NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL



Honors in

COMPUTER SCIENCE AND ENGINEERING

SCHEME OF INSTRUCTION AND SYLLABI

(for students of B.Tech., Computer Science and Engineering)

(Effective from 2021-22)

Department of Computer Science and Engineering

SCHEME OF INSTRUCTION

Honors in Computer Science and Engineering

S. No.	Course Code		L	T	P	Credits	Sem.
1	CS5123	Advanced Compiler Design *	3	0	0	3	V
2	CS5105	Advanced Software Engineering *	3	0	0	3	VI
3	CS5179	Software Defined Networks *	3	0	0	3	VI
4	CS5222	Digital Video Processing	3	0	0	3	VII
5	CS5252	Data Privacy	3	0	0	3	VII
6	CS5120	Big Data	3	0	0	3	VII
7	CS5261	Foundations of Blockchain Technology	3	0	0	3	VIII
8	CS5182	Soft Computing Techniques	3	0	0	3	VIII
9	CS5178	Reinforcement Learning	3	0	0	3	VIII

- * Mandatory courses
- An honors student should complete total six courses (18 credits).

DETAILED SYLLABUS

Course Code: CS5123	ADVANCED COMPILER DESIGN	Credits 3-0-0: 3

Pre-requisites: Language Processors

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand code generation methods
CO2	Apply scalar variable optimizations and procedural optimizations on intermediate code.
CO3	Apply machine level optimizations on the low level intermediate code.
CO4	Perform loop restructuring transformations

Syllabus:

Introduction: Interpreters - Recursive and iterative interpreters

Code generation: Arithmetic statement translation, acyclic graph representation, pattern matching in the acyclic graph, graph rewriting, linearization following the dependencies, code generation for purely register machine, purely stack machine and with memory addressing. Code generation for a basic block.

Simple optimizations : Constant folding, Scalar replacement of aggregates and Algebraic simplifications, Value numbering, Loop invariants identification, loop invariant code motion, partial redundancy elimination, procedure optimizations - call and return optimizations, Code hoisting, Induction variable identification and optimizations, Unnecessary bounds checking elimination

Register Allocation and Code scheduling : Local methods and global methods, representation of the register allocation problem as graph colouring problem, heuristics, register tracking and spilling, Pipeline and Code scheduling effect on performance of the program, Software pipelining, speculative scheduling, boosting

Inter-procedural data flow analysis - static and dynamic, optimizations, Optimizing for memory hierarchies

High performance systems – Scalar, vector, multiprocessor, SIMD, Message Passing Architectures. Sequential and parallel loops. Data dependence, Use-Def chains. Dependence system, GCD test, Banerjee's Inequality, Exact algorithm, exact algorithm, Vectorization, Concurrentization, Array region analysis, Loop restructuring transformations.

Text Books / Reference Books / Online Resources:

1. Steven S. Muchnick, "*Advanced Compiler Design & Implementation*", Morgan Kaufmann, Elsevier Science, 2003.
2. Michael Wolfe, "*High Performance Compilers for Parallel Computing*", Addison Wesley, 1995.

Course Code: CS 5105	ADVANCED SOFTWARE ENGINEERING	Credits 3-0-0: 3

Pre-Requisites: Software Engineering

Course Outcomes:

At the end of the course, the student will be able to

CO1	Apply the Object Oriented Software-Development Process to design software
CO2	Design large-scale, reusable and complex software systems with Design and Architectural patterns
CO3	Develop and apply testing strategies for software applications
CO4	Analyze different Software Reliability parameters using Markovian Models, Finite Failure Category Models and Infinite Failure category Models
CO5	Design and Plan software solutions to security problems using various paradigms

Syllabus:

Introduction and System Engineering: Introduction, Software Process and Methodology, System Engineering. Analysis and Architectural Design: Software Requirement Elicitation, Domain Modeling, Architectural Design. Modeling and Design of Interactive Systems and Other Types of Systems: Deriving Use Cases from Requirements, Actor-System Interaction Modeling, Object Interaction Modeling, Applying Responsibility-Assignment Patterns, Deriving a design class diagram, User Interface Design. Object State modeling of Event-Driven Systems, Activity Modeling for Transformational Systems. Implementation and Quality Assurance: Implementation Considerations, Software Quality Assurance, Software Testing.
Software Reliability Modeling: Markovian Models, Finite Failure Category Models, Infinite Failure Category Models. Comparison of Software Reliability Models.
Project Management and Software Security: Software Project Management, Software Security.

Text Books/Reference Books/Online Resources:

1. Kung, David. Object-oriented software engineering: an agile unified methodology. McGraw-Hill Higher Education, 2013.
2. Gamma, Erich. Design patterns: elements of reusable object-oriented software, Pearson Education India, 1995.
3. M. Xie, Software Reliability Modelling, World Scientific; 1991.
4. John D. Musa, Anthony Iannino, Kazuhira Okumoto, Software Reliability Measurement, Prediction, Application.McGraw-Hill Book Company; 1987.

