reactive vs. Proactive Risk strategies, Software risks, Risk identification, Risk projection, Risk refinement, RMMM Plan.

UNIT-V: Software Project Estimation& Maintenance: Decomposition techniques, Empirical Estimation Models, Maintenance Process, Reengineering, Configuration Management. Metrics for Products & Quality Management: Software Measurement, Metrics for software quality, Quality concepts, Software Reviews, Formal technical reviews, SEI-CMM Model, Six Sigma and ISO 9000 quality standards.

Text Books:

1. Software Engineering, A practitioner's Approach- Roger S.Pressman, 7th Edition, McGrawHill International Edition.
2. Software Engineering- Ian Sommerville, 9th Edition, Pearson education.Software Engineering, A Precise approach, PankajJalote, Wiley

Reference Books:

1. CMMI and Six Sigma: Partners in Process Improvement, Jeannine M. Siviy,M. Lynn Penn, Robert W. Stoddard, 1st edition, Addison Wesley;
2. Software Engineering principles and practice,WSJawadekar, 3rdEdition,TMH.

Semester	V to VII SEM	L	T	P	C	COURSE CODE
Regulation	V23	3	0	0	3	V23CSTOE09
Name of the Course	Cyber Security (**Open Elective**)					
Branch	Common to CIVIL,MECH,EEE,ECE and ECT					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Explain the fundamental concepts of cybercrime, its origins, types, and the challenges posed by modern devices and technologies. **(K2)**

CO2: Demonstrate various tools and techniques used in committing cybercrimes, including phishing, malware, spoofing, session hijacking, and social engineering. **(K2)**

CO3: Demonstrate knowledge of cybercrime investigation processes including digital evidence collection, eDiscovery, email/IP tracking, and the use of encryption/decryption tools. **(K2)**

CO4: Apply computer forensic techniques and use current tools to investigate and analyze digital evidence on different platforms such as Windows, Linux, and mobile devices. **(K3)**

CO5: Demonstrate the legal framework related to cybercrime, with a focus on the Indian IT Act, its challenges, amendments, and the global legal scenario. **(K2)**

UNIT I: Introduction to Cybercrime: Introduction, Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Cybercriminals, Classifications of Cybercrime, Cyberstalking, Cybercafe and Cybercrimes, Botnets. Attack Vector, Proliferation of Mobile and Wireless Devices, Security Challenges Posed by Mobile Devices, Attacks on Mobile/Cell Phones.

UNIT II: Tools and Methods: Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, Sniffers, Spoofing, Session Hijacking Buffer over flow, DoS and DDoS Attacks, Attacks on Wireless Networks, Identity Theft (ID Theft), Foot Printing and Social Engineering, Port Scanning.

UNIT III: Cyber Crime Investigation: Introduction, Investigation Tools, eDiscovery, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Hands on Case Studies. Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

UNIT IV: Computer Forensics and Investigations: Understanding Computer Forensics, Preparing for Computer Investigations. Current Computer Forensics Tools: Evaluating Computer Forensics Tools, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Graphics and Network Forensics, E-mail Investigations, Cell Phone and Mobile Device Forensics.

UNIT V: Cyber Crime Legal Perspectives: Introduction, Cybercrime and the Legal Landscape around the World, The Indian IT Act, Challenges to Indian Law and Cybercrime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Amendments to the Indian IT Act, Cybercrime and Punishment, Cyber crime and offences, Organisations dealing with Cyber crime and Cyber security in India.

Textbooks

1. Sunit Belapure Nina Godbole "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", WILEY, 2011.
2. Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi, 2009.

Reference Books:

1. Michael T. Simpson, Kent Backman and James E. Corley, "Hands on Ethical Hacking and Network Defence", Cengage, 2019.
2. Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi, 2009.
3. Alfred Basta, Nadine Basta,Mary Brown and Ravinder Kumar "Cyber Security and Cyber Laws" , Cengage,2018.

Semester	V to VII SEM	L	T	P	C	COURSE CODE
Regulation	V23	3	0	0	3	V23CSTOE10
Name of the Course	Human Computer Interaction **(Open Elective)**					
Branch	Common to CIVIL,MECH,EEE,ECE,ECT,CAI , AIM and CSDS					

Syllabus Details

Course Outcomes: After Successful completion of the Course, the student will be able to:

CO1: Describe the principles and characteristics of GUI. **(K2)**

CO2:Describe how a computer system may be modified to include human diversity. **(K2)**

CO3: Select an effective style and screen design for a specific business application. **(K2)**

CO4: Discuss System Menus & Navigation Schemes**.** **(K2)**

CO5: Select Device and Screen based controls. **(K2)**

UNIT I:The User Interface: Introduction, Importance of the User Interface, Importance and benefits of Good Design, Characteristics of Graphical and Web User Interface Graphical User Interface, popularity of graphics, concepts of Direct Manipulation, Graphical System advantage and disadvantage, Characteristics of GUI, Characteristics of Web Interface, Principles of User Interface Design.

UNIT II:TheUserInterfaceDesignProcess:ObstaclesandPitfallinthedevelopmentProcess,Usability, The Design Team, Human Interaction with Computers, Important Human Characteristics in Design, Human Consideration in Design, Human Interaction Speeds, Performance versus Preference, Methods for Gaining and Understanding of Users.

UNIT III: Understanding Business Functions: Business Definitions& Requirement analysis, Determining Business Functions. **Principles of Good Screen Design:** Human considerations in screen Design, interface design goals, screen meaning and purpose, Technological considerations in Interface Design.

UNIT IV: System Menus and Navigation Schemes: Structure, Functions, Context, Formatting, PhrasingandSelecting,NavigatingofMenus,KindsofGraphicalMenusWindowsInterface:Windowschar acteristic,ComponentsofWindow,WindowsPresentationStyles,TypesofWindows,WindowManageme nt,

UNITV: Device and Screen-Based Control: Device based controls, Operable Controls,Textentry/read-Only Controls, Section Controls, Combining Entry/Selection Controls Presentation Controls, Selecting proper controls.

Text Books:

1. "The Essential Guide to User Interface Design", Wilbert O. Galitz, 2nd edition,2002,Wiley India Edition.
2. Prece, Rogers, "Sharps Interaction Design", Wiley India.
3. "Designing the user interfaces". Ben Shneidermann 3rd Edition, Pearson Education Asia.

Reference Books:

1. "User Interface Design" , SorenLauesen, Pearson Education
2. "Essentials of Interaction Design", Alan Cooper, Robert Riemann, David Cronin, Wiley
3. "Human Computer Interaction", Alan Dix, Janet Fincay, GreGoryd, Abowd, Russell, Bealg, Pearson Education.

Annexure-IV
List of Courses for B.Tech(Hons) in Computer Science (SWAYAM)

S.No.	Course Name	No. of Weeks	Credits
1*	Foundation of Data Science	8	2
	Data Science for Engineers		
	Data Science and Big Data	12	3
2	Introduction to Haskell Programming	8	2
3	Introduction to Information Technology	12	3
4	Cloud Computing and Distributed Systems	8	2
5	Foundation of Cloud IoT Edge ML	8	2
6	AI: Constraint Satisfaction	8	2
7	Artificial Intelligence using Prolog Programming	12	3
	Artificial Intelligence: Search Methods for Problem Solving	12	3
8	Applied Accelerated Artificial Intelligence	12	3
9	Responsible & Safe AI Systems	12	3
10	Machine Learning Using Python Programming	12	3
11	Machine Learning For Soil And Crop Management	12	3
12	Essentials Of Data Science With R Software - 1: Probability And Statistical Inference	12	3
13	Essentials Of Data Science With R Software - 2: Sampling Theory And Linear Regression Analysis	12	3
14	Natural Language Processing	12	3
15	**Deep Learning for Natural Language Processing**	12	3
16	**Reinforcement Learning**	12	3
17*	**Computer Vision And Image Processing - Fundamentals And Applications**	12	3
	Deep Learning for Computer Vision		
18*	Modern Computer Vision	12	3
	Deep Learning For Visual Computing		
19	Introduction to Cyber Security	12	3
20	**Privacy and Security in Online Social Media**	12	3
21	Cyber Security, Tools, Techniques and Counter Measures	12	3
22	Blockchain and its Applications	12	3
23	Introduction To Internet Of Things	12	3
24	Introduction To Industry 4.0 And Industrial Internet Of Things	12	3
25	**Social Networks**	12	3
26	Information Security and Cyber Forensics	12	3
27	Digital Forensics	12	3
28*	Web based Technologies and Multimedia Applications	12	3
	Web-designing and multimedia Technology	12	3
29	Advanced Computer Networks	12	3

*** The Student Can only Opt for one course of the corresponding cluster.**

Note:

1. The List may be updated every semester.
2. A student can do a **maximum of 2 courses** per semester.
3. A student should obtain **18 credits** by the **end of the VII semester** and submit the proofs at the time of **VIII sem exam registration**.

List of Courses for B.Tech(Minors) in Computer Science

S.No.	Course Name	No. of Weeks	Credits
1*	**Programming, Data Structures And Algorithms Using Python**	8	2
	Introduction to Data Structures	8	2
	Data Structures	12	3
2	Programming In Java	12	3
3	Object Oriented System Development Using UML, Java And Patterns	12	3
4*	Programming in Python	12	3
	The Joy of Computing using Python		
5	**Operating systems**	8	2
6	**Introduction to Database Systems**	12	3
7	**Computer Networks And Internet Protocol**	12	3
8	Cloud Computing	12	3
9	Introduction to Cyber Security	12	3
10*	Cryptography and Network Security	12	3
	Network Security	12	3
	Information Security	16	4
11	Introduction to Machine Learning	12	3
12	Machine Learning For Soil And Crop Management	12	3
13	Deep Learning (IIT Kharagpur)	12	3
14	**Artificial Intelligence: Knowledge Representation And Reasoning**	12	3
15	Artificial Intelligence using Prolog Programming	12	3
16	Introduction To Internet Of Things	12	3
17	Foundation of Cloud IoT Edge ML	12	3

*** The Student Can only Opt for one course of the corresponding cluster.**

Note:

1. The List may be updated every semester.
2. A student can do a **maximum of 2 courses** per semester.
3. A student should obtain **18 credits** by the **end of the VII semester** and submit the proofs at the time of **VIII sem exam registration**.
4. A student must not study the same course in their curriculum and minors.

