SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING :: TIRUPATI – 517 502

Department of Computer Science and Engineering

Choice Based Credit System – 2020 Regulations

Scheme of Instruction and Examinations

List of subjects for B.Tech (Honors) in Computer Science and Engineering**

(Effective from the batch of students admitted from the academic year 202021)

Course Code	Course Title	Scheme of Instruction (Hours/Week)				No. of Credits
		L	T	P	Total	
CSHN01	Distributed Databases	3	1		4	4
CSHN02	Advanced Operating Systems	3	1		4	4
CSHN03	Multicore Computing	3	1		4	4
CSHN04	Natural Language Processing	3	1		4	4
CSHN05	Software Architecture and Design Patterns	3	1		4	4
CSHN06	Multi Agent Systems	3	1		4	4
CSHN07	Deep Learning	3	1		4	4
CSHN08	Advanced Parallel Processing	3	1		4	4

- All Courses - 40 marks (Internal) + 60 Marks (Univ. Semester End)

** Students shall register for any 4 subjects (4*4 = 16 credits) from the above listed subjects, choosing one subject each in IV, V, VI and VII semester. Further, they shall acquire 4 credits through two MOOCs (each of 2 credits), which shall be discipline-specific.

CSHN 01

SRI VENKATESWARA UNIVERSITY :: TIRUPATI

B.Tech (Honors in CSE) – CBCS Regulations-2020

DISTRIBUTED DATABASES

No.of Credits: 4 Instruction Hours/Week: 3L+1T

Course Objectives:

- To expose the need for distributed database technology to confront with the deficiencies of the centralized database systems.
- To introduce basic principles and implementation techniques of distributed database systems.
- To familiarize students with the principles and knowledge of parallel databases.

UNIT I

Introduction: What Is a Distributed Database System?, History of Distributed DBMS, Data Delivery Alternatives, Promises of Distributed DBMSs, Design Issues, Distributed DBMS Architectures.

Distributed and Parallel Database Design: Data Fragmentation, Allocation, Combined Approaches, Adaptive Approaches, Data Directory.

Distributed Data Control: View Management, Access Control, Semantic Integrity Control.

UNIT II

Distributed Query Processing: Overview, Data Localization, Join Ordering in Distributed Queries, Distributed Cost Model, Distributed Query Optimization, Adaptive Query Processing.

Distributed Transaction Processing: Background and Terminology, Distributed Concurrency Control, Distributed Concurrency Control Using Snapshot Isolation, Distributed DBMS Reliability, Modern Approaches to Scaling Out Transaction Management.

UNIT III

Data Replication: Consistency of Replicated Databases, Update Management Strategies, Replication Protocols, Group Communication, Replication and Failures.

Database Integration - Multidatabase Systems: Database Integration, Multidatabase Query Processing.

Parallel Database Systems: Objectives, Parallel Architectures, Data Placement, Parallel Query Processing, Load Balancing, Fault-Tolerance, Database Clusters.

UNIT IV

Peer-to-Peer Data Management: Infrastructure, Schema Mapping in P2P Systems, Querying Over P2P Systems, Replica Consistency, Blockchain.

Big Data Processing: Distributed Storage Systems, Big Data Processing Frameworks, Stream Data Management, Graph Analytics Platforms, Data Lakes.

UNIT V

NoSQL, NewSQL, and Polystores: Motivations for NoSQL, Key-Value Stores, Document Stores, Wide Column Stores, Graph DBMSs, Hybrid Data Stores, Polystores.

Web Data Management: Web Graph Management, Web Search, Web Querying, Question Answering Systems, Searching and Querying the Hidden Web, Web Data Integration.

Course Outcomes:

After completion of the course the students will be able to

- Design and implement distributed databases.
- Handle query processing in a distributed database system.
- Comprehend transaction management and analyze various approaches to concurrency control in distributed databases.
- Design and implement various algorithms and techniques for deadlock and recovery in distributed databases.

Text Books:

1. M. Tamer Ozsu and Patrick Valduriez, "Principles of Distributed Database Systems", Fourth Edition, Springer, 2020.

Reference Books:

1. Stefano Ceri and Giuseppe Pelagatti, Distributed Databases: Principles and Systems, McGraw Hill Education, 2017.
2. Saeed K. Rahimi and Frank S. Haug, Distributed Database Management Systems: A Practical Approach, Wiley.
3. Chhanda Ray, Distributed Database Systems, First Edition, Pearson Education India.
4. Sachin Deshpande, Distributed Databases, Dreamtech Press.
5. David Bell and Jane Grimson, Distributed Database Systems, First Edition, Addison-Wesley, 1992.
6. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: Database Systems: The Complete Book, Second Edition, Pearson Education.

