B. TECH.

COMPUTER SCIENCE AND ENGINEERING

FIFTH SEMESTER

STUDY AND EVALUATION SCHEME

B.TECH. COMPUTER SCIENCE AND ENGINEERING

III Year: V Semester

S. No.	Subject Code	Subject name	L	T	P	Sessional Assessment			SEE	Subject Total	Credit
						MST	TA	Total			
THEORY SUBJECT											
1	TCS501	Artificial Intelligence	3	1	0	15	15	30	70	100	4
2	TCS502	Operating System	3	1	0	15	15	30	70	100	4
3	TCSE051 - 054	Departmental Elective- I	3	1	0	15	15	30	70	100	4
4	TCSE501	Theory of Computation	3	0	0	15	15	30	70	100	4
5	THM501	Constitution of India	2	0	0	15	15	30	70	100	0
6	TGP501	General Proficiency	-	-	-	-	-	50	0	50	0
PRACTICAL/DESIGN/DRAWING											
7	TCS551	Artificial Intelligence Lab	0	0	2	15	15	30	70	100	1
8	TCS552	Operating System Lab	0	0	2	15	15	30	70	100	1
9	TCS553	Mini Project or Internship Assessment	0	0	4	15	15	30	70	100	2
		Total	**14**	**3**	**8**	**120**	**120**	**290**	**560**	**850**	**20**

TCSE051 - 054	**Departmental Elective - I**
TCSE-051	Digital Image Processing
TCSE-052	Object Oriented Design and Modeling
TCSE-053	Cloud Computing
TCSE-054	Internet of Things

L- Lecture
T -Tutorial
P-Practical
MST- Mid Semester Test
TA-Teacher's Assessment
SEE- Semester End Examination

ARTIFICIAL INTELLIGENCE
(TCS501)

COURSE OUTCOMES (COs):	
CO 1:	Understand different types of AI agents.
CO 2:	apply various AI search algorithms (uninformed, informed, heuristic, constraint satisfaction, genetic algorithms.
CO 3:	Understand the fundamentals of knowledge representation, reasoning, and machine learning techniques and apply them to real world problems.
CO 4:	Student should be aware of techniques used for classification and clustering.
CO 5:	Apply simple knowledge-based systems using languages like LISP, Prolog, and AI tools like JESS.
Unit	**Topic**
I	Introduction to Artificial Intelligence, Brief history, Various approaches to AI, Areas of application, Simulation of sophisticated and Intelligent Behavior in different area, Problem solving in games, natural language processing, automated reasoning, and visual perception, Knowledge and its role in AI, Heuristic algorithm versus solution guaranteed algorithms, Introduction to soft computing.
II	Representing problems in state space, informed versus uninformed search, Production System Model, Evaluation of the Production System, Depth First Search and Breadth First Search, Heuristics, Heuristic Search Techniques: Hill Climbing, Best First search, A* Algorithm, Branch and Bound, Cryptarithmetic Problem, Means End Analysis, AO* Algorithm, Game Playing: MINMAX Search, Alpha-Beta Pruning, Heuristic Estimation.
III	Knowledge Representation and Reasoning: Propositional Logic, First Order Predicate Logic, Graphs, Associative Network, Semantic Networks, Conceptual Dependencies, Frames, Scripts, Horn Clauses, Introductory Examples from PROLOG, Case Grammar Theory, Production Rules Knowledge Base, The Interface System, Forward and Backward Deduction, Inference System in Propositional and Predicate Logic, Reasoning under Uncertainty.
IV	Understanding Natural Languages, Various Approaches of NLP, Parsing techniques, Context free and transformational grammars, Transition nets, Augmented transition nets, Fillmore's grammars, Grammar free analyzers, Sentence generation, and translation, Introduction to Pattern Recognition, Structured Description, Symbolic Description, Machine Perception, Object Identification, Speech Recognition.
V	Expert Systems: Architecture of Expert System, Representing and using domain knowledge, Expert System Shell, Explanation System, Knowledge Acquisition System, Case study of Existing Expert Systems like DENDRAL, MYCIN, Development of a small Expert System using programming Languages and tools like LISP, PROLOG, JESS.

Text Book (s):

1. N. J. Nilsson, “Artificial Intelligence: A New Synthesis”, Elsevier Publications, Standard Edition,1998.
2. I. Bratko, ―Prolog: Programming for Artificial Intelligence‖, Fourth edition, Addison-Wesley Educational Publishers Inc., 2011.
3. M. Tim Jones, ―Artificial Intelligence: A Systems Approach(Computer Science)‖, Jones and Bartlett Publishers, Inc.; First Edition, 2008
4. Charnick, “Introduction to A.I.”, Addison Wesley, 1985.
5. Rich and Knight, “Artificial Intelligence”, McGraw-Hill Publication, 3rd Edition, 2017.
6. Gerhard Weiss, ―Multi Agent Systems‖, Second Edition, MIT Press, 2013.