Annexure-V

M.Tech(CS) Programme Course Structure

(With effect from **2025-26** Admitted Batch onwards)

SEMESTER-I

S.No.	Course Code	Course Title	L	T	P	C
1.	V25CTT01	Data Structures and Algorithm Analysis	3	1	0	4
2.	V25CTT02	Artificial Intelligence	3	1	0	4
3.	V25CTT03	Mathematical Foundations of Computer Science	3	1	0	4
4.	V25CTT04 - V25CTT11	Program Elective –I	3	0	0	3
5.	V25CTT04 - V25CTT11	Program Elective – II	3	0	0	3
6.	V25CTL01	Data Structures and Algorithms Analysis Lab	0	1	2	2
7.	V25CTL02	Artificial Intelligence Lab	0	1	2	2
8.	V25CTL05	Seminar-I	0	0	2	1
Total Credits			**15**	**5**	**6**	**23**

SEMESTER-II

S.No.	Course Code	Course Title	L	T	P	C
1.	V25CTT12	Machine Learning	3	1	0	4
2.	V25CTT13	OS and UNIX Programming	3	1	0	4
3.	V25CTT14	Cloud Computing	3	1	0	4
4.	V25CTT15 - V25CTT22	Program Elective – III	3	0	0	3
5.	V25CTT15 - V25CTT22	Program Elective-IV	3	0	0	3
6.	V25CTL03	Machine Learning Lab	0	1	2	2
7.	V25CTL04	OS and Unix lab	0	1	2	2
8.	V25CTL06	Seminar–II	0	0	2	1
Total Credits			**15**	**5**	**6**	**23**

SEMESTER-III

S.No.	Course Code	Course Title	L	T	P	C
1.	V25MOOCS1	Research Methodology and IPR/ *Swayam 12 week MOOC course – RM&IPR*	3	0	0	3
2.	V25CTP01	Summer Internship/Industrial Training(8-10weeks)*	-	-	-	3
3.	V25CTL07	Comprehensive Viva[#]	-	-	-	2
4.	V25CTP02	Dissertation Part–A[$]	-	-	20	10
Total Credits			**3**	**-**	**20**	**18**

**Student attended during summer/ year break and assessment will be done in 3rd Sem.*

 # Comprehensive viva can be conducted courses completed up to second sem.

$ Dissertation–Part A, internal assessment

SEMESTER-IV

S.No.	Course Code	Course Title	L	T	P	C
1.	V25CTP03	Dissertation Part–B[%]	-	-	32	16
Total Credits			**3**	**-**	**-**	**-**

%External Assessment

List of Program Elective Courses in I Semester(Electives I & II)

S.No.	Course Code	Course Title
1	V25CTT04	Full Stack Technologies
2	V25CTT05	Computer Networks
3	V25CTT06	Social Network Analysis
4	V25CTT07	Automata Theory and Compiler Design
5	V25CTT08	Object Oriented Software Engineering
6	V25CTT09	Data Warehousing and Data Mining
7	V25CTT10	Advanced Computer Architecture
8	V25CTT11	Artificial Neural Networks

List of Program Elective Courses in II Semester(Electives III & IV)

S.No.	Course Code	Course Title
1	V25CTT15	Data Science
2	V25CTT16	Quantum Computing
3	V25CTT17	Object Oriented and Analysis Design
4	V25CTT18	Cryptography and Network Security
5	V25CTT19	Secure coding
6	V25CTT20	DevOps
7	V25CTT21	Web Application Security
8	V25CTT22	Big Data Analytics

I SEM	**Data Structures and Algorithm Analysis**	Course Code: V25CTT01	**L**	**T**	**P**	**C**
			3	1	0	4

Course Outcomes: After completion of course, students would be able to

CO1: Explain basic data structures like linked lists, stacks, queues, and trees. **(K2)**
CO2: Describe searching and sorting methods along with time and space complexity. **(K2)**
CO3: Explain how trees and graphs are represented and how they can be traversed. **(K2)**
CO4: Describe abstract data types like stacks, queues, and hash tables, and explain how collisions are handled. **(K2)**
CO5: Apply algorithms to implement and perform operations on data structures such as linked lists, heaps, and search trees. **(K3)**

UNIT–I: Introduction to Data Structures - Singly Linked Lists, Doubly Linked Lists, Circular Lists - Algorithms, Stacks and Queues - Algorithm Implementation using Linked Lists, time and space complexity, Asymptotic notations.

UNIT–II: Searching - Linear and Binary Search Methods, Sorting - Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Trees - Binary trees, Operations - Insertion, Deletion, Properties, Representation and Traversals (DFT, BFT), Expression Trees (Infix, prefix, postfix), Graphs - Basic Concepts, Storage structures and Traversals.

UNIT–III: Dictionaries, ADT, The List ADT, Stack ADT, Queue ADT, Hash Table Representation, Hash Functions, Collision Resolution - Separate Chaining, Open Addressing - Linear Probing, Double Hashing.

UNIT–IV: Priority Queues - Definition, ADT, Realising a Priority Queue Using Heaps, Definition, Insertion, Deletion, Search Trees - Binary Search Trees, Definition, ADT, Implementation, Operations - Searching, Insertion, Deletion.

UNIT–V: Search Trees - AVL Trees, Definition, Height of AVL Tree, Operations - Insertion, Deletion and Searching. Introduction to Red-Black and Splay Trees, B-Trees, Height of B-Tree, Insertion, Deletion and Searching, Comparison of Search Trees .

Text Books:

1. Data Structures: A Pseudo code Approach with C,2nd Edition, Richard F. Gilberg, Behrouz A. Forouzon, Cengage Learning, 2004
2. Data Structures, Algorithms and Applications in java, 2nd Edition,SartajSahni, University Press/Orient Black Swan, 2005

Reference Books:

1. Data Structures and Algorithm Analysis,2nd Edition, Mark Allen Weiss,Pearson,2002
2. Data Structures and Algorithms in C++,3rd Edition, Adam Drozdek, Cengage Learning, 2005
3. C and Data Structures: A Snap Shot Oriented Treatise Using Live Engineering Examples, 1st Edition, N.B.Venkateswarulu, E.V. Prasad, S Chand &Co,2009
4. Classic Data Structures,2nd Edition, Debasis Samantha, PHI Learning,2009

I SEM	**Artificial Intelligence**	Course Code: V25CTT02	**L**	**T**	**P**	**C**
			3	1	0	4

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Describe the basics, history, and applications of Artificial Intelligence. **(K2)**
CO2: Apply game-playing methods and logic concepts to solve problems. **(K3)**
CO3: Use different knowledge representation techniques in AI. **(K3)**
CO4: Apply probability-based methods to handle uncertainty. **(K3)**
CO5: Explain and use fuzzy logic for reasoning with uncertainty. **(K3)**

UNIT-I: Introduction to Artificial Intelligence: Introduction, history, intelligent systems, foundations of AI, applications, tic-tac-toe game playing, development of AI languages, current trends in AI.
Problem Solving – State-Space Search and Control Strategies: Introduction, general problem solving, characteristics of problems, exhaustive searches, heuristic search techniques, iterative-deepening A*, constraint satisfaction.

UNIT-II: Problem Reduction and Game Playing: Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games.
Logic Concepts: Introduction, propositional calculus, propositional logic, natural deduction system, axiomatic system, semantic tableau system in propositional logic, resolution refutation in propositional logic, predicate logic.

UNIT-III: Knowledge Representation: Introduction, approaches to knowledge representation, knowledge representation using semantic networks, extended semantic networks for KR, knowledge representation using frames.
Advanced Knowledge Representation Techniques: Introduction, conceptual dependency theory, script structure, CYC theory, case grammars, semantic web.

UNIT-IV: Uncertainty Measure – Probability Theory: Introduction, probability theory, Bayesian belief networks, certainty factor theory, Dempster-Shafer theory, non-monotonic reasoning, TMS (Truth Maintenance System).

UNIT-V: Fuzzy Sets and Fuzzy Logic: Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi-valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.

Text Books:
1. Artificial intelligence, A modern Approach, 2nd ed, Stuart Russel, Peter Norvig, Prentice Hall
2. Artificial Intelligence, Saroj Kaushik, 1st Edition, CENGAGE Learning, 2011.

Reference Books:
1. Artificial intelligence, structures and Strategies for Complex problem solving, 5th Edition, George F Lugar, PEA
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang, Springer, 2017
3. Artificial Intelligence, A new Synthesis, 1st Edition, Nils J Nilsson, Elsevier, 1998
4. Artificial Intelligence- 3rd Edition, Rich, Kevin Knight, Shiv Shankar B Nair, TMH
5. Introduction To Artificial Intelligence And Expert Systems, 1st Edition, Patterson, Pearson India, 2015

I SEM	**Mathematical Foundations for Computer Science**	Course Code: V25CTT03	**L**	**T**	**P**	**C**
			3	1	0	4

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain probability concepts, random variables, and distribution functions in discrete and continuous cases.**(K2)**

CO2: Describe sampling methods, estimation techniques, and computation of statistical measures for grouped data. **(K2)**

CO3: Illustrate hypothesis testing procedures, error types, significance levels, and quality control methods in statistical analysis. **(K2)**

CO4: Apply algebraic properties, number theory algorithms, and modular arithmetic concepts to solve mathematical problems. **(K3)**

CO5: Construct graphs using matrix representations, paths, circuits, spanning trees, and graph colouring techniques. **(K3)**

UNIT–I: Basic Probability and Random Variables: Random Experiments, Sample Spaces, Events, the Concept of Probability, the Axioms of Probability, Some Important Theorems on Probability, Assignment of Probabilities, Conditional Probability, Theorems on Conditional Probability, Independent Events, Bayes Theorem or Rule. Random Variables, Discrete Probability Distributions, Distribution Functions for Random Variables, Distribution Functions for Discrete Random Variables, Continuous Random Variables.