Course Code: CS 5179	SOFTWARE DEFINED NETWORKS	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Differentiate between traditional networks and software defined networks and understand the key benefits and use cases of SDN.
CO2	Interpret the SDN data plane devices and OpenFlow Protocols
CO3	Implement the operation of SDN control plane with different controllers
CO4	Apply techniques that enable applications to control the underlying network using SDN
CO5	Evaluate Network Functions Virtualization components and their roles in SDN

Syllabus:

Networking Basics: Switching, Addressing, Routing
SDN Background and Motivation: Evolving network requirements-The SDN Approach: Requirements, SDN Architecture, and Characteristics of Software-Defined Networking.
SDN Data plane and OpenFlow: Data plane Functions, Data plane protocols, OpenFlow: Switch-Controller Interaction, Flow Table, Packet Matching, Actions and Packet Forwarding Flow Table Structure, Flow Table Pipeline, The Use of Multiple Tables, Group Table, Extensions and Limitations, Data plane scalability.
SDN Control Plane: SDN Control Plane Architecture: Control Plane Functions, Southbound Interface, Northbound Interface, Routing, Cooperation and Coordination among Controllers, Controller placement problem, SDN controllers: OpenDaylight, Ryu, ONOS, Floodlight, Control plane scalability, fault tolerance.
SDN Application Plane: SDN Application Plane Architecture: Northbound Interface, Network Applications, User Interface- Network Services Abstraction Layer: Abstractions in SDN, Frenetic-Traffic Engineering Measurement and Monitoring, Security, network updates, SDN usecases: Traffic engineering, network management, network virtualization.
Network Functions Virtualization: Background and Motivation for NFV- NFV Principles, High-Level NFV Framework, NFV Benefits and Requirements- SDN vs. NFV, Network Functions, Service Creation and Chaining, NFV Orchestration, VNF deployment, Service function Chain Deployment. NFV Reference Architecture: NFV Management and Orchestration.
Emerging SDN Models: Protocol Models: NETCONF, BGP, MPLS, Controller Models, Application Models: Proactive, Declarative, External, SDN in Datacenters: Multitenancy, Failure Recovery, SDN in Internet eXchange Points (IXPs)

Text Books/Reference Books/Online Resources:

1. Paul Goransson Chuck Black Timothy Culver: Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, 2016.
2. Ken Gray Thomas Nadeau: Network Function Virtualization, Morgan Kaufmann, 2016.
3. Larry Peterson , Carmelo Cascone , Bruce Davie: Software-Defined Networks: A Systems Approach, Systems Approach, 2021

Course Code: CS5222	Digital Video Processing	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Use the suitable image/video acquisition approach, representation and quality assessment for a given scenario.
CO2	Apply image/video processing operations for enhancement and restoration.
CO3	Understand image/video compression standards
CO4	Solve a video indexing, summarization, browsing and retrieval problem
CO5	Implement an approach to solve a video analytics problem

Syllabus:

Introduction – Introduction to digital image and video processing.
Image/Video processing – Intensity transformations, Applications of linear filtering and non-linear filtering. Morphological operations. Colour image processing. Features for Image and Video processing. Geometric transforms and Image registrations, Image segmentation and analysis. Spatiotemporal noise filtering, coding artifact reduction, Blotch detection and removal, video stabilization, and recent image/video operations and features in contemporary literature.
Image Compression – lossless coding, block transform coding, fundamentals of vector quantization, wavelet image compression, JPEG image compression.
Video Compression – basic concepts and techniques of video coding and the H.261 Standard, spatiotemporal subband/Wavelet Video Compression, Object-based video coding, introduction to video coding standards and formats: H.261, MPEG-2/DVB, MPEG-4, H.264/AVC, H.265/HEVC, H.266/VVC.
Image and Video Acquisition – Image scanning, sampling and interpolation, video sampling and interpolation. Recent standards and practices in image and video acquisition.
Video Quality Assessment – Introduction, HVS Modelling based methods, feature based methods, motion modelling-based methods, and approaches based on contemporary research.
Indexing, Summarization, Browsing and Retrieval – Introduction, Image and Video features, Video analysis, video representation, video browsing and video retrieval. Video features in contemporary research.
Video Analytics – Review of video analytics algorithms for: motion and change detection, object detection, object tracking, behaviour analysis, face recognition, Image and Video classification and recent visual analytics approaches in contemporary research.