CSHN 02

SRI VENKATESWARA UNIVERSITY :: TIRUPATI

B.Tech (Honors in CSE) – CBCS Regulations-2020

ADVANCED OPERATING SYSTEMS

No.of Credits: 4 Instruction Hours/Week: 3L+1T

Course Objectives:

- Familiarize the students with the basics of hardware and software issues in distributed computing systems.
- Understand issues related to Synchronization, mutual exclusion, naming, synchronization, consistency and replication,
- Introduce the concepts of distributed file systems, distributed shared virtual memory fault tolerance and deadlocks in distributed systems.

UNIT I

Distributed systems – Introduction, Hardware concepts, Software concepts and Design issues. Layered protocols, Asynchronous transfer mode networks, Client server model, Remote procedure call, Group communication.

UNIT II

Clock synchronization, Mutual exclusion, Election algorithms, Atomic transactions, Deadlocks in distributed systems.
Threads, System models, Processor allocation, Scheduling in distributed systems.

UNIT III

Fault tolerance, Real-time distributed systems, Distributed file systems – Design, Implementation and Trends.
Distributed shared memory – Introduction and shared memory concept.

UNIT IV

DSM Consistency models, Page-based distributed shared memory.
Case study Amoeba – Introduction, Objects and capabilities, Process management, Memory management, Communication and Servers.

UNIT V

Multimedia operating systems – Introduction, Multimedia files, Video compression, Audio compression, Multimedia process scheduling, Multimedia file system paradigms, File placement, Caching, Disk scheduling for multimedia.

Text Books:

1. Tanenbaum A S, Distributed Operating Systems, Pearson Education, 2005.
2. Tanenbaum A S, Modern Operating Systems, 3rd Edition, Pearson Education, 2008.

Reference Books:

1. Andrew S Tanenbaum Maarten Van Steen, Distributed Systems: Principles and Paradigms, Pearson, 2006.

Course Outcomes:

After completion of the course the students will be able to

- Understand the design principles in distributed operating systems and the architectures for distributed systems
- Implement clock synchronization, concurrency control, deadlock detection, load balancing related to distributed operating systems.
- Efficiently handle the issues of Mutual Exclusion and Deadlock detection and recovery in distributed operating systems.
- Design and implement distributed file systems and Distributed shared virtual memory.

CSHN 04

SRI VENKATESWARA UNIVERSITY :: TIRUPATI

B.Tech (Honors in CSE) – CBCS Regulations-2020

NATURAL LANGUAGE PROCESSING

No.of Credits: 4 Instruction Hours/Week: 3L+1T

Course Objectives:

The course is designed to

- Understand computational properties of natural languages and the commonly used algorithms for processing linguistic information.
- Learn basics of semantic analysis and discourse analysis and drives it to machine translation
- Learn how to process written text from basic of fundamental knowledge of Finite automata, Regular expression and probabilistic model with n-grams.

.

UNIT I

Introduction to SLP, Regular Expressions and Automata, Words and Transducers, N-grams, Part-of-Speech Tagging, Hidden Markov and Maximum Entropy Models.

UNIT II

Phonetics, Speech Synthesis, Automatic Speech Recognition, Speech Recognition: Advanced Topics, Computational Phonology.

UNIT III

Formal Grammars of English, Syntactic Parsing, Statistical Parsing, Features and Unification, Language and Complexity.

UNIT IV

Semantics and Pragmatics, The Representation of Meaning, Computational Semantics, Lexical Semantics, Computational Lexical Semantics, Computational Discourse.

UNIT V

Information Extraction, Question Answering and Summarization, Dialogue and Conversational Agents, Machine Translation.

Course Outcomes:

After completion of the course the students will be able to

- Comprehend and analyze the various elements of speech processing
- Design and develop machine learning techniques in the area of NLP
- Demonstrate the state-of-the-art algorithms and techniques for text-based processing
- Perform POS tagging for a given natural language

.

Text Books:

1. Jurafsky D, Martin J H, *Speech and Language Processing: An Introduction to Natural Language Processing*, Computational Linguistics and Speech Recognition, 2nd edition, Pearson Education, 2013.