UNIT–II: Sampling and Estimation Theory: Population and Sample, Statistical Inference, Sampling With and Without Replacement, Random Samples, Random Numbers, Population Parameters, Sample Statistics, Sampling Distributions, Frequency Distributions, Relative Frequency Distributions, Computation of Mean, Variance, and Moments for Grouped Data, Unbiased Estimates and Efficient Estimates, Point Estimates and Interval Estimates, Reliability, Confidence Interval Estimates of Population Parameters, Maximum Likelihood Estimates.

UNIT–III: Tests of Hypothesis and Significance: Statistical Decisions, Statistical Hypotheses, Null Hypotheses, Tests of Hypotheses and Significance, Type I and Type II Errors, Level of Significance, Tests Involving the Normal Distribution, One-Tailed and Two-Tailed Tests, P-Value, Special Tests of Significance for Large Samples, Special Tests of Significance for Small Samples, Relationship between Estimation Theory and Hypothesis Testing, Operating Characteristic Curves, Power of a Test, Quality Control Charts, Fitting Theoretical Distributions to Sample Frequency Distributions, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates' Correction for Continuity, Coefficient of Contingency.

UNIT–IV: Algebraic Structures and Number Theory: Algebraic Systems, Examples, General Properties, Semi Groups and Monoids, Homomorphism of Semi Groups and Monoids, Group, Subgroup, Abelian Group, Homomorphism, Isomorphism. Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat's Theorem and Euler's Theorem).

UNIT–V: Graph Theory: Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multi graphs, Planar Graphs, Euler's Formula, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Algorithms for Spanning Trees (Problems Only and Theorems without Proofs).

Text Books:

1. Foundation Mathematics for Computer Science,1st edition, John Vince,Springer,2015
2. Probability&Statistics,3rd Edition, Murray R. Spiegel, John J. Schiller and R. Alu Srinivasan, Schaum's Outline Series, Tata McGraw-Hill Publishers, 2018
3. Probability and Statistics with Reliability,2nd edition,K.Trivedi,Wiley,2011
4. Discrete Mathematics and its Applications with Combinatorics and GraphTheory,7th Edition, H. Rosen, Tata McGraw Hill, 2003

Reference Books:

1. Probability and Computing: Randomized Algorithms and Probabilistic Analysis,1st edition, M. Mitzenmacher and E.Upfal, 2005
2. Applied Combinatorics,6th edition,AlanTucker,Wiley,2012

I SEM	**Full Stack Technologies (Program Elective-I & Program Elective-II)**	Course Code: V25CTT04	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain the concepts of internet protocols, web page structure, XML, and document models for web development. **(K2)**

CO2: Describe the basics of JavaScript, AngularJS expressions, and form validation techniques for client-side programming. **(K2)**

CO3: Apply PHP and Node.js concepts for server-side scripting and database-driven applications. **(K3)**

CO4: Use jQuery and MySQL queries to implement interactive web applications with database connectivity. **(K3)**

CO5: Develop and deploy applications using MongoDB, cloud platforms, and web services (SOAP, WSDL, REST). **(K3)**

UNIT-I: Introduction to Web- Internet and World Wide Web, Domain Name Service, Protocols: HTTP, FTP, SMTP, HTML5 concepts, CSS3, Anatomy of a Web Page, X**ML**- Document Type Definition, XML Schemas, Document Object Model, XSLT, DOM and SAX Approaches.

UNIT-II: JavaScript- The Basics of JavaScript: Objects, Primitives, Operations and Expressions, Control Statements, Arrays, Functions, Constructors, Pattern Matching using Regular Expressions.
Angular JavaScript – AngularJS Expressions: Array, Objects, $eval, Strings, AngularJS Form Validation & Form Submission.

UNIT-III: PHP Programming: Back-end Scripts PHP, Node.js, Working with PHP – Using Variables, Using Constants, Data Types, Operators, Conditional & Control Statements, Arrays, Functions. Working with Forms and Databases such as MySQL. **Node.js** – Introduction, Advantages, Node.js Process Model, Node.js Modules.

UNIT-IV: jQuery: Introduction to jQuery, Syntax, Selectors & Events.
MySQL: Practice MySQL Queries, Aggregate Functions, Regular Expressions, Joins & Unions, Sub-Queries, Database Connectivity with MySQL.

UNIT-V: MongoDB – Introduction, Architecture, Features, Examples, Database Creation & Collection in MongoDB. **Deploying Applications** – Web Hosting & Domains, Deployment Using Cloud Platforms. **Web Services** – SOAP, WSDL and RESTful Architecture.

Text Books:

1. Programming the World Wide Web, 7th Edition, Robert W. Sebesta, Pearson, 2013
2. Web Technologies, 1st Edition, 7th Impression, Uttam K. Roy, Oxford, 2012
3. Pro Mean Stack Development, 1st Edition, Elad Elrom, Apress O'Reilly, 2016
4. JavaScript & jQuery: The Missing Manual, 2nd Edition, David Sawyer McFarland, O'Reilly, 2011
5. Web Hosting for Dummies, 1st Edition, Peter Pollock, John Wiley & Sons, 2013
6. RESTful Web Services, 1st Edition, Leonard Richardson, Ruby, O'Reilly, 2007

Reference Books:

1. Ruby on Rails Up and Running: Lightning-fast Web Development, 1st Edition, Bruce Tate, Curt Hibbs, O'Reilly, 2006
2. Programming Perl, 4th Edition, Tom Christiansen, Jonathan Orwant, O'Reilly, 2012
3. Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML and AJAX Black Book, 1st Edition, Dream Tech, 2009
4. An Introduction to Web Design, Programming, 1st Edition, Paul S. Wang, Sanda S. Katila, Cengage Learning, 2003

Web Reference Links

1. http://www.upriss.org.uk/perl/PerlCourse.html

I SEM	**Computer Networks (Program Elective-I & Program Elective-II)**	Course Code: V25CTT05	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain network layer functions, routing algorithms, and congestion control techniques.**(K2)**

CO2: Describe IP protocols, addressing schemes, and internet control protocols. **(K2)**

CO3: Illustrate transport protocols like UDP, TCP, and SCTP, including their services and control mechanisms. **(K3)**

CO4: Apply wireless network technologies such as WLAN, Bluetooth, WiMAX, and cellular telephony in practical scenarios. **(K3)**

CO5: Use mobile computing concepts, sensor networks, mesh networks, and P2P protocols to design secure and efficient communication systems. **(K3)**

UNIT-I:Network layer: Network Layer design issues: store-and-forward packet switching, services provided transport layers, implementation of connectionless services, implementation of connection-oriented services, comparison of virtual-circuit and datagram subnets, Routing Algorithms - shortest path routing, flooding, distance vector routing, link state routing, hierarchical routing, congestion control algorithms - approaches to congestion control, traffic-aware routing, admission control, traffic throttling, choke packets, load shedding, random early detection, Quality of Service, application requirements, traffic shaping, leaky and token buckets.

UNIT-II: Internetworking and IP protocols: How networks differ, how networks can be connected, internetworking, tunneling, the network layer in the internet, IPV4 protocol, IP addresses, subnets, CIDR, classful and special addressing, network address translation (NAT), IPV6 address structure, address space, IPV6 advantages, packet format, extension headers, transition from IPV4 to IPV6, internet control protocols - ICMP, ARP, DHCP.

UNIT-III: Transport Layer Protocols: Introduction, services, port numbers, user datagram protocol (UDP): user datagram, UDP services, UDP applications; transmission control protocol (TCP): TCP services, TCP features, segment, a TCP connection, state transition diagram, windows in TCP, flow control and error control, TCP congestion control, TCP timers; SCTP: SCTP services, SCTP features, packet format, an SCTP association, flow control, error control.

UNIT-IV: Wireless LANs: Introduction, architectural comparison, access control; the IEEE 802.11 project: architecture, MAC sub-layer, addressing mechanism, physical layer; Bluetooth: architecture, Bluetooth layers; other wireless networks: WiMAX - services, IEEE project 802.16, layers in project 802.16; cellular telephony: operations, first generation (1G), second generation (2G), third generation (3G), fourth generation (4G); satellite networks: operation, GEO satellites, MEO satellites, LEO satellites.

UNIT-V:Emerging trends in computer networks: Mobile computing - motivation for mobile computing, protocol stack issues in mobile computing environment, mobility issues, security issues in mobile networks; MOBILE Ad Hoc Networks - applications, challenges, and issues in MANETs, MAC layer issues, routing protocols, transport layer issues, ad hoc network security; Wireless Sensor Networks - functioning, operating system support, characteristics, operation, sensor architecture, cluster management; Wireless Mesh Networks - design, issues; Computational grids - features, construction design, grid design features; P2P networks - characteristics, classification, Gnutella, BitTorrent; Session Initiation Protocol (SIP) - characteristics, addressing, components, establishment, security.