Text Books/Reference Books/Online Resources:

1. Alan C Bovik, *Handbook of Image and Video Processing*, 2nd Edition, Academic Press, 2005.
2. Alan C. Bovik, *The Essential Guide to Video Processing,* 1st Edition, Academic Press, 2009.
3. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, 4th Edition, Pearson, 2018.
4. Recent articles in Research and Industry.

Course Code: CS5252	Data Privacy	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Define differential privacy
CO2	Design techniques to achieve differential privacy for linear queries.
CO3	Design mechanisms for query release problem using online learning algorithms.
CO4	Analyze computational complexity of differentially private mechanisms

Syllabus:

The Promise of Differential Privacy: Privacy-preserving data analysis; Basic Terms: The model of computation, Towards defining private data analysis, Formalizing differential privacy; Basic Techniques and Composition Theorems: Useful probabilistic tools, Randomized response, The laplace mechanism, The exponential mechanism, Composition theorems, The sparse vector technique; Releasing Linear Queries with Correlated Error: An offline algorithm: SmallDB, An online mechanism: private multiplicative weights; Generalizations: Mechanisms via α-nets, The iterative construction mechanism, Connections; Boosting for Queries: The boosting for queries algorithm, Base synopsis generators; When Worst-Case Sensitivity is Atypical: Subsample and aggregate, Propose-test-Release, Stability and privacy; Lower Bounds and Separation Results: Reconstruction attacks, Lower bounds for differential privacy; Differential Privacy and Computational Complexity: Polynomial time curators, Some hard-to-Syntheticize distributions, Polynomial time adversaries; Differential Privacy and Mechanism Design: Differential privacy as a solution concept, Differential privacy as a tool in mechanism design, Mechanism design for privacy aware agents; Differential Privacy and Machine Learning: The sample complexity of differentially private machine learning, Differentially private online learning, Empirical risk minimization; Additional Models: The local model, Pan-private streaming model, Continual observation, Average case error for query release.

Text Books/Reference Books/Online Resources:

1. C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, now Publishers, 2014.
2. Charu C. Aggarwal, *Privacy-Preserving Data Mining: Models and Algorithms*, 1st Edition, Springer, 2008.
3. Relevant Research Papers

Course Code: CS 5120	Big Data	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze big data challenges in different domains including social media, transportation, finance and medicine
CO2	Explore relational model, SQL and capabilities of emergent systems in terms of scalability and performance
CO3	Apply machine learning algorithms for data analytics
CO4	Analyze the capability of No-SQL systems
CO5	Analyze MAP-REDUCE programming model for better optimization

Syllabus:

Overview of Big Data, Map Reduce basics, Overview of Hadoop, Map Reduce Algorithm Design, Inverted Indexing for Text Retrieval, Graph Algorithms, Data Mining with Big Data, No SQL databases, Stream Computing Challenges, Stages of analytical evolution, Big Data Analytics in Industry Verticals, Data Analytics Lifecycle, Operationalizing Basic Data Analytic Methods Using R, Analytics for Unstructured Data.

Reading:

1. Bill Franks, *Taming The Big Data Tidal Wave*, 1st Edition, Wiley, 2012.
2. Jure Leskovec, Anand Rajaraman, J D Ullman, Mining Massive Datasets.
3. Jimmy Lin and Chris Dyer, Data Intensive Text Processing with Map Reduce, Pre-production manuscript, Downloadable from Internet.
4. Johannes Ledolter, Data Mining and Business Analytics with R, Wiley, 2013.

Course Code: CS5261	Foundations of Blockchain Technology	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand how blockchain work, including private and public platforms
CO2	Understand the technical underpinnings of blockchain technology at sufficient depth to perform analysis.
CO3	Apply various blockchain concepts to analyse examples, proposals, case studies, and preliminary blockchain system design discussions.
CO4	Know and be able to apply the concepts, tools, and frameworks for building blockchain decentralized applications.
CO5	Design secure smart contract applications on blockchain.

Syllabus:

Blockchain technology Introduction, Concepts of Blockchain Systems, Key Problem Challenges and Solutions, Bitcoin Concept- Merkle Tree - hardness of mining - transaction verifiability - anonymity, Attacks on Bitcoin- Double-spend attacks, Selfish mining, Security of Transactions in Bitcoin, Privacy in Bitcoin, Cryptographic Primitives in Blockchain- Cryptosystems in practice, Cryptographic Hash Functions, Digital Signatures-Aggregate Signature, Threshold Signature Blockchain Platforms - Blockchain-Ethereum, Smart Contracts - Attacks on smart contracts, Permissioned Blockchain – Hyperledger, Blockchain Applications & Use Cases, Consensus Protocols- The consensus problem- Byzantine Generals problem, Asynchronous Byzantine Agreement, Consensus mechanisms used in Bitcoin Blockchain, Ethereum Blockchain and Hyperledger Blockchain, Blockchain (BoT)- Advantages of integrating Blockchain to IoT, Trust Building, Cost Reduction, Accelerate Data Exchanges, Scaled Security for IoT,