OPERATING SYSTEM
(TCS502)

COURSE OUTCOMES (COs):	
CO 1:	Understand the fundamental concepts and components of operating systems.
CO 2:	Analyze and apply CPU scheduling and process management techniques.
CO 3:	Identify and manage deadlock situations using prevention, avoidance, detection, and recovery methods.
CO 4:	Understand and apply memory management techniques, including paging and segmentation.
CO 5:	Describe and manage file systems and I/O operations, including disk scheduling and RAID configurations

Unit	Topic
I	**Introduction to Operating Systems** Computer system overview, concept of an operating system, batch system, multiprogramming, multiprocessing, multi user, time sharing, personal system, real time system, general system architecture, operating system services, system calls, system structure, Approaches to OS design and implementation: Microkernel, Layered, Kernel Approach.
II	**CPU Scheduling:** Concept of process, process states, process state transitions, process control block, threads, mutual exclusion and synchronization, Schedulers, Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Inter process communication Deadlock Problem, Deadlock Characterization, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Methods for Deadlock Handling.
III	**Inter-process Communication:** Race conditions, Critical sections, Mutual exclusion, Critical-section problem, and Algorithmic approach to implementing critical sections, Hardware support for process synchronization, Semaphores, Mutexes, and Monitors. **Classic Problems of Synchronization:** Producers-consumers with bounded buffers problem, Readers-writers problem, Dining-philosophers problem.
IV	**Memory Management:** Introduction, Logical and Physical Address Space, Swapping, Contiguous Memory Allocation, Fragmentation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging. Virtual Memory: Demand Paging, Performance of Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.
V	**File system:** File Concept, Access Methods, Directories, Mounting of File-System, File-System Structure, File-System Implementation, Allocation Methods. I/O Devices, and I/O Subsystems, I/O Buffering, Disk Storage and Disk Scheduling, Disk Management, RAID.

Text Book (s):

1. Operating system, Galvin and Silberschatz, 7th Edition, John Willey 2004
2. Operating Systems-A Concept Based Approach, Dhamdhare, TMH 2006
3. Dhananjay M. Dhamdhere, “Operating Systems A Concept-Based Approach”, TMH
4. Operating System Concepts, Ekta Walia, Khanna Book Publishing 2020
5. Operating Systems –A Design Oriented Approach, Crowley, TMH, 2001

DEPARTMENTAL ELECTIVE - I
(TCSE051-054)
DIGITAL IMAGE PROCESSING
(TCSE051)

COURSE OUTCOMES (COs):	
CO 1:	Understand the fundamentals of visual perception, image models, and sampling techniques.
CO 2:	Explore image enhancement techniques in the spatial domain, including basic gray level functions, histogram processing, and spatial filtering methods such as smoothing and sharpening.
CO 3:	Analyse the principles of image enhancement in the frequency domain, including Fourier transform, frequency domain filtering, and homomorphic filtering.
CO 4:	Understand the fundamentals of color image processing, including color models, transformations, and techniques for color enhancement and segmentation.
CO 5:	Understand the knowledge of image compression techniques, including different coding methods and standards, and apply these methods to optimize image storage and transmission.

Unit	Topic
I	**Introduction and Fundamentals:** Motivation and Perspective, Applications, Components of Image Processing System, Element of Visual Perception, A Simple Image Model, Sampling and Quantization. **Image Enhancement in Spatial Domain:** Introduction; Basic Gray Level Functions – Piecewise-Linear Transformation Functions: Contrast Stretching; Histogram Specification; Histogram Equalization; Local Enhancement; Enhancement using Arithmetic/Logic Operations – Image Subtraction, Image Averaging; Basics of Spatial Filtering; Smoothing - Mean filter, Ordered Statistic Filter; Sharpening – The Laplacian.
II	**Image Enhancement in Frequency Domain:** Fourier Transform and the Frequency Domain, Basis of Filtering in Frequency Domain, Filters – Low-pass, High-pass; Correspondence Between Filtering in Spatial and Frequency Domain; Smoothing Frequency Domain Filters – Gaussian Low-pass Filters; Sharpening Frequency Domain Filters – Gaussian High-pass Filters; Homomorphic Filtering.
III	**Image Restoration:** A Model of Restoration Process, Noise Models, Restoration in the presence of Noise only-Spatial Filtering – Mean Filters: Arithmetic Mean filter, Geometric Mean Filter, Order Statistic Filters – Median Filter, Max and Min filters; Periodic Noise Reduction by Frequency Domain Filtering– Band-pass Filters; Minimum Mean-square Error Restoration.
IV	**Color Image Processing:** Color Fundamentals, Color Models, Converting Colors to different models, Color Transformation, Smoothing and Sharpening, Color Segmentation. Morphological Image Processing: Introduction, Logic Operations involving Binary Images, Dilation and Erosion, Opening and Closing, Morphological Algorithms – Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening.
V	**Image Compression:** Fundamentals, image compression models, Compression methods: Huffman coding, Golomb Coding, Arithmetic Coding, LZW coding, Run-Length coding, Symbol based coding. Error-free compression, lossy predictive coding, image compression standards. **Image Segmentation:** Fundamentals, Point, Line and edge detection. Thresholding: foundation, Basic Global Thresholding, Otsu's Method, Image smoothing to improve global thresholding.

Text Book (s):

1. Rafael C. Gonzalez, Richard E. Woods, Steven L., "Digital Image Processing Using MATLAB", TMH.
2. Anil Jain K. "Fundamentals of Digital Image Processing", PHI.