Text Books:

1. Data Communications and Networking 4th edition Behrouz A Fourzan,TMH-2007
2. Computer Networks 4th edition Andrew S Tanenbaum,Pearson,2012
3. Computer Networks, Mayank Dave, CENGAGE, First edition.2012

Reference Books:

Computer Networks, A System Approach,5thed,LarryL Peterson and Bruce S Davie, Elsevier-2012

I SEM	**Social Network Analysis (Program Elective-I & Program Elective-II)**	Course Code: V25CTT06	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain key concepts in social network analysis including centrality, balance, and homophily.**(K2)**
CO2: Describe random graph models, network growth patterns, and cohesive subgroups. **(K2)**
CO3: Illustrate network topology, diffusion processes, and navigation strategies. **(K3)**
CO4: Apply small world models and clustering techniques to analyze real-world networks. **(K3)**
CO5: Use network algorithms, game theory concepts, and agent-based models to explore network structure and dynamics. **(K3)**

UNIT-I: Social Network Analysis: Preliminaries and definitions, Erdos Number Project, centrality measures, balance and homophily.

UNIT-II: Random Graph Models: Random graphs and alternative models, models of network growth, navigation in social networks, cohesive subgroups, multidimensional scaling, structural equivalence, roles and positions.

UNIT-III: Network Topology and Diffusion: Contagion in networks, complex contagion, percolation and information, navigation in networks revisited.

UNIT-IV: Small World Models: Small world experiments, small world models, origins of small world, heavy tails, small diameter, clustering of connectivity, the Erdos–Renyi model, clustering models.

UNIT-V: Network Structure and Dynamics: Important vertices and page rank algorithm, towards rational dynamics in networks, basics of game theory, coloring and consensus, biased voting, network formation games, network structure and equilibrium, behavioral experiments, spatial and agent-based models.

Text Books:

1. S. Wasserman and K. Faust. Social Network Analysis: Methods and Applications(Cambridge, Cambridge University Press, 1994)
2. D. Easley and J. Kleinberg, Networks, Crowds and Markets: Reasoning about a highly connected world-2010

Reference Books:

1. Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences) by Stanley Wasserman, Katherine Faust, 1994.

I SEM	**Automata Theory and Compiler Design (Program Elective-I & Program Elective-II)**	Course Code: V25CTT07	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Interpret the structure and behavior of finite automata through alphabets, states, and transitions. **(K2)**

CO2: Analyze regular expressions and grammars to identify patterns and ambiguity in language processing. **(K2)**

CO3: Examine the computational power of pushdown and Turing machines, and their implications for decidability. **(K2)**

CO4: Construct lexical and syntax analyzers by integrating parsing algorithms and grammar rules. **(K3)**

CO5: Develop intermediate code and memory management strategies to enhance compiler performance. **(K3)**

UNIT- I:Introduction to Finite Automata: Structural representations, automata and complexity, the central concepts of automata theory – alphabets, strings, languages, problems.
Nondeterministic Finite Automata: Formal definition, an application, text search, finite automata with epsilon-transitions. Deterministic Finite Automata: Definition of DFA, how a DFA processes strings, the language of DFA, conversion of NFA with epsilon-transitions to NFA without epsilon-transitions, conversion of NFA to DFA.

UNIT- II: Regular Expressions: Finite automata and regular expressions, applications of regular expressions, algebraic laws for regular expressions, conversion of finite automata to regular expressions. Pumping Lemma for Regular Languages: Statement of the pumping lemma, applications of the pumping lemma. Context-Free Grammars: Definition of context-free grammars, derivations using a grammar, leftmost and rightmost derivations, the language of a grammar, parse trees, ambiguity in grammars and languages.

UNIT- III: Push Down Automata: Definition of the pushdown automaton, the languages of a PDA, equivalence of PDAs and CFGs, acceptance by final state.
Turing Machines: Introduction to Turing machine, formal description, instantaneous description, the language of a Turing machine. Undecidability: Undecidability, a language that is not recursively enumerable, an undecidable problem that is RE, undecidable problems about Turing machines.

UNIT- IV: Introduction to Compiler Design: The structure of a compiler, lexical analysis – the role of the lexical analyzer, input buffering, recognition of tokens, the lexical-analyzer generator Lex. Syntax Analysis: Context-free grammars, writing a grammar, top-down parsing, bottom-up parsing, introduction to LR parsing – simple LR, more powerful LR parsers.

UNIT- V:Syntax-Directed Translation: Syntax-directed definitions, evaluation orders for SDDs, syntax-directed translation schemes, implementing L-attributed SDDs. Intermediate-Code Generation: Variants of syntax trees, three-address code. Run-Time Environments: Stack allocation of space, access to nonlocal data on the stack, heap management.

Text Books:

1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
3. Theory of Computer Science–Automata languages and computation, Mishra and Chandra shekaran, 2nd Edition, PHI.

Reference Books:

1. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
2. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
3. Lex & yacc–John R. Levine, Tony Mason, Doug Brown, O' reilly
4. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology

I SEM	**Object Oriented Software Engineering (Program Elective-I & Program Elective-II)**	Course Code: V25CTT08	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain OOP concepts and UML basics. **(K2)**
CO2: Apply OOAD principles using UML diagrams. **(K3)**
CO3: Demonstrate test-driven development and software testing. **(K3)**
CO4: Describe software maintenance and refactoring techniques. **(K2)**
CO5: Apply advanced OOP concepts in software projects. **(K3)**

UNIT-I: Introduction to Object-Oriented Programming: Overview of software engineering, introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, introduction to software development process and software development lifecycle (SDLC).

UNIT-II: Requirements Analysis and Design: Requirements analysis and specification, use cases and scenarios, Object-Oriented Analysis and Design (OOAD), design patterns, UML modelling techniques (class diagrams, sequence diagrams, state machine diagrams, activity diagrams).

UNIT-III: Software Construction and Testing: Software construction basics, Object-Oriented design principles, Object-Oriented programming languages (Java, C++, Python), software testing basics (unit testing, integration testing, system testing), test-driven development (TDD).

UNIT-IV: Software Maintenance and Evolution: Software maintenance basics, refactoring techniques, software version control, code review and inspection, software evolution and re-engineering.

UNIT-V: Advanced Topics in Object-Oriented Software Engineering: Model-driven engineering (MDE), aspect-oriented programming (AOP), component-based software engineering (CBSE), service-oriented architecture (SOA), agile software development and Scrum methodologies.

Text Books:

1. Craig Larman, An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd Edition, Prentice-Hall.
2. Sachin Malhotra, Programming in Java, Oxford University Press.

Reference Books:

1. G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, Wiley, 1998.
2. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns.
3. J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison Wesley.

Online Learning Resources:

1. https://nptel.ac.in/courses/106105153 - NPTEL – Object-Oriented Analysis and Design
2. https://nptel.ac.in/courses/106105237 - NPTEL – Object-Oriented System Development
3. https://spoken-tutorial.org/course/ - IIT Bombay Spoken Tutorial – Java Programming
4. https://vlab.amrita.edu/?sub=3&brch=273 - Virtual Labs (Vlabs) – Object-Oriented Programming

I SEM	**Data Warehousing and Data Mining (Program Elective-I & Program Elective-II)**	Course Code: V25CTT09	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Describe the architecture, modeling techniques, and implementation strategies of data warehouses and OLAP systems, including modern cloud-based approaches. **(K2)**

CO2: Apply statistical and visualization techniques to summarize datasets and perform preprocessing tasks such as cleaning, integration, reduction, and transformation. **(K3)**

CO3: Develop and evaluate classification models using decision trees, Bayesian classifiers, and rule-based methods to address predictive analytics problems. **(K2)**

CO4: Discover meaningful associations and sequential patterns in datasets using algorithms like Apriori, FP-Growth, and sequential pattern mining techniques. **(K3)**

CO5: Implement clustering techniques such as K-means, hierarchical clustering, and DBSCAN, and analyze advanced data mining methods for text, spatial, and graph data. **(K3)**

UNIT-I: Data Warehousing and Online Analytical Processing: Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse; Data Mining Methodologies: CRISP-DM and SEMMA, Comparison of Data Mining Methodologies. Statistical Limits on Data Mining, Introduction to Predictive Analytics, Technologies, Applications, Major issues (Text Book- 1)

UNIT-II: Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity. **Data Preprocessing:** An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization. (Text Book- 1)

UNIT-III: Classification: General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection. (Text Book- 2)

UNIT-IV: Association Analysis: Problem Definition, Frequent Itemset Generation, Rule Generation: Confidence-Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm. Sequential Patterns: Preliminaries, Sequential Pattern Discovery. (Text Book- 2)

UNIT-V: Cluster Analysis: Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bi-secting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm, DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses. Mining rich data types: Mining text data, Spatial-temporal data, Graph and networks. (Text Book- 2)

Text Books:

1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:

1. Data Mining: Vikram Pudi and P. Radha Krishna, Oxford Publisher.
2. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press, 2013.

Online Resources:

1. http://onlinecourses.nptel.ac.in/noc17_mg24/preview - NPTEL course by Prof. Pabitra Mitra
2. http://www.saedsayad.com/data_mining_map.htm

I SEM	Advanced Computer Architecture **(Program Elective-I & Program Elective-II)**	Course Code: V25CTT10	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain advanced concepts of computer architecture, including parallel and scalable systems. **(K2)**
CO2: Illustrate different parallel architectures. **(K3)**
CO3: Explain parallel programming concepts using appropriate models and languages. **(K3)**
CO4: Apply the design principles of pipelined and multiprocessor systems. **(K3)**
CO5: Demonstrate knowledge of modern computer architectures, I/O devices, and low-level programming concepts. **(K3)**

UNIT-I: Theory of Parallelism: Parallel Computer Models, The State of Computing, Multiprocessors and Multicomputer, Multi-vector and SIMD Computers, PRAM and VLSI Models, Program and Network Properties, Conditions of Parallelism, Program Partitioning and Scheduling, Program Flow Mechanisms, System Interconnect Architectures, Principles of Scalable Performance, Performance Metrics and Measures, Parallel Processing Applications, Speedup Performance Laws, Scalability Analysis and Approaches.

UNIT-II: Hardware Technologies: Processors and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors, Memory Hierarchy Technology, Virtual Memory Technology.