Text Books/Reference Books/Online Resources:

1. Arvind Narayanan, “Bitcoin and Cryptocurrency Technologies- A Comprehensive Introduction”, Princeton University Press, 2016.
2. William Magnuson, “Blockchain Democracy- Technology, Law and the Rule of the Crowd”, Cambridge University Press, 2020.
3. Pethuru Raj, Kavita Saini, Chellammal Surianarayanan, “Blockchain Technology and Applications”, CRC Press, 2021.
4. Chandramouli Subramanian, “Blockchain Technology”, Universities Press, 2020.
5. Relevant Research Paper and While Papers.

Course Code: CS 5182	SOFT COMPUTING TECHNIQUES	Credits 3-0-0: 3

Pre-Requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understanding of optimizations problems, comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory
CO2	Understand the fundamental theory and concepts of neural networks and Identify different neural network architectures, algorithms, applications and their limitations.
CO3	Apply genetic algorithms and neural networks to solve real world problems
CO4	Apply soft computing techniques to solve engineering and other societal problems

Syllabus:

Overview of course and Basic of Soft Computing, Introduction of Neural Networks, Learning Process and Learning Task, Supervised Learning – Single and Multi – Layer Network, Associative Memory, Self-organizing Maps, Neuro-Dynamics, Hopfield Network, Fuzzy Logic and Systems-Fuzzy Sets and Membership Functions, Operations on Fuzzy Sets, Fuzzification. Fuzzy Numbers- Uncertain Fuzzy Values, Fuzzy Numbers and its L-R representation, Operations on Fuzzy Numbers. Fuzzy Relations, Fuzzy Inference Systems- Architecture of Fuzzy Inference System, Fuzzy Inference Rules and Reasoning, Defuzzification. Applications of Fuzzy Logic, Genetic algorithms and evolutionary computation. Applications of Genetic Algorithms & Hybrid Systems.

Text Books/Reference Books/Online Resources:

1. R.A. Aliev, R.R. Aliev, Soft Computing and Its Applications, World Scientific Publications, 2001.
2. Roger Jang, Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence, PHI, 2008.
3. Simon Haykin, Neural Network: A Comprehensive Foundation, PHI, 1999.
4. Kishan Mehtrotra, S. Ranka, Elements of artificial Neural Networks, Penram International Publishing (India), 2009
5. Timothy Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, McGraw-Hill, 2010.
6. Bart Kosko, Neural Networks and Fuzzy Systems, PHI, 1994.

Course Code: CS 5178	REINFORCEMENT LEARNING	Credits 3-0-0: 3

Pre-Requisites: Machine Learning

Course Outcomes:

At the end of the course, the student will be able to

CO1	Formulate Reinforcement Learning problems
CO2	Apply various Tabular Solution Methods to Markov Reward Process Problems
CO3	Apply various Iterative Solution methods to Markov Decision Process Problems
CO4	Comprehend Function approximation methods

Syllabus:

Introduction: Introduction to Reinforcement Learning (RL) – Difference between RL and Supervised Learning, RL and Unsupervised Learning. Elements of RL, Markov property, Markov chains, Markov reward process (MRP).
Evaluative Feedback - Multi-Arm Bandit Problem: An n-Armed Bandit Problem, Exploration vs Exploitation principles, Action value methods, Incremental Implementation, tracking a non-stationary problem, optimistic initial values, upper-confidence-bound action selection, Gradient Bandits.
Introduction to and proof of Bellman equations for MRPs
Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.
Dynamic Programming (DP): Overview of dynamic programming for MDP, principle of optimality, Policy Evaluation, Policy Improvement, policy iteration, value iteration, asynchronous DP , Generalized Policy Iteration.
Monte Carlo Methods for Prediction and Control: Overview of Monte Carlo methods for model free RL, Monte Carlo Prediction, Monte Carlo estimation of action values, Monto Carlo Control, On policy and off policy learning, Importance sampling.
Temporal Difference Methods: TD Prediction, Optimality of TD(0), TD Control methods - SARSA, Q-Learning and their variants.
Eligibility traces: n-Step TD Prediction, Forward and Backward view of TD(λ), Equivalence of forward and backward view, Sarsa(λ),, Watkins's Q(λ), Off policy eligibility traces using importance of sampling.
Function Approximation Methods: Value prediction with function approximation, gradient descent methods, Linear methods, control with function approximation.

Text Books/Reference Books/Online Resources:

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction", 2nd Edition, The MIT Press.
2. Csaba Szepesvari – Algorithms for Reinforcement Learning – Morgan & Claypool, 2010.