Master of Technology

In

Computer Science and Engineering

Course Structure & Syllabus

Department of Computer Science and Engineering

National Institute of Technology Hamirpur

Hamirpur (HP) – 177005, India

Course Structure of M. Tech. Computer Science and Engineering

SEMESTER-I

Sr. No	Course No.	Course Name	Teaching Schedule			Hours/ Week	Credit
			L	T	P		
1	CS-611	Topics in Computer Networks	4	0	0	4	4
2	CS-612	Topics in Software Engineering	4	0	0	4	4
3	CS-613	Architecture of Large Systems	4	0	0	4	4
4	CS-7MN	Programme Elective-I	4	0	0	4	4
5	CS-7MN	Programme Elective-II	4	0	0	4	4
6	CS-614	Computational Lab–I	0	0	4	4	2
	Total		**20**	**0**	**4**	**24**	**22**

Programme Elective-I & II: List of Programme Electives is given in the Annexure.

SEMESTER-II

Sr. No	Course No.	Course Name	Teaching Schedule			Hours/ week	Credit
			L	T	P		
1	CS-621	Advanced Algorithms	4	0	0	4	4
2	CS-622	Advanced Operating Systems	4	0	0	4	4
3	CS-623	Advanced DBMS	4	0	0	4	4
4	CS-7MN	Programme Elective-III	4	0	0	4	4
5	CS-7MN	Programme Elective-IV	4	0	0	4	4
6	CS-624	Computational Lab–II	0	0	4	4	2
	Total		**20**	**0**	**4**	**24**	**22**

Programme Elective –III & IV: List of Programme Electives is given in the Annexure.

SEMESTER-III

Sr. No	Course No.	Course Name	Hours/week	Credit
1	CS-800	M.Tech. Dissertation	--	20
	Total			**20**

SEMESTER-IV

Sr. No	Course No.	Course Name	Hours/week	Credit
1	CS-800	M. Tech Dissertation	--	20
	Total		--	**20**

Total Credits of the Programme = 84

Annexure
List of Programme Electives

Programme Elective-I

CS-711	Artificial Intelligence
CS-712	Soft Computing
CS-713	Speech and Natural Language Processing
CS-714	Data Mining
CS-715	Bioinformatics

Programme Elective-II

CS-721	Formal Languages and Automata Theory
CS-722	Parallel Algorithms
CS-723	Information Theory and Coding
CS-724	Game Theory
CS-725	Combinatorial Optimization

Programme Elective-III

CS-731	Computer Vision and Image Processing
CS-732	Performance Evaluation of Computer System
CS-733	Fault Tolerant Computing
CS-734	Embedded Systems

Programme Elective-IV

CS-741	Cryptography and Computer Security
CS-742	Intrusion Detection System
CS-743	Biometric Security
CS-744	Cluster and Grid Computing

Course Name: **Topics in Computer Networks** Course Code: **CS-611** Course Type: **Core**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To impart knowledge about the principles of internetworking.
- To introduce the fundamental concepts relevant to design issues of network layers, various wireless networks concepts, etc.
- To enable the students to understand the basic principles of mobility in wireless networks.

Course Content

Overview and motivation, Resource sharing, Design issues for the network layers Networking devices. Principles of internetworking, Tunneling, Fragmentation, Naming and addressing concepts, Hierarchical naming, Domain name system, Name resolution process, IP address classes and concept of sub netting, Classless Inter-domain routing (CIDR) and DHCP concepts, The internet protocols: IP, ICMP, ARP and RARP. The design issues for the transport layer, addressing, establishing connection, flow control and multiplexing. The internet protocols: TCP and UDP, Multicast routing, Mobility in networks, Mobile IP, Emerging trends in networking.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand network models and architectures.

CO2: Analyze the performance of various MAC, routing, and transport protocols, and design of new protocol.

CO3: Solve basic network design problems using knowledge of wired and wireless networks.

CO4: Apply knowledge of networking technologies to design a network as per the organization requirements.

Books and References

1. Data Communications and Networking by B.A. Forouzan, McGraw Hill.
2. Computer Networks by A.S. Tanenbaum, PHI.
3. Understanding TCP/IP by Libor D. and Alena K, PACKT Publishing.
4. Introduction to Wireless and Mobile Systems by Dharma P. Agrawal and Q-An Zeng, Thomson Learning Inc.