OBJECT ORIENTED DESIGN AND MODELING
(TCSE052)

COURSE OUTCOMES (COs):	
CO 1:	Analyse information systems in real-world settings and use an object-oriented method for analysis and design.
CO 2:	Understand features of object-oriented design such as encapsulation, polymorphism, inheritance, and UML.
CO 3:	Understand and prepare different types of UML diagrams like use case diagrams, interaction diagrams, nested state diagrams, state chart diagrams, activity diagram etc.
CO 4:	Understand and appreciate the use of Design Patterns in the Software Development.
CO 5:	Understand the core and advance Java Programming features and apply them
Unit	**Topic**
I	Object Oriented Design and Modeling: Object oriented fundamentals, Objects and Classes, Object-oriented Design Process, importance of modeling, principles of modeling, OOAD Methods, Software Development Life Cycle, Introduction to Unified Process, Introduction to UML: UML Terminology, conceptual model of the UML, Use of UML in Unified Process.
II	Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams. Class and Object Diagrams: Terms, concepts, modeling techniques for Class and Object Diagrams, Links and Associations, Link Attributes and Link Classes, Generalization and Inheritance, Aggregation and Composition, Qualified Association, Handling multiplicity in Object creation, Abstract Classes, Specifying constraints in Class Diagrams,
III	Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages, Use Case Modeling: Use Cases and Use Case Diagrams.
IV	Behavioral Modeling: Interactions and Interaction Diagrams, Use-Case Realization: Scenario, Events Trace Diagram, Collaboration Diagrams, State Chart Diagrams, Nested State Diagrams, Activity Diagrams, Advanced Behavioral Modeling Concepts, Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.
V	Elementary Design Patterns, The MVC Architecture Pattern, Features of Elegant Software Design: Elegant variable, Elegant Classes, Elegant Methods, Elegant Packages, Introduction to Object Oriented Software Quality Metrics.

Text Book (s):

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.
2. Mark Priestley: Practical Object-Oriented Design with UML, TATA Mc-GrawHill.
3. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, Pearson Education.

CLOUD COMPUTING
(TCSE053)

COURSE OUTCOMES (COs):	
CO 1:	Analyze the evolution, and underlying principles of cloud computing and its characteristics.
CO 2:	Evaluate client systems, networks, server systems, and security from a service perspective within cloud computing.
CO 3:	Understand the conceptual models and functionalities of Infrastructure.
CO 4:	Assess resource management techniques and security challenges associated with inter-cloud resource provisioning.
CO 5:	Use and Examine cloud computing platforms along with current trends and research.
Unit	**Topic**
I	**Introduction To Cloud Computing:** Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristic, cloud environment.
II	Cloud Computing Technology - Client systems, Networks, server systems and security from services perspectives; Accessing the cloud with platforms and applications; cloud storage.
III	**Working with Cloud-** Infrastructure as a Service – conceptual model and working Platform as a Service – conceptual model and functionalities Software as a Service – conceptual model and working Technologies and Trends in Service provisioning with clouds.
IV	**Resource Management and Security In Cloud:** Inter Cloud Resource Management – Resource Provisioning and Resource Provisioning Methods – Global Exchange of Cloud Resources – Security Overview – Cloud Security Challenges
V	Case studies-Microsoft Azure, Google App Engine and Open-source clouds- Open-Nebula and Eucalyptus, Current trends and research.

Text Book (s):

1. Gautam Shroff, Enterprise Cloud Computing Technology Architecture Applications, Cambridge.
2. Toby Velte, Anthony Velte, Robert Elsenpeter, Cloud Computing, A Practical Approach.
3. Dimitris N. Chorafas, Cloud Computing Strategies

.

INTERNET OF THINGS
(TCSE054)

COURSE OUTCOMES (COs):	
CO 1:	Understand the fundamental concepts of IoT, including its functionality and ecosystem.
CO 2:	Understand various IoT hardware components, including computing platforms, sensors, and actuators.
CO 3:	Understand Compare and contrast wireless communication protocols used in IoT, including Bluetooth, Wi-Fi, ZigBee, and cellular IoT.
CO 4:	Analyze IoT solution architecture models and their application in real-world scenarios.
CO 5:	Explore IoT applications and case studies across different industries fields
Unit	**Topic**
I	**Introduction to IoT:** What is IoT, how does it work? Difference between Embedded device and IoT device, Properties of IoT device, IoT Ecosystem, IoT Decision Framework, IoT Solution Architecture Models, Major IoT Boards in Market
II	**Elements of IoT Hardware Components-** Computing (Arduino, Raspberry Pi), Sensing, Actuation, I/O interfaces.
III	**Communication Protocols used in IoT:** Types of wireless communication, Major wireless Shortrange communication devices, properties, comparison of these devices (Bluetooth, WIFI, ZigBee, 6LoWPAN), Major wireless Long-range communication devices, properties, comparison of these devices (Cellular IoT, LPWAN
IV	**IoT Applications:** Industrial Internet 4.0, Applications such as: Smart home, wearables, smart city, smart grid, connected car, connected health (digital health, telehealth, telemedicine), smart retail
V	**IoT Case Studies:** IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation

Text Book (s):

1. Vijay Madisetti, Arshdeep Bahga, Ïnternet of Things, “A Hands-on Approach”, University Press
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, “Introduction to Internet of Things: A practical Approach”, ETI Labs
3. Pethuru Raj and Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases”, CRC Press
4. Jeeva Jose, “Internet of Things”, Khanna Publishing House, Delhi
5. Adrian McEwen, “Designing the Internet of Things”, Wiley
6. Raj Kamal, “Internet of Things: Architecture and Design”, McGraw Hill
7.CunoPfister, Getting Started with the Internet of Things, O Reilly Media, 2011
8.. Kyung, C.-M., Yasuura, H., Liu, Y., Lin, Y.-L., Smart Sensors and Systems, Springer International Publishing, 2015

THEORY OF COMPUTATION
(TCSE501)