UNIT-III: Bus, Cache, and Shared Memory: Bus Systems, Cache Memory Organizations, Shared Memory Organizations, Sequential and Weak Consistency Models, Pipelining and Superscalar Techniques, Linear Pipeline Processors, Nonlinear Pipeline Processors, Instruction Pipeline Design, Arithmetic Pipeline Design.

UNIT-IV: Parallel and Scalable Architectures: Multiprocessors and Multicomputers, Multiprocessor System Interconnects, Cache Coherence and Synchronization Mechanisms, Three Generations of Multicomputers, Message-Passing Mechanisms, Multi-vector and SIMD Computers, Vector Processing Principles, Multi-vector Multiprocessors, Compound Vector Processing, SIMD Computer Organizations (up to 8.4), Scalable, Multithreaded and Dataflow Architectures, Latency-Hiding Techniques, Principles of Multithreading, Fine-Grain Multicomputers, Scalable and Multithreaded Architectures, Dataflow and Hybrid Architectures.

UNIT-V: Software for Parallel Programming: Parallel Models, Languages, and Compilers, Parallel Programming Models, Parallel Languages and Compilers, Dependence Analysis of Data Arrays, Parallel Program Development and Environments, Synchronization and Multiprocessing Modes. Instruction and System Level Parallelism, Instruction Level Parallelism, Computer Architecture, Basic Design Issues, Problem Definition, Model of a Typical Processor, Compiler-detected Instruction Level Parallelism.

Text Books:

1. "Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability", Kai Hwang and Naresh Jotwani, McGraw Hill Education 3rd Edition. 2015
2. "Computer Architecture and Parallel Processing", Kai Hwang and Faye Briggs, Mc Graw-Hill International Edition, 2000

Reference Books:

1. John L. Hennessy and David A. Patterson, Computer Architecture:A quantitative approach, 5th edition, Morgan Kaufmann Elseveir, 2013
2. Introduction to High Performance Computing for Scientists and Engineers,G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
3. Sima D, Fountain T and Kacsuk P, "Advanced Computer Architectures: A Design Space Approach", Addison Wesley, 2000.

I SEM	Artificial Neural Networks **(Program Elective-I & Program Elective-II)**	Course Code: V25CTT11	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain neural network architectures, including feed forward and feedback networks. **(K2)**

CO2: Apply learning rules to train neural networks. **(K3)**

CO3: Interpret testing methods and performance metrics for neural networks in pattern recognition. **(K2)**

CO4: Apply Self-Organizing Maps (SOM), feature mapping models, algorithms, and simulations for classification. **(K3)**

CO5: Explain neuro-dynamics concepts, including dynamical systems, stability, attractors, and Hopfield models. **(K2)**

UNIT-I: Introduction: A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural Networks.

Learning Process: Error Correction Learning, Memory Based Learning, Hebbian Learning, Competitive, Boltzmann Learning, Credit Assignment Problem, Memory, Adaptation, Statistical Nature of the Learning Process.

UNIT-II: Single Layer Perceptrons: Adaptive Filtering Problem, Unconstrained Organization Techniques, Linear Least Square Filters, Least Mean Square Algorithm, Learning Curves, Learning Rate Annealing Techniques, Perceptron – Convergence Theorem, Relation Between Perceptron and Bayes Classifier for a Gaussian Environment.

Multilayer Perceptron: Back Propagation Algorithm, XOR Problem, Heuristics, Output Representation and Decision Rule, Computer Experiment, Feature Detection.

UNIT-III: Back Propagation: Back Propagation and Differentiation, Hessian Matrix, Generalization, Cross Validation, Network Pruning Techniques, Virtues and Limitations of Back Propagation Learning, Accelerated Convergence, Supervised Learning.

UNIT-IV: Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification.

UNIT-V: Neuro Dynamics: Dynamical Systems, Stability of Equilibrium States, Attractors, Neuro Dynamical Models, Manipulation of Attractors as a Recurrent Network Paradigm.

Hopfield Models: Hopfield Models, Computer Experiment.

Text Books:

1. Neural Networks a Comprehensive Foundations, Simon Haykin, PHI edition.

Reference Books:

1. Artificial Neural Networks -B. Vegnanarayana Prentice Hall of India PLtd 2005
2. Neural Networks in Computer Intelligence, LiMinFu MCGRAW HILLE DUCATION 2003
3. Neural Networks-James A Freeman David M S Kapura Pearson Education2004.
4. Introduction to Artificial Neural Systems Jacek M. Zurada, JAICO Publishing House Ed.2006.

I SEM	Data Structures and Algorithms Analysis Lab	Course Code: V25CTL01	**L**	**T**	**P**	**C**
			0	1	2	2

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Construct various data structures such as linked lists, stacks, queues, and trees using Java. **(K3)**
CO2: Apply sorting and searching algorithms, including Quick Sort, Merge Sort, and Heap Sort, to solve computational problems. **(K3)**
CO3: Examine advanced data structures such as graphs, AVL trees, and B-trees to solve complex problems. **(K3)**
CO4: Demonstrate the use of hashing and dictionary (ADT) operations to manage and retrieve data efficiently. **(K3)**

List of Experiments

1. Construct a Java program to perform various operations on a single linked list.
2. Develop a Java program for the following:
   a) Reverse a linked list b) Sort the data in a linked list
   c) Remove duplicates d) Merge two linked lists

3. Develop a Java program to perform various operations on a doubly linked list.
4. Develop a Java program to perform various operations on a circular linked list.
5. Demonstrate a Java program for performing various operations on stack using linked list.
6. Demonstrate a Java program for performing various operations on queue using linked list.
7. Develop a Java program for the following using stack:
   a) Infix to postfix conversion b) Expression evaluation
   c) Obtain the binary number for a given decimal number

8. Develop a Java program to perform various operations on Binary Search Tree using recursive and non-recursive methods.

9. Construct a Java program to perform the following operations on a graph:
   a) BFS b) DFS

10. Demonstrate a Java program to perform Merge & Heap Sort on given elements.
11. Demonstrate a Java program to perform Quick Sort on given elements.
12. Demonstrate a Java program to perform various operations on AVL trees.
13. Construct a Java program to perform the following operations on B-tree:
    a) Insertion b) Searching
14. Demonstrate recursive and non-recursive functions for Binary Tree traversals.
15. Demonstrate all the functions of Dictionary (ADT) using Hashing.

I SEM	Artificial Intelligence Lab	Course Code: V25CTL02	L	T	P	C
			0	1	2	2

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Develop Python programs to solve classical AI problems such as the Water Jug, Tic-Tac-Toe, Monkey Banana, and Puzzle Problems using search algorithms.**(K3)**

CO2: Apply heuristic and optimization techniques, including the Traveling Salesman Problem, Simulated Annealing, and Hill-Climbing algorithms, to solve complex problems. **(K3)**

CO3: Apply advanced search algorithms like A* and AO* in Python to solve path-finding and decision-making problems. **(K3)**

CO4: Demonstrate reasoning methods, including forward chaining, backward chaining, and the Min-Max algorithm, in expert systems and game playing applications. **(K3)**

List of Experiments

1. Develop a Python program to solve the Water Jug problem using DFS.
2. Develop a Python program to solve the Tic-Tac-Toe problem using BFS.
3. Apply a heuristic approach in Python to solve the Traveling Salesman Problem (TSP).
4. Apply the Simulated Annealing Algorithm in Python to solve optimization problems.
5. Apply the Hill-Climbing Algorithm in Python to solve a Puzzle Problem.
6. Develop a Python program to solve the Monkey Banana Problem.
7. Apply the A* Algorithm in Python for path-finding problems.
8. Apply the AO* Algorithm in Python to solve search problems.
9. Develop a Python program to implement the Min-Max Game Playing Algorithm.
10. Demonstrate forward chaining in an Expert System using Python.
11. Demonstrate backward chaining in an Expert System using Python.

Text Books:

1. Artificial intelligence, A modern Approach, 2nd ed,Stuart Russel, Peter Norvig, Prentice Hall
2. Artificial Intelligence, Saroj Kaushik,1st Edition, CENGAGE Learning,2011.

Reference Books:

1. Artificial intelligence, Structures and Strategies for Complex problem solving,5th Edition, George F Lugar, PEA.
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang,Springer,2017.
3. Artificial Intelligence, A new Synthesis,1st Edition, Nils J Nilsson,Elsevier,1998.
4. Artificial Intelligence-3rd Edition, Rich, Kevin Knight, Shiv Shankar B Nair, TMH.
5. Introduction to Artificial Intelligence and Expert Systems,1st Edition, Patterson, Pearson India,2015.

II SEM	**Machine Learning**	Course Code: V25CTL12	**L** 3	**T** 1	**P** 0	**C** 4

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain the fundamental concepts of supervised and unsupervised learning. **(K2)**
CO2: Apply regression, classification, and clustering algorithms to solve real-world problems. **(K3)**
CO3: Apply machine learning models using ensemble methods and Python libraries. **(K3)**
CO4: Analyze time-series, sequence data, and deep learning models for feature representation. **(K3)**
CO5: Develop scalable machine learning solutions for advanced topics, including IoT applications. **(K3)**

UNIT-I: Supervised Learning (Regression/Classification) – Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes, Linear models: Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, Nonlinearity and Kernel Methods, Beyond Binary Classification: Multi-class/Structured Outputs, Ranking.

UNIT-II: Unsupervised Learning – Clustering: K-means/Kernel K-means, Dimensionality Reduction: PCA and kernel PCA, Matrix Factorization and Matrix Completion, Generative Models (mixture models and latent factor models).

UNIT-III: Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, and Random Forests), Python Libraries – Introduction to Python Libraries: TensorFlow, PyTorch, Keras, Scikit-learn, NumPy, SciPy, Matplotlib, Pandas, Seaborn.

UNIT-IV: Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning.