Reference Books:

1. Chopra D, Joshi N, Mastering Natural Language Processing with Python, Packt, 2016.
2. Watanabe S, Jen-Tzung Chien J T, Bayesian Speech and Language Processing, Cambridge University Press, 2015.
3. Kurdi M Z, Natural Language Processing and Computational Linguistics: Speech, Morphology and Syntax, Wiley, 2016.
4. Tiwary U S, Siddiqui T, Natural Language Processing & Information Retrieval, Oxford University Press, 2008.
5. Bikel D M, Zitouni I, Multilingual Natural Language Processing Applications: From Theory to Practice, Pearson Education, 2012.
6. Bharati A, Chaitanya V, Sangal R, Natural Language Processing: A Paninian Perspective, PHI, 1995.
7. Steven B, Natural Language Processing with Python, Shroff, 2009.
8. Mariani J, Language and Speech Processing, Wiley, 2013.

CSHN 05

SRI VENKATESWARA UNIVERSITY :: TIRUPATI
B.Tech (Honors in CSE) – CBCS Regulations-2020

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

No.of Credits: 4 Instruction Hours/Week: 3L+1T

Course Objectives:
The course is designed to

- Impart knowledge on the different architectural styles and architectural patterns for the software
- Understand design patterns and their underlying object oriented concepts
- Focus on the architectural design issues as well as the use of design patterns for different applications

UNIT I
Design Patterns: Origin and History, Architectural to Software Design Patterns, What Is a Design Pattern?, More about Design Patterns.
UML: A Quick Reference - Structure Diagrams, Behavior Diagrams, Model Management Diagram, Class Diagrams, Sequence Diagrams.
Basic Patterns: Interface, Abstract Parent Class, Private Methods, Accessor Methods, Constant Data Manager, Immutable Object, Monitor.

UNIT II
Creational Patterns: Factory Method, Singleton, Abstract Factory, Prototype, Builder.
Collectional Patterns: Composite, Iterator, Flyweight, Visitor.

UNIT III
Structural Patterns: Decorator, Adapter, Chain of Responsibility, Façade, Proxy, Bridge, Virtual Proxy, Counting Proxy, Aggregate Enforcer, Explicit Object Release, Object Cache.

UNIT IV
Behavioral Patterns: Command, Mediator, Memento, Observer, Interpreter, State, Strategy, Null Object, Template Method, Object Authenticator, Common Attribute Registry.

UNIT V
Concurrency Patterns: Critical Section, Consistent Lock Order, Guarded Suspension, Read-Write Lock.
Case Study: A Web Hosting Company – Objective, KPS Hosting Solutions: A Brief Overview, Requirements, Business Objects and Their Association, Framework for Application Processing, Conclusion.

Text Books:

1. Partha Kuchana, Software Architecture Design Patterns in Java, Auerbach Publications – A CRC Press Company, 2004.

Reference Books:

1. Mark Richards, Neal Ford, Fundamentals of Software Architecture: An Engineering Approach, O’Reilly, First Edition, 2020.
2. Kamalmeet Singh, Adrian Ianculescu, Lucian-Paul Torje, Design Patterns and Best Practices in Java: A Comprehensive Guide to Building Smart and Reusable Code in Java, Packt Publishing, First Edition, 2018.
3. Erich Gamma, Design Patterns, Pearson Education.
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley/Pearson Education, 1995.
5. Steven John Metsker, William C. Wake, Design Patterns in Java, Addison-Wesley/ Pearson Education, 2006.
6. Alan Shalloway, James R. Trott, Design Patterns Explained: A New Perspective on Object-Oriented Design, Pearson Education, 2004.
7. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Pattern-Oriented Software Architecture: A System of Patterns, John Wiley & Sons, 1996.
8. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, Addison-Wesley Professional, Third Edition, 2012.

Course Outcomes:

After completion of the course the students will be able to

- Apply his knowledge to create an architecture for given application.
- Explain the role of analyzing architectures.
- Able to identify different structural patterns.
- Design creational, structural and behavioral patterns.
- Recognize major software architectural styles, design patterns, and frameworks.