COURSE OUTCOMES (COs):	
CO 1:	Explain and classify mathematical models for representing finite state systems
CO 2:	Design various types of automata and write regular expressions for regular languages, and interconvert automata and regular expressions
CO 3:	Explain, analyze, and design context free grammars for context free languages
CO 4:	Design of PDA for various competitive level problems
CO 5:	Design and analyze Turing machines and explain the limits of computing
Unit	**Topic**
I	**Introduction to Finite Automata:** Introduction to Finite Automata, strings, alphabets and languages, state tables and diagram, NDFA and DFA concepts, Language of DFA and NFA, Conversion of NFA to DFA, Minimization of FA, Mealy and Moore machines, state and machine equivalence.
II	**Regular Expression:** Introduction to Regular Expressions, Identities for Regular expressions, Arden's Theorem, Regular expression to FA, DFA to Regular expression, Properties of Regular Languages, Pumping Lemma for Regular sets.
III	**Context Free Grammars and Languages:** Introduction to Grammars and Languages, Chomsky Classification of languages, Context free Grammar, Left Most and Right Derivations, Derivation tress, Conversion of FA into CFG and Regular Grammar into FA, Ambiguity, Simplification of CFG, Conversion to Chomsky and Greibach Normal form, Pumping Lemma for context free languages.
IV	**Push Down Automata (PDA):** Introduction to Push Down Automaton (PDA) for Context Free languages, Basic Design of a PDA, Instantaneous configuration of PDA, Construction of PDA for Context free languages, conversion from CFG to PDA.
V	**Turing machines (TM) and Computability:** Basic model, definition and representation of TM, Language acceptance by TM, Variants of Turing Machine, Universal TM, Church's Thesis, Recursive and Recursively Enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TM, Class P and NP Languages, NP Completeness, Some NP Complete Problems.

Text Book (s):

1. J.E. Hopcroft, Motwani and J.D. Ullmann, "Introduction to Automata Theory, Languages and Computation", Narosa Publications
2. H.R. Lewis and C.H. Papadimitrou, "Elements of the Theory of Computation", PHI
3. John C. Martin, "Introduction to Languages and the Theory of Computation", TMH
4. Michael Sipser, "Introduction to the Theory of Computation ", Thomson Learning, PWS publishing company
5. D.A. Cohen , "Introduction to Computer Theory", John Wiley.
6. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data," O'Reilly Media.

CONSTITUTION OF INDIA
(THM501)

COURSE OUTCOMES (COs):	
CO 1:	Identify and explore the basic features and modalities about Indian constitution.
CO 2:	Differentiate and relate the functioning of Indian parliamentary system at the centre and state level.
CO 3:	Differentiate different aspects of Indian Legal System and its related bodies.
CO 4:	Knowledge about Human Rights.
CO 5:	Knowledge about Professional Ethics.
Unit	**TOPICS**
I	**Constitution of India** Introduction to the Constitution of India, The Making of the Constitution and Salient features of the Constitution, Preamble to the Indian Constitution Fundamental Rights and its limitations.
II	**Fundamental Duties and Union Executives** Directive Principles of State Policy and Relevance of Directive Principles State Policy Fundamental Duties. Union Executives — President, Prime Minister, Parliament, Supreme Court of India.
III	**State Legislature and Electoral Process** State Executives — Governor Chief Minister, State Legislature High Court of State, Electoral Process in India, Amendment Procedures, 42nd, 44th, 74th, 76th, 86th and9Ist Amendments.
IV	**Human Rights** Special Provision for SC and ST Special Provision for Women, Children and Backward Classes Emergency Provisions. Human Rights — Meaning and Definitions, Legislation Specific Themes in Human Rights- Working of National Human Rights Commission in India, Powers and functions of Municipalities, Panchayats and Co - Operative Societies.
V	**Professional Ethics** Scope and Aims of Engineering Ethics, Responsibility of Engineers Impediments to Responsibility. Risks, Safety and liability of Engineers, Honesty, Integrity and Reliability in Engineering.

Text Book(s):

1. J. N. Pandey, "Constitutional Law of India," Central Law Agency.
2. M. P. Jain, "Indian Constitutional Law," LexisNexis.
3. Granville Austin, "The Indian Constitution: Cornerstone of a Nation," Oxford University Press.
4. D. D. Basu, "Introduction to the Constitution of India," LexisNexis.
5. V. N. Shukla, "Constitution of India," Eastern Book Company.
6. H. M. Seervai, "Constitutional Law of India: A Critical Commentary," Universal Law Publishing.

ARTIFICIAL INTELLIGENCE LAB
(TCS551)

LIST OF THE EXPERIMENTS:

1. Write a python program to implement Breadth First Search Traversal

2. Write a python program to implement Water Jug Problem

3. Write a python program to remove punctuations from the given string

4. Write a python program to sort the sentence in alphabetical order

5. Write a program to implement Hangman game using python.

6. Write a program to implement Tic-Tac-Toe game using python.

7. Write a python program to remove stop words for a given passage from a text file using NLTK

8. Write a python program to implement stemming for a given sentence using NLTK

9. Write a python program to POS (Parts of Speech) tagging for the give sentence using NLTK

10. Write a python program to implement Lemmatization using NLTK

11. Write a python program to for Text Classification for the give sentence using NLTK

12. Design and Development of Expert System using JESS

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

OPERATING SYSTEM LAB
(TCS552)

LIST OF THE EXPERIMENTS:

1. Implement CPU Scheduling Policies:
    i. SJF
    ii. Priority
    iii. FCFS
    iv. Multi-level Queue

2. Implement any file allocation technique:
    - Linked
    - Indexed
    - Contiguous

3. Implementation of contiguous allocation techniques:
    i. Worst-Fit
    ii. Best- Fit
    iii. First- Fit

4. Calculation of external and internal fragmentation
    i. Free space list of blocks from system
    ii. List process file from the system