UNIT-V: Scalable Machine Learning (Online and Distributed Learning) – A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference, Recent trends in various learning techniques of machine learning and classification methods for IoT applications, Various models for IoT applications.

Text Books:

1. Machine Learning: A Probabilistic Perspective, 1st edition, Kevin Murphy, MIT Press, 2012
2. The Elements of Statistical Learning, 2nd edition, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer 2009 (freely available online)
3. Pattern Recognition and Machine Learning, 2nd edition, Christopher Bishop, Springer, 2011
4. Programming Collective Intelligence: Building Smart Web 2.0 Applications, 1st edition, Toby Segaran, 2007
5. Building Machine Learning Systems with Python, 1st edition, Willi Richert, Luis Pedro Coelho, 2013
6. Applied Machine Learning, 1st edition, M. Gopal, McGraw Hill Education, 2019

II SEM	OS and UNIX Programming	Course Code: V25CTL13	L	T	P	C
			3	1	0	4

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain the structure, functions, and services of operating systems. **(K2)**

CO2: Apply process scheduling, inter-process communication, and multithreading concepts to solve OS problems. **(K3)**

CO3: Explain memory management, virtual memory and deadlock handling techniques in operating systems.**(K2)**

CO4: Demonstrate file system operations and Unix commands for process and directory management. **(K3)**

CO5: Develop programs to handle process control, signals, and file operations in Unix/Linux environments. **(K3)**

UNIT-I: Operating Systems Overview – Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems. **System Structures** – Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

UNIT-II: Process Concept – Process scheduling, Operations on processes, Inter-process communication, Communication in client-server systems. **Multithreaded Programming** – Multithreading models, Thread libraries, Threading issues. **Process Scheduling** – Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling. **Inter-process Communication** – Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems: Dining philosophers problem, Readers and writers problem.

UNIT-III: Memory-Management Strategies – Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation. **Virtual Memory Management** – Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.
Deadlocks – Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention. **File Systems** – Files, Directories, File system implementation, management and optimization. **Secondary-Storage Structure** – Overview of disk structure and attachment, Disk scheduling, RAID structure, Stable storage implementation.

UNIT-IV: Introduction to Unix – Architecture of Unix, Responsibilities of shell, Unix file system, vi editor. **Unix commands** – Some Basic Commands, file utilities, process utilities, text processing utilities, network utilities, disk utilities, backup utilities, Security by file permissions.

UNIT-V: Files – Introduction, file descriptors, open, creat, read, write, close, lseek, dup2, file status information (stat family), file and record locking (fcntl function), file permissions (chmod, fchmod), file ownership (chown, lchown), links (soft and hard links – symlink, link, unlink). **Directories** – Creating, removing and changing Directories (mkdir, rmdir, chdir), obtaining current working directory (getcwd), Directory contents, Scanning Directories (opendir, readdir, closedir, rewinddir functions). **Process Control** – process identifiers, fork function, vfork function, exit function, wait and waitpid functions, exec functions, user identification. **Signals** – signal handling using signal function, kill and raise, alarm, pause, abort, and sleep functions.

Text Books:

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2013
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008 (for Inter- process Communication and File Systems)
3. Unix the Ultimate Guide, 3rd edition, Sumitabha Das, TMH
4. Advanced Programming in the Unix Environment, W. Richard Stevens
5. Unix Network Programming, W. Richard Stevens

Reference Books:

1. Dhamdhere D M, Operating Systems: A Concept-Based Approach, 3rd edition, Tata McGraw-Hill,2012
2. Stallings W, Operating Systems - Internals and Design Principles, 6th edition, Pearson Education,2009
3. Venkateshmurthy, Introduction to Unix and Shell Programming
4. B.M. Harwani, Unix and Shell Programming, Oxford University Press

e-Resource: NPTEL – Operating Systems (https://nptel.ac.in/courses/106/105/106105214/)

II SEM	Cloud Computing	Course Code: V25CTL14	**L**	**T**	**P**	**C**
			3	1	0	4

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain cloud computing fundamentals, service models, deployment models, and major cloud providers. **(K2)**
CO2: Describe enabling technologies for cloud such as distributed computing, SOA, web services, and virtualization. **(K2)**
CO3: Apply virtualization and container technologies for cloud-based solutions. **(K3)**
CO4: Expalin challenges in cloud computing related to scalability, interoperability, security, and energy efficiency. **(K3)**
CO5: Use advanced cloud concepts like serverless computing, IoT integration, edge/fog computing, and DevOps for modern applications. **(K3)**

UNIT-I: Introduction to Cloud Computing Fundamentals: Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services (IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google App Engine).

UNIT-II: Cloud Enabling Technologies: Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures for parallel computing (SISD, SIMD, MISD, MIMD), elements of distributed computing, inter-process communication, technologies for distributed computing, remote procedure calls (RPC), service-oriented architecture (SOA), web services, virtualization.

UNIT-III: Virtualization and Containers: Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM (e.g., Amazon EC2) and container (e.g., Amazon Elastic Container Service) offerings.

UNIT-IV: Cloud Computing Challenges: Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.

UNIT-V:Advanced Concepts in Cloud Computing: Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g., AWS Lambda) and open-source (e.g., OpenFaaS) serverless platforms, Internet of Things (IoT) applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.

Text Book:

1. Learning Agile – Andrew Stellman, Jill Alison Hart, O'Reilly, 2015

Reference Books:

1. Head First Agile – Andrew Stellman, Jennifer Green, O'Reilly, 2017
2. Essential Scrum: A Practical Guide to the Most Popular Agile Process – Rubin K., Addison-Wesley, 2013

II SEM	Data Science **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL15	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain Python programming concepts including data types, control structures, functions, modules, and OOP. **(K2)**

CO2: Develop GUI-based and networked Python applications using standard libraries**.(K3)**

CO3: Explain NumPy and Pandas for data processing, analysis, and handling missing or hierarchical data. **(K3)**

CO4: Illustrate datasets using data wrangling, aggregation, transformation and merging techniques**.(K3)**

CO5: Explain data visualizations using Matplotlib and Pandas for time series, financial, and economic datasets.**(K3)**

UNIT-I: PYTHON Basics and Programming Concepts – Introducing Python; Types and Operations: Numbers, Strings, Lists, Tuples, Dictionaries, Files, Numeric Types, Dynamic Typing; Statements and Syntax: Assignments, Expressions, Statements, Loops, Iterations, Comprehensions; Functions: Function Basics, Scopes, Arguments, Advanced Functions; Modules: Module Coding Basics, Module Packages, Advanced Module Topics; Classes and OOP: Class, Operator Overloading, Class Designing; Exceptions and Tools: Exception Basics, Exception Coding Details, Exception Objects, Designing with Exceptions, Parallel System Tools.

UNIT-II: GUI Programming – Graphical User Interface: Python GUI development options, Adding Widgets, GUI Coding Techniques, Customizing Widgets; Internet Programming: Network Scripting, Client-Side Scripting, Pymailgui client, Server-Side Scripting, Pymailcgi server; Tools and Techniques: Databases and persistence, Data structures, Text and Language, Python/C Integration.

UNIT-III: Pandas and NumPy – NumPy Basics: Fast Element-wise array functions, Multidimensional Array, Data Processing using arrays, File I/O with arrays; Pandas: Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics, Handling Missing Data, Hierarchical Indexing.

UNIT-IV: Data Preprocessing – Data Loading, Storage, and File Formats: Reading and Writing data in text format, Binary data formats, Interacting with HTML and Web APIs, Interacting with Databases; Data Wrangling: Clean, Transform, Merge, Reshape – Combining and Merging Data Sets, Reshaping and Pivoting, Data Transformation, String Manipulation; Data Aggregation and Group Operations – Group by Mechanics, Data Aggregation, Groupby Operations and Transformations, Pivot Tables and Cross Tabulation.

UNIT-V: Data Visualization – A Brief Matplotlib API Primer, Plotting Functions in Pandas, Time Series, Financial and Economic Data Applications.

Text Books:

1. Learning Python, O'Reilly, Mark Lutz
2. Programming Python, O'Reilly, Mark Lutz
3. Python for Data Analysis, O'Reilly, Wes McKinney

Reference Books:

1. Python: The Complete Reference, Martin C. Brown, McGraw Hill Education
2. Head First Python, Paul Barry, O'Reilly

II SEM	Quantum Computing **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL16	**L** 3	**T** 0	**P** 0	**C** 3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain the history, principles, and foundational concepts of quantum computing. **(K2)**
CO2: Discuss linear algebra, quantum physics, and biological concepts to understand quantum systems. **(K2)**
CO3: Explain qubits and design quantum circuits including single and multiple qubit gates. **(K3)**
CO4: Illustrate quantum algorithms such as Deutsch, Deutsch–Jozsa, Shor, and Grover for problem-solving. **(K3)**
CO5: Explain quantum error correction techniques and apply quantum information and cryptography protocols. **(K3)**

UNIT-I: History of Quantum Computing – Importance of Mathematics, Physics, and Biology. Introduction to Quantum Computing – Bits vs Qubits, Classical vs Quantum logical operations.

UNIT-II: Background Mathematics – Basics of Linear Algebra, Hilbert Space, Probabilities, and Measurements.
Background Physics – Paul's Exclusion Principle, Superposition, Entanglement and Supersymmetry, Density Operators and Correlation, Basics of Quantum Mechanics, Measurements in bases other than computational basis.
Background Biology – Basic concepts of Genomics and Proteomics (Central Dogma).

UNIT-III: Qubit – Physical implementations of Qubit, Qubit as a quantum unit of information, The Bloch Sphere. **Quantum Circuits** – Single qubit gates, Multiple qubit gates, Designing quantum circuits, Bell states.

UNIT-IV: Quantum Algorithms – Classical computation on quantum computers, Relationship between quantum and classical complexity classes, Deutsch's Algorithm, Deutsch–Jozsa Algorithm, Shor's Factorization Algorithm, Grover's Search Algorithm.