CSHN 07

SRI VENKATESWARA UNIVERSITY :: TIRUPATI
B.Tech (Honors in CSE) – CBCS Regulations-2020

DEEP LEARNING

No.of Credits: 4 | Instruction Hours/Week: 3L+1T

Course Objectives:
The course is designed to

- Learn some advanced topics such as recurrent neural networks, long short term memory cells and convolutional neural networks.
- Learn deep recurrent and memory networks and deep Turing machines
- Understand different types of deep learning network models.

UNIT I
Introduction to Machine Learning: Learning Algorithms, Capacity, Overfitting and Underfitting, Hyperparameters and Validation Sets, Estimators, Bias and Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning Algorithms, Unsupervised Learning Algorithms, Stochastic Gradient Descent, Building a Machine Learning Algorithm, Challenges Motivating Deep Learning.

UNIT II
Review of fundamental learning techniques, Feed forward neural network - Artificial Neural Network, activation function, multi-layer neural network, Training Neural Network: Risk minimization, loss function, back propagation, regularization, model selection, and optimization. Conditional Random Fields: Linear chain, partition function, Markov network, Belief propagation, Training CRFs, Hidden Markov Model, Entropy.

UNIT III
Deep Feedforward Networks: Example: Learning XOR, Gradient-Based Learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms, Historical Notes.
Regularization for Deep Learning: Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise Robustness, Semi-Supervised Learning, Multi-Task Learning, Early Stopping, Parameter Tying and Parameter Sharing, Sparse Representations, Bagging and Other Ensemble Methods, Dropout, Adversarial Training, Tangent Distance, Tangent Prop, and Manifold Tangent Classifier.

UNIT IV
Optimization for Training Deep Models: How Learning Differs from Pure Optimization, Challenges in Neural Network Optimization, Basic Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates, Approximate Second-Order Methods, Optimization Strategies and Meta-Algorithms.
Convolutional Networks: The Convolution Operation, Motivation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Variants of the Basic Convolution Function, Structured Outputs, Data Types, Efficient Convolution Algorithms, Random or Unsupervised Features,

The Neuroscientific Basis for Convolutional Networks, Convolutional Networks and the History of Deep Learning.

UNIT V

Sequence Modeling: Recurrent and Recursive Nets, Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks, The Challenge of Long-Term Dependencies, Echo State Networks, Leaky Units and Other Strategies for Multiple Time Scales, The Long Short-Term Memory and Other Gated RNNs, Optimization for Long-Term Dependencies, Explicit Memory.
Practical Methodology: Performance Metrics, Default Baseline Models, Determining Whether to Gather More Data, Selecting Hyperparameters, Debugging Strategies, Example: Multi-Digit Number Recognition.
Applications: Large-Scale Deep Learning, Computer Vision, Speech Recognition, Natural Language Processing, Other Applications.

Course Outcomes:

After completion of the course the students will be able to

- Explain different types of deep learning network models
- Apply optimization techniques to improve the performance of deep neural networks.
- Implement tools on Deep Learning techniques.

Text Books:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.

Reference Books:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, Artificial Neural Networks, B., PHI Learning Pvt. Ltd, 2009.
3. Raúl Rojas, Neural Networks: A Systematic Introduction, Springer, 1996.
4. Golub, G.,H., and Van Loan,C.,F, Matrix Computations, JHU Press,2013.
5. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
6. M Tim Jones, Artificial Intelligence - A Systems Approach, Infinity Science Press, 2008.
7. Russel S, Norvig P, Artificial Intelligence: A Modern Approach, 3rd edition, Pearson Education, 2010.

CSHN 08

SRI VENKATESWARA UNIVERSITY :: TIRUPATI

B.Tech (Honors in CSE) – CBCS Regulations-2020

ADVANCED PARALLEL PROCESSING

No.of Credits: 4 Instruction Hours/Week: 3L+1T

Course Objectives:

The course is designed to

- Understand engineering trade-offs involved in designing modern parallel computers
- Develop programming skills to effectively implement parallel architecture

UNIT I

Introduction: Heterogeneous Parallel Computing, Architecture of a Modern GPU, Why More Speed or Parallelism?, Speeding Up Real Applications, Parallel Programming Languages and Models, Overarching Goals.

History of GPU Computing: Evolution of Graphics Pipelines, GPGPU: An Intermediate Step, GPU Computing.

Introduction to Data Parallelism and CUDA C: Data Parallelism, CUDA Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading.

UNIT II

Data-Parallel Execution Model: Cuda Thread Organization, Mapping Threads to Multidimensional Data, Matrix-Matrix Multiplication - A More Complex Kernel, Synchronization and Transparent Scalability, Assigning Resources to Blocks, Querying Device Properties, Thread Scheduling and Latency Tolerance.

CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Strategy for Reducing Global Memory Traffic, A Tiled Matrix-Matrix Multiplication Kernel, Memory as a Limiting Factor to Parallelism.

Performance Considerations: Warps and Thread Execution, Global Memory Bandwidth, Dynamic Partitioning of Execution Resources, Instruction Mix and Thread Granularity.

UNIT III

Parallel Patterns: Convolution – Background, 1D Parallel Convolution - A Basic Algorithm, Constant Memory and Caching, Tiled 1D Convolution with Halo Elements, A Simpler Tiled 1D Convolution - General Caching.

Parallel Patterns: Prefix Sum – Background, A Simple Parallel Scan, Work Efficiency Considerations, A Work-Efficient Parallel Scan, Parallel Scan for Arbitrary-Length Inputs.

Parallel Patterns: Sparse Matrix-Vector Multiplication – Background, Parallel SpMV Using CSR, Padding and Transposition, Using Hybrid to Control Padding, Sorting and Partitioning for Regularization.

UNIT IV

Application Case Study: Molecular Visualization and Analysis – Application Background, A Simple Kernel Implementation, Thread Granularity Adjustment, Memory Coalescing.

Parallel Programming and Computational Thinking: Goals of Parallel Computing, Problem Decomposition, Algorithm Selection, Computational Thinking.

An Introduction to OpenCL™: Background, Data Parallelism Model, Device Architecture, Kernel Functions, Device Management and Kernel Launch, Electrostatic Potential Map in OpenCL.
Parallel Programming with OpenACC: OpenACC versus CUDA C, Execution Model, Memory Model, Basic OpenACC Programs, Future Directions of OpenACC.

UNIT V
Thrust: A Productivity-Oriented Library for CUDA – Background, Motivation, Basic Thrust Features, Generic Programming, Benefits of Abstraction, Programmer Productivity, Best Practices.
Programming a Heterogeneous Computing Cluster: Background, A Running Example, MPI Basics, MPI Point-to-Point Communication Types, Overlapping Computation and Communication, MPI Collective Communication.
CUDA Dynamic Parallelism: Background, Dynamic Parallelism Overview, Important Details, Memory Visibility, A Simple Example, Runtime Limitations, A more Complex Example.

Course Outcomes:
After completion of the course the students will be able to

- Design and implement parallel programs in environments such as CUDA, OpenMP
- Generate parallel programs for matrix, graph and sorting problems using CUDA library
- Learn about different OpenMP programming, MPI programming,
- Develop parallel programs using OpenCL library
- Analyze the functionality of GPU architecture using parallel computing platform CUDA.

Text Books:
1. David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann/ Elsevier, Second Edition, 2013.

Reference Books:
1. Peter Pacheco, Matthew Malensek, An Introduction to Parallel Programming, Morgan Kaufmann/ Elsevier, Second Edition, 2022.
2. Roman Trobec, Boštjan Slivnik, Patricio Bulić, Borut Robič, Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms, Springer International Publishing, 2018.
3. Bertil Schmidt, Jorge Gonzalez- Dominguez, Christian Hundt, Moritz Schlarb, Parallel Programming: Concepts and Practice, Morgan Kaufmann/ Elsevier, 2017.
4. Gerassimos Barlas, Multicore and GPU Programming: An Integrated Approach, Elsevier Science, 2014.
5. Nicholas Wilt, The CUDA Handbook: A Comprehensive Guide to GPU Programming, Addison-Wesley/ Pearson Education, 2013.
6. Rob Farber, CUDA Application Design and Development, Morgan Kaufmann/ Elsevier, 2011.

7. Geoffrey C. Fox, Roy D. Williams, Guiseppe C. Messina, Parallel Computing Works!, Elsevier Science, 2014.
8. Janusz Kowalik, Tadeusz Puźniakowsk, Using OpenCL: Programming Massively Parallel Computers, IOS Press, 2012.
9. Wen-mei W. Hwu, David B. Kirk and Izzat El Hajj, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, Fourth Edition, 2022.
10. S K Basu, Parallel and Distributed Computing: Architectures and Algorithms, Prentice Hall, 2016.
11. Kai Hwang, Jack J. Dongarra, Geoffrey C. Fox, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann/ Elsevier, 2013.