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

B. TECH.

COMPUTER SCIENCE AND ENGINEERING

SIXTH SEMESTER

STUDY AND EVALUATION SCHEME

B.TECH. COMPUTER SCIENCE AND ENGINEERING

III Year: VI Semester

S. No.	Subject code	Subject name	L	T	P	Sessional Assessment			SEE	Subject Total	Credit
						MST	TA	Total			
THEORY SUBJECT											
1	TCS601	Computer Networks	3	1	0	15	15	30	70	100	4
2	TCS602	Software Engineering	3	1	0	15	15	30	70	100	4
3	TCSE601	Compiler Design	3	1	0	15	15	30	70	100	4
4	TCSE061 - 064	Departmental Elective-II	3	0	0	15	15	30	70	100	3
5	TOE 061 - 064	Open Elective –I	3	0	0	15	15	30	70	100	3
6	THM601	Indian Tradition, Culture and Society	2	0	0	15	15	30	70	100	0
7	TGP601	General Proficiency	-	-	-	-	-	50	0	50	0
PRACTICAL/DESIGN/DRAWING											
7	TCS651	Computer Networks Lab	0	0	2	15	15	30	70	100	1
8	TCS652	Software Engineering Lab	0	0	2	15	15	30	70	100	1
		Total	**17**	**3**	**4**	**120**	**120**	**290**	**560**	**850**	**20**

The internship (5 - 6 weeks) conducted during summer break after VI semester and will be assessed during VII

TCSE061 - 064	**Departmental Elective - II**
TCSE-061	Robotics
TCSE-062	Data Science
TCSE-063	Mobile Computing
TCSE-064	Pattern Recognition

TOE061 - 064	**Open Elective - I**
TOE-061	Remote Sensing
TOE-062	Bioterrorism and National Security
TOE-063	Quality Management
TOE-064	Soft Computing

COMPUTER NETWORKS

(TCS601)

COURSE OUTCOMES (COs):	
CO 1:	Understand fundamental concepts of computer network technology.
CO 2:	Comprehend various network topologies and protocols.
CO 3:	Examine the roles and functions of different network devices.
CO 4:	Analyze the architecture and principles underlying contemporary computer networks.
CO 5:	Understand future Internet requirements and their implications for computer network architecture.

Unit	Topic
I	**Computer Networks and The Internet:** Networks, Internet, Network Components, Network Categories, network edge, Network Core, Delay, Loss and throughput in Packet-Switched Networks, Protocol Layers and their Service Models
II	**Data Link Layer:** Introduction to the link layer; Error Detection and Correction Techniques; Multiple Access links and Protocols; Switched local area networks.
III	**Network Layer:** Introduction; Virtual circuit and datagram networks; Router, Internet Protocol (IP): Forwarding and Addressing in the Internet, Routing Algorithms; Routing in the Internet, Broadcast and Multicast Routing.
IV	**Transport Layer:** Introduction and Transport-Layer Services; Multiplexing and Demultiplexing; Connectionless Transport: UDP; Principles of Reliable of Data Transfer; Connection-Oriented Transport: TCP; Principles of Congestion Control, TCP Congestion Control.
V	**Application Layer:** Principles of Network Applications, WWW and HTTP, Non-Persistent and Persistent Connections, Cookies, Web Caching, File Transfer, Remote Logging, Electronic Mail in the Internet, Domain Name System, Security: Introduction, Cryptography and Cryptanalysis, Public Key Cryptography Algorithms, RSA Algorithm, DES, Authentication and Authorization

Text Book (s):

1. A.S. Tanenbaum, DJ Wetherall, Computer Networks, Prentice-Hall, 2010. .
2. LL Peterson, BS Davie, Computer Networks: A Systems Approach, Morgan-Kauffman,2011.
3. G. Keiser, “Local Area Networks”, 2nd Edition, TMH 2002
4. D. Bertesekas and R. Gallager, “Data Networks”, 2nd Edition, PHI 2000
5. William Stallings, “Data and Computer Communication”, PHI, 10th Edition 2013
6. B.A. Forouzan, “Data communications and networking”, TMH, 5th Edition2012
7. B.A. Forouzan, “Local Area Networks”, TMH. 2002
8. B.A. Forouzan, “TCP/IP Protocol Suite”, TMH.2004

SOFTWARE ENGINEERING
(TCS602)

COURSE OUTCOME (COs):	
CO 1:	Understand and apply fundamental software engineering principles and practices in the design, development, and maintenance of software systems.
CO 2:	Develop and analyze software requirements specifications, ensuring they are complete, consistent, and feasible.
CO 3:	Design software architecture and components using appropriate modelling techniques, such as UML, to create effective and efficient software solutions.
CO 4:	Implement software systems using modern programming languages and development environments while adhering to best coding practices.
CO 5:	Conduct software testing and debugging to ensure software reliability, performance, and correctness.

Unit	Topic
I	Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes. Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.
II	Software Requirement Specifications (SRS). Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Attributes, Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.
III	Software Design: Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.
IV	Software Testing: Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.
V	Software Maintenance and Software Project Management, Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

Text Book (s):

1. Software Engineering: A Practitioner's Approach, Pressman Roger, TMH.
2. An Integrated Approach to Software Engineering, Pankaj Jalote. Narosa Pub.
3. Software Engineering Concepts: Richard Fairly, Tata McGraw Hill.
4. Schaum's Series, "Software Engineering", TMH
5.Ghezzi, Carlo and Others, "Fundamentals of Software Engineering", PHI
6.Alexis, Leon and Mathews Leon, "Fundamental of Software Engineering", Vikas Publication.