UNIT-V: Noise and Error Correction – Graph states and codes, Quantum error correction, Fault-tolerant computation. **Quantum Information and Cryptography** – Comparison between classical and quantum information theory, Quantum Cryptography, Quantum Teleportation.

Text Books:
1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

Reference Books:
1. Yanofsky N. S. and Mannucci M. A., Quantum Computing for Computer Scientists
2. Benenti G., Casati G., and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol. II: Basic Tools and Special Topics, World Scientific
3. Pittenger A. O., An Introduction to Quantum Computing Algorithms

II SEM	Object Oriented and Analysis Design (**Program Elective–III & Program Elective–IV**)	Course Code: V25CTL17	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain the complexity of software systems and techniques to bring order in designing complex systems. **(K2)**

CO2: Apply UML modeling principles and structural diagrams to represent software systems. **(K3)**

CO3: Construct class and object diagrams using advanced modeling techniques. **(K3)**

CO4: Analyze behavioral modeling diagrams such as use cases, interactions, and activities for system design. **(K3)**

CO5: Develop architectural models including component and deployment diagrams for software applications. **(K3)**

UNIT- I: Introduction – The Structure of Complex Systems, The Inherent Complexity of Software, Attributes of Complex Systems, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems. **Case Study:** System Architecture – Satellite-Based Navigation.

UNIT- II: Introduction to UML – Importance of Modeling, Principles of Modeling, Object-Oriented Modeling, Conceptual Model of UML, Architecture, and Software Development Life Cycle. **Basic Structural Modeling** – Classes, Relationships, Common Mechanisms, and Diagrams.
Case Study: Control System – Traffic Management.

UNIT- III: Class & Object Diagrams – Terms, Concepts, Modeling Techniques for Class & Object Diagrams. **Advanced Structural Modeling** – Advanced Classes, Advanced Relationships, Interfaces, Types and Roles, Packages. **Case Study:** AI – Cryptanalysis.

UNIT- IV: Basic Behavioral Modeling-I – Interactions, Interaction Diagrams, Use Cases, Use Case Diagrams, Activity Diagrams. **Case Study:** Web Application – Vacation Tracking System.

UNIT-V: Advanced Behavioral Modeling – Events and Signals, State Machines, Processes and Threads, Time and Space, State Chart Diagrams.
Architectural Modeling – Component, Deployment, Component Diagrams, and Deployment Diagrams. **Case Study:** Weather Forecasting.

Text Books:

1. Grady Booch, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston , "Object- Oriented Analysis and Design with Applications",3rd edition, 2013, PEARSON.
2. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dream tech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis &Design, The McGraw-Hill Companies.
4.Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

II SEM	Cryptography and Network Security **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL18	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain security concepts, attacks, services, and classical encryption methods. **(K2)**
CO2: Apply mathematical concepts to understand symmetric and asymmetric cryptography. **(K3)**
CO3: Demonstrate symmetric and asymmetric key algorithms. **(K3)**
CO4: Explain hash functions, MACs, and digital signatures in security. **(K2)**
CO5: Apply security protocols at the Application, Transport, and Network layers. **(K3)**

UNIT - I: Security Concepts: Introduction, need for security, security approaches, principles of security, types of security attacks, security services, security mechanisms, a model for network security.
Cryptography – Classical Encryption Techniques: Symmetric cipher model, substitution techniques, transposition techniques, rotor machines, steganography.

UNIT - II: Introduction to Symmetric Cryptography: Algebraic structures – groups, rings, fields, GF(2n) fields, polynomials.
Mathematics of Asymmetric Cryptography: Primes, checking for primeness, Euler's phi-functions, Fermat's Little Theorem, Euler's Theorem, generating primes, primality testing, factorization, Chinese Remainder Theorem, quadratic congruence, exponentiation and logarithm.

UNIT - III: Symmetric Key Ciphers: Block cipher principles, DES, AES, Blowfish, IDEA, block cipher operation. **Stream Ciphers:** RC4, RC5.
Asymmetric Key Ciphers: Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman key exchange, Elgamal cryptographic system, elliptic curve arithmetic, elliptic curve cryptography.

UNIT - IV: Cryptographic Hash Functions: Applications of cryptographic hash functions, two simple hash functions, requirements and security, hash functions based on cipher block chaining, secure hash algorithms (SHA).
Message Authentication Codes (MACs): Message authentication requirements, message authentication functions, requirements for MACs, security of MACs, MACs based on hash functions (HMAC), MACs based on block ciphers (DAA, CMAC).
Digital Signatures: Digital signatures, Elgamal digital signature scheme, elliptic curve digital signature algorithm, RSA-PSS digital signature algorithm.

UNIT - V: Network and Internet Security: Transport-Level Security: Web security considerations, transport level security, HTTPS, SSH.
IP Security: IP security overview, IP security policy, encapsulating security payload, authentication header protocol.
Electronic Mail Security: Internet mail security, email format, email threats and comprehensive email security, S/MIME, PGP.

Text Books:
1. William Stallings, Cryptography and Network Security – Principles and Practice, 7th Edition, Pearson Education, 2017.
2. Behrouz A. Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, 3rd Edition, McGraw Hill, 2015.

Reference Books:
1. Atul Kahate, Cryptography and Network Security, 3rd Edition, McGraw Hill.
2. Wade Trappe, Lawrence C. Washington, Introduction to Cryptography with Coding Theory, Pearson.
3. Wenbo Mao, Modern Cryptography: Theory and Practice, Pearson.

II SEM	Secure coding **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL19	**L** 3	**T** 0	**P** 0	**C** 3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain secure systems, principles, and common security attacks. **(K2)**
CO2: Demonstrate secure coding practices in C programs. **(K3)**
CO3: Apply secure coding techniques in C++ and Java. **(K3)**
CO4: Classify database and web-related vulnerabilities and remedies. **(K2)**
CO5: Explain requirements and practices for secure software engineering. **(K2)**

UNIT - I: Introduction: Need for secure systems, proactive security development process, security principles to live by, threat modelling.

UNIT - II: Secure Coding in C: Character strings – string manipulation errors, string vulnerabilities and exploits, mitigation strategies for strings. Pointers – mitigation strategies in pointer-based vulnerabilities. Buffer overflow–based vulnerabilities.

UNIT - III: Secure Coding in C++ and Java: Dynamic memory management, common errors in dynamic memory management, memory managers, double-free vulnerabilities, integer security, mitigation strategies.

UNIT - IV: Database and Web-Specific Input Issues: Quoting the input, use of stored procedures, building SQL statements securely, XSS-related attacks and remedies.

UNIT - V: Software Security Engineering: Requirements engineering for secure software, misuse and abuse cases, SQUARE process model, software security practices and knowledge for architecture and design.

Text Book:

1. Writing Secure Code, 2nd Edition, Michael Howard, David LeBlanc, Microsoft Press, 2003

Reference Books:

1. Secure Coding in C and C++, Robert C. Sea cord, 2nd edition, Pearson Education, 2013
2. Software Security Engineering: A guide for Project Managers, 1st ed, Julia H. Allen, Sean J.Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, Addison-Wesley Professional,2008

II SEM	DevOps **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL20	**L** 3	**T** 0	**P** 0	**C** 3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain the basic concepts and tools of DevOps and how it helps in software development.**(K2)**
CO2: Use Git for managing source code and apply tools like SonarQube to check code quality. **(K3)**
CO3: Operate build automation and continuous integration using Jenkins. **(K3)**
CO4: Experiment with continuous delivery and deployment using Docker and testing tools like Selenium.**(K3)**
CO5: Apply tools like Ansible, Kubernetes, Puppet, and Chef to manage and deploy applications easily. **(K3)**

UNIT-I: Introduction to DevOps: Introduction to SDLC, Agile Model, Introduction to DevOps, DevOps Features, DevOps Architecture, DevOps Lifecycle, Understanding Workflow and principles, Introduction to DevOps tools, Build Automation, Delivery Automation, Understanding Code Quality, Automation of CI/CD, Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples.

UNIT-II: Source Code Management (GIT): The need for source code control, the history of source code management, roles and code, source code management system and migrations, what is version control and GIT, GIT installation, GIT features, GIT workflow, working with remote repository, GIT commands, GIT branching, GIT staging and collaboration.
Unit Testing – Code Coverage: JUnit, nUnit & Code Coverage with SonarQube, SonarQube – Code Quality Analysis.

UNIT-III: Build Automation – Continuous Integration (CI): Build Automation, what is CI, why CI is required, CI tools, introduction to Jenkins (with architecture), Jenkins workflow, Jenkins master-slave architecture, Jenkins Pipelines, pipeline basics – Jenkins master, node, agent, and executor freestyle projects & pipelines, Jenkins for continuous integration, create and manage builds, user management in Jenkins, schedule builds, launch builds on slave nodes.

UNIT-IV: Continuous Delivery: Importance of continuous delivery, continuous deployment CD flow, containerization with Docker: introduction to Docker, Docker installation, Docker commands, images & containers, Docker file, running containers, working with containers and publish to Docker Hub. **Testing Tools:** Introduction to Selenium and its features, JavaScript testing.

UNIT-V: Configuration Management – ANSIBLE: Introduction to Ansible, Ansible tasks, roles, Jinja2 templating, vaults, deployments using Ansible.
Containerization Using Kubernetes (OpenShift): Introduction to Kubernetes namespace & resources, CI/CD on OpenShift Container Platform, BC, DC & config maps, deploying apps on OpenShift container pods, introduction to Puppet master and Chef.

Text Books:

1. Joyner, Joseph. DevOps for Beginners: DevOps Software Development Method Guide for Software Developers and IT Professionals, 1st Edition, Mihails Konoplows, 2015.
2. Alisson Machado de Menezes. Hands-on DevOps with Linux, 1st Edition, BPB Publications, India, 2021.