COMPILER DESIGN
(TCSE601)

COURSE OUTCOMES (COs):	
CO 1:	Acquire knowledge of different phases, passes and different compiler tools
CO 2:	Understand the parser and its types i.e., Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table.
CO 3:	Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.
CO 4:	Acquire knowledge about Intermediate Code Forms and Run-time Environments.
CO 5:	Techniques used in Code Generation and Code Optimization
Unit	**Topic**
I	**Introduction to Compiler and Lexical Analysis Phase:** Introduction to compilation, Language processing system, Analysis of the Source Program, Phases and Passes in compilers, compiler construction tools. Introduction to Lexical analysis, Input buffering, tokens, lexemes and pattern, FA and Regular Expressions, NFA to DFA, Minimization, Specification and recognition of tokens, Design of lexical analyzer generator.
II	**Syntax Analysis Phase (Parsing)**: Role and position of a Parser, A simple Backtracking parser, Predictive Parsing, A review of Context Free Grammar, Derivation tree, Ambiguity. Parsing approaches. Top-down Parsing: LL Parsing; Bottom-Up Parsing technique: LR Parsing, SLR, CLR and LALR Parsing, Error recovery strategies, Yacc: an LALR Parser generator.
III	**Semantic Analysis and Type Checking:** Syntax Directed Definitions and translations, Attributes and Attribute grammar, construction of syntax trees, bottom-up evaluation of S attributed definition, Type Checking: Type systems, Specification of simple type checker, Type checking for expression and statements, type conversions.
IV	**Intermediate Code Generation:** Intermediate Code Forms, Intermediate Code Generation for Arithmetic Expression, Boolean Expression, if-then-else, goto, while statements etc. Run-Time Environment.
V	**Code Generation and Code Optimization:** Issues in the design of Code Generator, Basic Blocks and Flow Graph, Register Allocation and Assignment. Sources of Optimization, Optimization of Basic Blocks, Global Data Flow Analysis – Reaching Definition, Available Expression, Live Variable etc. Loops in Flow Graph.

Text Book (s):

1. Aho, Lam, Sethi, Ullman, “Compilers: Principles, Techniques and Tools”, Pearson Education.
2. Aho, Ullman, “Principles of Compiler Design”, Narosa Publising House.
3. Steven S. Muchnick, “Advanced Compiler Design Implementation”, Harcourt Asia Pte Ltd.
4. K. Muneeswaran, Compiler Design, First Edition, Oxford University Press
5. J.P. Bennet, “Introduction to Compiler Techniques”, Second Edition, McGraw-Hill,2003.
6. Ben Fry, "Visualizing Data: Exploring and Explaining Data with the Processing Environment," O'Reilly Media.

DEPARTMENTAL ELECTIVE - II
(TCSE061-064)
ROBOTICS
(TCSE061)

COURSE OUTCOMES (COs):	
CO 1:	Understand the types, components, and classification of robots, including kinematic systems and control systems, while addressing social issues and safety in robotics.
CO 2:	Analyze robot kinematics and dynamics, focusing on translation and rotation representation, coordinate transformation, and dynamic modeling using Euler-Lagrange formulation.
CO 3:	Explore various sensors and vision systems used in robotics, including contact, proximity, position, and vision applications, with a focus on camera calibration and image formation geometry.
CO 4:	Apply control theories to robotics, including P, PD, PID, and advanced control laws, and understand the selection and function of various robot actuation systems.
CO 5:	Develop skills in control hardware and interfacing, focusing on embedded systems architecture, integration with sensors and actuators, and programming for robot applications.

Unit	**Topic**
I	Introduction to Robotics: Types and components of a robot, Classification of robots, closed-loop and openloop control systems, Kinematics systems; Definition of mechanisms and manipulators, Social issues and safety.
II	Robot Kinematics and Dynamics: Kinematic Modelling: Translation and Rotation Representation, Coordinate transformation, DH parameters, Jacobian, Singularity, and Statics, Dynamic Modelling: Equations of motion: Euler-Lagrange formulation
III	Sensors and Vision System: Sensor: Contact and Proximity, Position, Velocity, Force, Tactile etc, Introduction to Cameras, Camera calibration, Geometry of Image formation, Euclidean/Similarity/Affine/Projective transformations, Vision applications in robotics.
IV	Robot Control: Basics of control: Transfer functions, Control laws: P, PD, PID, Non-linear and advanced controls, Robot Actuation Systems: Actuators: Electric, Hydraulic and Pneumatic; Transmission: Gears, Timing Belts and Bearings, Parameters for selection of actuators.
V	Control Hardware and Interfacing: Embedded systems: Architecture and integration with sensors, actuators, components, Programming for Robot Applications

Text Book (s):

1. Saha, S.K., “Introduction to Robotics, McGraw-Hill Higher Education, New Delhi.
2. Ghosal, A., “Robotics”, Oxford, New Delhi.
3. Niku Saeed B., “Introduction to Robotics: Analysis, Systems, Applications”, PHI.
4. Mittal R.K. and Nagrath I.J., “Robotics and Control”, Tata McGraw Hill.
5. Mukherjee S., “Robotics and Automation”, Khanna Publishing House, Delhi.
6. Craig, J.J., “Introduction to Robotics: Mechanics and Control”, Pearson, New Delhi.

DATA SCIENCE
(TCSE062)

COURSE OUTCOMES (COs):	
CO 1:	Understand the core concepts of data science, including big data traits, web scraping techniques, and the distinctions between analysis and reporting.
CO 2:	Explore Python toolkits for data science such as Matplotlib, NumPy, Scikit-learn, and NLTK, and apply these tools to visualize data using various chart types.
CO 3:	Analyze mathematical foundations relevant to data science, including linear algebra, statistics, probability, and hypothesis testing, and apply these concepts to data analysis.
CO 4:	Assess machine learning models using metrics for classification errors and time series analysis.
CO 5:	Apply data science methodologies to real-world case studies such as weather forecasting, stock market prediction, object recognition, and real-time sentiment analysis, to draw actionable insights.
Unit	**Topic**
I	Introduction to Data Science: Concept of Data Science, Traits of Big data, Web Scraping, Analysis vs Reporting.
II	Introduction to Programming Tools for Data Science : Toolkits using Python: Matplotlib, NumPy, Scikit-learn, NLTK, Visualizing Data: Bar Charts, Line Charts, Scatterplots, Working with data: Reading Files, Scraping the Web, Using APIs (Example: Using the Twitter APIs), Cleaning and Munging, Manipulating Data, Rescaling, Dimensionality Reduction.
III	Mathematical Foundations: Linear Algebra: Vectors, Matrices, Statistics: Describing a Single Set of Data, Correlation, Simpson's Paradox, Correlation and Causation, Probability: Dependence and Independence, Conditional Probability, Bayes's Theorem, Random Variables, Continuous Distributions, The Normal Distribution, the Central Limit Theorem, Hypothesis and Inference.
IV	Machine Learning: Overview of Machine learning concepts – Over fitting and train/test splits, Types of Machine learning – Supervised, Unsupervised, Reinforced learning, Introduction to Bayes Theorem, Linear Regression- model assumptions, regularization (lasso, ridge, elastic net), Classification and Regression algorithms- Naïve Bayes, K-Nearest Neighbors, logistic regression, support vector machines (SVM), decision trees, and random forest, Classification Errors, Analysis of Time Series- Linear Systems Analysis, Nonlinear Dynamics, Rule Induction, Neural Networks- Learning And Generalization, Overview of Deep Learning.
V	Case Studies of Data Science Application: Weather forecasting, Stock market prediction, Object recognition, Real Time Sentiment Analysis.

Text Book (s):

1. Joel Grus, "Data Science from Scratch: First Principles with Python", O'Reilly Media
2. Jain V.K., "Data Sciences", Khanna Publishing House, Delhi.
3. Jain V.K., "Big Data and Hadoop", Khanna Publishing House, Delhi.
4. Jeeva Jose, "Machine Learning", Khanna Publishing House, Delhi.
5. Chopra Rajiv, "Machine Learning", Khanna Publishing House, Delhi.
6. Text Compression1st Edition by Timothy C. Bell Prentice Hall.

MACHINE LEARNING
(TCSE063)

COURSE OUTCOMES (COs):	
CO1:	Understand and explain the fundamental concepts of machine learning, including supervised, unsupervised, and reinforcement learning, and their applications in solving real-world problems.
CO2:	Apply various machine learning algorithms to build predictive models using appropriate datasets.
CO3:	Analyze and evaluate the performance of machine learning models using various metrics and improve model accuracy through various techniques.
CO4:	Implement and utilize machine learning libraries and frameworks to develop and deploy machine learning solutions in practical scenarios.
CO5:	To optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Unit	Topic
I	Basic Concepts, Introduction to Machine Learning, Applications of ML, Design Perspective and Issues in ML, Supervised, Unsupervised, Semi-supervised learning with applications and issues, A Formal Learning Model, The Runtime of Learning.
II	Model (or hypothesis) representation, decision boundary, cost function, gradient descent, regularization, Diagnostic: debugging a learning algorithm, evaluating a hypothesis (Model selection), training/validating/testing procedures, diagnosing bias versus variance and vice versa, regularization and bias/variance, learning curves, Accuracy and Error measures: classifier accuracy measures, predictor error measure, evaluating the accuracy of a classifier or predictor, Confusion metric, precision, recall, tradeoff between both, accuracy.
III	Decision Tree: representation, hypothesis, issues in Decision Tree Learning, Pruning, Rule extraction from Tree, Learning rules from Data, Probabilistic classifier: Bayes rule, Maximum Likelihood Estimation, case study, Support Vector Machine, Nearest Neighbor.
IV	Clustering: Unsupervised learning technique, Similarity and Distance Measures, k-means and k-medoids algorithm, optimization objective, random initialization, choosing value of k, EM algorithm Bayesian networks, bag of words classifiers, N-gram models; Markov and Hidden Markov models, Graphical Models, Combining Multiple Learners.
V	Reinforcement Learning: Elements of Reinforcement Learning, Model-Based Learning, Temporal Difference Learning, Generalization, Design and Analysis of Machine Learning Experiments.