Reference Books:

1.Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective, Addison Wesley; ISBN-10.
2. Gene Kim, Jez Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona, Joakim. Practical DevOps, 1st Edition, Packt Publishing, 2016.
4. Joakim Verona. Practical DevOps, 2nd Edition, Ingram Short Title; 2nd Edition (2018), ISBN-10: 1788392574.
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint, Wiley Publications, ISBN: 9788126579952.

Web Resources:

1.https://archive.nptel.ac.in/courses/106/104/106104220/
2.https://www.tutorialspoint.com/blockchain/index.htm

II SEM	Web Application Security **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL21	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain basic web security concepts like HTTP, encoding, and authentication.**(K2)**
CO2: Apply session and access control methods to protect against security problems. **(K3)**
CO3: Use techniques to find and fix attacks on databases and backend systems. **(K3)**
CO4: Explain logic flaws and XSS attacks, and apply methods to prevent them. **(K3)**
CO5: Use tools and scripts to test for vulnerabilities and prevent data leaks. **(K3)**

UNIT-I: Web application security, handling user access, input, and attacks, HTTP protocol, web functionalities, encoding schemes, mapping the application.
Client-Side Controls: Transmitting data via the client, capturing user data, capturing user data, authentication technologies, design flaws in authentication, implementation flaws in authentication, securing authentication

UNIT-II: Attacks on Session and Access Control
The need for state, weaknesses in token generation handling, hijacking liberal cookie scope securing session management generate strong tokens protect tokens throughout their life cycle log, monitor, and alert, common vulnerabilities, testing with different user accounts, limited access testing direct access to methods, testing controls over static resources.

UNIT-III: Attacking data-stores and back end components
Injecting into Interpreted Contexts, injecting into SQL, injecting into NoSQL, injecting into XPath, injecting into LDAP, Injecting OS Commands, Manipulating File Paths, Injecting into XML Interpreters, Injecting into Back-end HTTP Requests, Injecting into Mail Services

UNIT-IV: Attacking application logic and users
The Nature of Logic Flaws, Real-World Logic Flaws, Avoiding Logic Flaws, Varieties of XSS, XSS Attacks in Action, Finding and Exploiting XSS Vulnerabilities, Preventing XSS Attacks, Inducing User Actions, Capturing Data Cross-Domain, The Same-Origin Policy Revisited, Other Client-Side Injection Attacks, Local Privacy Attacks, Attacking the Browser, Exploiting Non-HTTP Services, Exploiting Browser Bugs

UNIT-V: Automating customized attacks and exploiting information disclosure
Uses for Customized Automation, Enumerating Valid Identifiers, The Basic Approach, Detecting Hits, Scripting the Attack, JAttack, harvesting useful data, fuzzing for common vulnerabilities, burp intruder, barriers to automation, session-handling mechanisms, CAPTCHA controls, exploiting error messages, using public information, gathering published information, using inference, preventing information leakage.

Text Book:
1. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook, Wiley Publishing Inc.

References:
1. Improving Web Application Security: Threats and Countermeasures, Microsoft Corporation.
2. Bryan Sullivan, Vincent Liu, Web Application Security: A Beginner's Guide, McGraw Hill Publishers.
3. Ron Lepofsky, The Manager's Guide to Web Application Security, Apress Publishers.

II SEM	Big Data Analytics **(Program Elective–III & Program Elective–IV)**	Course Code: V25CTL22	**L**	**T**	**P**	**C**
			3	0	0	3

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Describe the concepts, trends, and industry uses of big data, including Hadoop, cloud computing, and analytics applications. **(K2)**

CO2: Apply data modeling, replication, and consistency techniques in NoSQL databases using tools like Cassandra. **(K3)**

CO3: Apply data processing methods in Hadoop and Hive for querying and optimization. **(K3)**

CO4: Apply data transformations, joins, and debugging techniques in Apache Spark for efficient processing. **(K3)**

CO5: Apply performance tuning and stream processing techniques in Spark to handle event-time data and structured streaming. **(K3)**

UNIT-I: What is big data, why big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open source technologies, cloud and big data, mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.

UNIT-II: Introduction to NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schema less databases, materialized views, distribution models, sharding, master-slave replication, peer- peer replication, sharding and replication, consistency, relaxing consistency, version stamps, Working with Cassandra ,Table creation, loading and reading data

UNIT-III: Data formats, analyzing data with Hadoop, scaling out, Architecture of Hadoop distributed file system (HDFS), fault tolerance ,with data replication, High availability, Data locality , Map Reduce Architecture, Process flow, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization. Introduction to Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, Logical joins, Window functions, Optimization, Table partitioning, Bucketing, Indexing, Join strategies.

UNIT-IV: Apache spark- Advantages over Hadoop, lazy evaluation, In memory processing, DAG, Spark context, Spark Session, RDD, Transformations- Narrow and Wide, Actions, Data frames ,RDD to Data frames, Catalyst optimizer, Data Frame Transformations, Working with Dates and Timestamps, Working with Nulls in Data, Working with Complex Types, Working with JSON, Grouping, Window Functions, Joins, Data Sources, Broadcast Variables, Accumulators, Deploying Spark- On-Premises Cluster Deployments, Cluster Managers- Standalone Mode, Spark on YARN , Spark Logs, The Spark UI- Spark UI History Server, Debugging and Spark First Aid

UNIT-V: Spark-Performance Tuning, Stream Processing Fundamentals, Event-Time and State full Processing - Event Time, State full Processing, Windows on Event Time- Tumbling Windows, Handling Late Data with Watermarks, Dropping Duplicates in a Stream, Structured Streaming Basics - Core Concepts, Structured Streaming in Action, Transformations on Streams, Input and Output

Text Books:

1. Big Data, Big Analytics: Emerging, Michael Minnelli, Michelle Chambers, and Ambiga Dhiraj, 1st edition ,2013
2. SPARK: The Definitive Guide, Bill Chambers &MateiZaharia, O'Reilley, 2018-first Edition.
3. Business Intelligence and Analytic Trends for Today's Businesses", Wiley, First edition-2013.
4. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World Polyglot Persistence", Addison-Wesley Professional, 2012
5. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012

Reference Books:

1. "Hadoop Operations", O'Reilley, Eric Sammer, First Edition -2012.
2. "Programming Hive", O'Reilley, E. Capriolo, D. Wampler, and J. Rutherglen, 2012.
3. "HBase: The Definitive Guide", O'Reilley, Lars George, September 2011: First Edition..
4. "Cassandra: The Definitive Guide", O'Reilley, Eben Hewitt, 2010.
5. "Programming Pig", O'Reilley, Alan Gates, October 2011: First Edition.

II SEM	Machine Learning Lab	Course Code: V25CTL03	L	T	P	C
			0	1	2	2

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Explain statistical measures and data preprocessing techniques. **(K2)**
CO2: Apply classification and regression algorithms on datasets. **(K3)**
CO3: Implement clustering algorithms and evaluate their results. **(K3)**
CO4: Analyze the performance of machine learning models using tuning and error measures. **(K4)**

List of Experiments

1. Compute Central Tendency Measures: Mean, Median, Mode. Measure of Dispersion: Variance, Standard Deviation.

2. Apply the following Pre-processing techniques for a given dataset.
   a) Attribute selection b) Handling Missing Values
   c) Discretization d) Elimination of Outliers

3. Apply KNN algorithm for classification and regression.

4. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results.

5. Demonstrate decision tree algorithm for a regression problem.

6. Apply Random Forest algorithm for classification and regression.

7. Demonstrate Naïve Bayes Classification algorithm.

8. Apply Support Vector algorithm for classification.

9. Demonstrate simple linear regression algorithm for a regression problem.

10. Apply Logistic regression algorithm for a classification problem.

11. Demonstrate Multi-layer Perceptron algorithm for a classification problem.

12. Develop the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.

13. Demonstrate the use of Fuzzy C-Means Clustering.

14. Demonstrate the use of Expectation Maximization based clustering algorithm.

II SEM	OS and Unix Lab	Course Code: V25CTL04	L	T	P	C
			0	1	2	2

Syllabus Details

Course Outcomes: After completion of course, students would be able to

CO1: Apply Unix/Linux commands, file structures, and environment settings. **(K3)**
CO2: Apply system calls to write C programs for file and process management. **(K3)**
CO3: Analyze scheduling, memory management, and deadlock problems. **(K4)**
CO4: Analyze and build multi-threaded programs and memory solutions. **(K4)**

List of Experiments

1. a) Describe Unix/Linux general-purpose utility commands such as man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.

   b) Describe the vi editor.
   c) Describe the Bash shell, Bourne shell, and C shell in the Unix/Linux operating system.
   d) Describe the Unix/Linux file system (tree structure).
   e) Describe .bashrc, /etc/bashrc, and Environment settings.

2. Develop a C program that copies a file using standard I/O and system calls.

3. Develop a C program that emulates the ls –l command.

4. Develop a C program to execute two commands concurrently using a command pipe (e.g. ls -l | sort).

5. Apply CPU scheduling algorithms such as Round Robin, SJF, FCFS, and Priority.

6. Operate memory management functions (`fork()`, `wait()`, `exec()`, and `exit()`) using system calls.

7. Construct simulations of multiprogramming with fixed (MFT) and variable (MVT) numbers of tasks.

8. Solve the Banker's algorithm for deadlock avoidance.

9. Demonstrate the Banker's algorithm for deadlock prevention.

10. Calculate file allocation strategies such as sequenced, indexed, and linked methods.

11. Employ shared memory communication between two processes using a C program.

12. Experiment with page replacement algorithms like FIFO, LRU, and LFU.

13. Demonstrate the producer-consumer problem using semaphores in C.

14. Construct a thread using the pthreads library and execute its function.

15. Operate concurrent execution of threads using the pthreads library.