Text Book (s):

1. Ethem Alpaydin, Introduction to Machine Learning, PHI, 2015.
2. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and TechniquesMorgan Kaufmann2005.
3. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

PATTERN RECOGNITION
(TCSE064)

COURSE OUTCOMES (COs):	
CO 1:	Understand the mathematical foundations of pattern recognition
CO 2:	Understand statistical pattern recognition methods and discriminant functions, to determine their effectiveness in various scenarios.
CO 3:	Apply parameter estimation techniques.
CO 4:	Implement nonparametric techniques for pattern recognition and fuzzy classification.
CO 5:	Explore unsupervised learning techniques, clustering methods and algorithms in real-world applications using appropriate evaluation metrics and techniques.
Unit	**Topic**
I	Basics of pattern recognition, Design principles of pattern recognition system,Learning and adaptation, Pattern recognition approaches, Mathematical foundations – Linear algebra, Probability Theory, Expectation, mean and covariance, Normal distribution, multivariate normal densities, Chi squared test.
II	Statistical Patten Recognition: Bayesian Decision Theory, Classifiers, Normal density and discriminant functions.
III	Parameter estimation methods: Maximum-Likelihood estimation, Bayesian Parameter estimation, Dimension reduction methods - Principal Component Analysis (PCA), Fisher Linear discriminant analysis, Expectation-maximization (EM), Hidden Markov Models (HMM), Gaussian mixture models.
IV	Nonparametric Techniques: Density Estimation, Parzen Windows, K-Nearest Neighbour Estimation, Nearest Neighbour Rule, Fuzzy classification.
V	Unsupervised Learning and Clustering: Criterion functions for clustering, Clustering Techniques: Iterative square - error partitional clustering – K means, agglomerative hierarchical clustering, Cluster validation.

Text Book (s):

1. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2009.
2. S. Theodoridis and K. Koutroumbas, “Pattern Recognition”, Academic Press, 2008.

INDIAN TRADITIONS, CULTURE AND SOCIETY
(THM601)

COURSE OUTCOMES (COs):	
CO 1:	Understand contemporary issues by exploring historical roots and finding solutions.
CO 2:	Recognize the importance of the environment and contribute to sustainable development
CO 3:	Develop sensitivity towards Indian culture, tradition, and its diverse character.
CO 4:	Learn about holistic lifestyles from Yogic science and Sanskrit literature relevant to modern society.
CO 5:	Gain knowledge of the Indian Knowledge System, modern scientific perspectives, and principles of Yoga and holistic health.
Unit	**Topic**
I	**Society State and Polity in India** Ancient Indian state theories, stages of state formation, kingship, administration, political ideals, societal conditions, Purusārtha, Varnāshrama system, stages of life, marriage, gender representation, and challenges faced by women.
II	**Indian Literature, Culture, Tradition, and Practices** Evolution of scripts, Vedas, Upanishads, Ramayana, Mahabharata, Puranas, Buddhist and Jain literature, Kautilya's Arthashastra, regional literatures in Sanskrit, Telugu, Kannada, Malayalam, and Hindi, Persian and Urdu influences.
III	**Indian Religion, Philosophy, and Practices** Pre-Vedic and Vedic religions, Buddhism, Jainism, Indian philosophy systems, Shankaracharya, philosophical doctrines, Bhakti and Sufi movements, 19th-century socio-religious reforms, modern practices.
IV	**Science, Management and Indian Knowledge System** Ancient Indian contributions to astronomy, chemistry, mathematics, physics, agriculture, medicine, metallurgy, geography, biology, Harappan technologies, water management, textile technology, and trade.
V	**Cultural Heritage and Performing Arts** Ancient Indian architecture, sculptures, seals, coins, pottery, puppetry, dance, music, theatre, painting, martial arts, fairs and festivals, current arts developments, and Indian cinema.

Text Book (s):

1. A. L. Basham, "The Wonder That Was India," Sidgwick and Jackson.
2. C. N. Srinath, "Indian Culture and Civilization," National Book Trust.
3. Sisir Kumar Das, "A History of Indian Literature," Sahitya Akademi.
4. C. Sivaramamurti, "The Art and Architecture of India," National Book Trust.
5. John Keay, "India: A History," HarperCollins.
6. S. Radhakrishnan, "Indian Philosophy: A Historical Analysis," Oxford University Press.
7. R. C. Majumdar, "Cultural Heritage of India," Bharatiya Vidya Bhavan.
8. A. K. Bag, "The History of Indian Science, Technology, and Culture," Abhinav Publications.
9. Amrit Rai (ed.), "The Encyclopaedia of Indian Literature," Sahitya Akademi.
10. James Fergusson, "Indian Architecture: Hindu and Buddhist," Dover Publications.

COMPUTER NETWORKS LAB
(TCS651)

LIST OF THE EXPERIMENTS:

1. To learn basics of the packet tracer simulator tool.
2. Write a program in C to implement bit stuffing and character stuffing.
3. To connect the computers in Local Area Network and to detect collision of packets.
4. To configure DHCP and DNS server for a given network in packet tracer simulator tool.
5. Write a C program to get the MAC or Physical address of the system using ARP (Address Resolution Protocol) and to subnet a given network according to the requirements in packet tracer simulator tool. .
6. To configure router using command line. Also observe the datagram formats in packet tracer simulator tool.
7. To configure NAT for a given network in packet tracer simulator tool.
8. Write a program to implement TCP and UDP Sockets.
9. Write a C program to transmit a character, a string and a file from one computer to another using RS-232 cable and to configure static routing in packet tracer simulator tool.
10. To configure dynamic routing protocols in packet tracer simulator tool.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

SOFTWARE ENGINEERING LAB
(TCS652)

LIST OF EXPERIMENTS

1. Introduction to Software Engineering Processes.
2. Basic steps required to create project and prepare it for data entry (project tasks, sequence the tasks and estimate task duration).
3. Setting up a project and establish the basic constraints that project will use for its calculation. Analyze the project from different views [Gantt Chart, Network Diagram, Flow Charts]
4. Identifying relationship among the different task and subtasks of the Project.
5. Explain how to enter resources and specific information in the Project and resources to specific tasks.
6. Case Study of the Project.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.