Course Name: **Topics in Software Engineering**
Course Code: **CS-612**
Course Type: **Core**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the principles of software engineering.
- To introduce the fundamental concepts relevant to software project management.
- To enable the students to understand the basic software models.

Course Content

State-of-the-art in various areas of Software Engineering, Software project management, Metrics and measurement, Software configuration management, Software risk management, Requirements engineering, Software quality assurance, Software reliability models, Object oriented design, Formal specifications, Formal verification of programs, Jackson method for design, CASE tools and technology, UML, eXtreme Programming, Aspect Oriented Programming, Secure software engineering principles.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Effectively apply software engineering practice over the entire system lifecycle. This includes requirements engineering, analysis, prototyping, design, implementation, testing, maintenance activities and management of risks involved in software and embedded systems.

CO2: Know classical and evolving software engineering methods, can select and tailor appropriate methods for projects, and can apply them as both team members and managers to achieve project goals.

CO3: Prepare and publish the necessary documents required throughout the project lifecycle.

Books and References

1. An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publishing House.
2. Software Engineering by S. L. Pfleeger, MacMillan Publishing Company.
3. Software Engineering: A Practitioner's Approach by Roger Pressman, McGraw-Hill Publishing.

Course Name: **Architecture of Large Systems** Course Code: **CS-613** Course Type: **Core**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To impart knowledge about examining the qualitative and quantitative computer design tradeoffs.
- To introduce the fundamental concepts relevant to art of selecting and interconnecting hardware components to create a computer that meets functional, performance and cost goals.
- To enable the students to understand the basic non-classical architectures such as parallel processors, multi-core chips, pipelined and VLIW machines.

Course Content

Introduction: review of basic computer architecture, quantitative techniques in computer design, measuring and reporting performance, CISC and RISC processors. Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards, and structural hazards, techniques for handling hazards, Exception handling, Pipeline optimization techniques. Compiler techniques for improving performance. Hierarchical memory technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies. Instruction-level parallelism: basic concepts, techniques for increasing ILP, superscalar, super pipelined and VLIW processor architectures. Array and vector processors. Multiprocessor architecture: taxonomy of parallel architectures. Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks. Distributed shared-memory architecture. Cluster computers. Non von Neumann architectures: data flow computers, reduction computer architectures, systolic architectures.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand that how hardware and software (especially the operating system and compilers) must work synergistically together to provide optimum throughput.

CO2: Understand the elements of modern computer along with measures of performance.

CO3: Understand the concepts of pipelining, multiprocessors, parallel processors, etc.

Books and References

1. Computer Organization and Design: A Hardware/Software Interface by David Patterson and John Henessey, Morgan Kaufmann publication.
2. Computer Architecture: A Quantitative Approach by John Henessey & David Patterson, Morgan Kaufmann publication.

Course Name: **Computational Lab–I**
Course Code: **CS-614**

Contact Hours/Week: **2P** Course Credits: **01**

Course Objectives

- To provide skills for designing and analyzing algorithms.
- To enable students to work on various simulators.
- To provide skills to work towards solution of real life problems.

List of Experiments

1. Installation and working on various simulators viz. ETHEREAL, OMNET++, NS2,NS3, MATLAB, etc.
2. Simulation of routing protocols for wired and wireless networks.
3. Simulation of MAC protocols for wired and wireless LAN.
4. Implementation of searching techniques over big data.
5. Implementation of various encryption and decryption algorithms and comparing their performance.
6. Content based searching in image data.
7. Application development for any management information system.
8. Working on parallel algorithms.
9. Creation of GUI.

Note: *The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Elicit, analyze and specify software requirements.
CO2: Simulate given problem scenario and analyze its performance.
CO3: Develop programming solutions for given problem scenario.

Course Name: **Advanced Algorithms** Course Code: **CS-621** Course Type: **Core**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To impart knowledge about the various approaches to design an algorithm.
- To introduce the fundamental concepts relevant to understand the concepts of time and space complexity, worst case, average case and best case complexities.
- To enable the students to understand the basics of algorithms.

Course Content

Introduction: Algorithm Design paradigms- motivation, concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations; Divide and Conquer approach, examples of some sorting techniques; Greedy Algorithms; Graph Algorithms: Representation of graphs, BFS, DFS, single source shortest path, all pair shortest path; Dynamic programming: Overview, difference between dynamic programming and divide and conquer, Traveling salesman Problem, longest Common sequence, 0/1 knapsack., Backtracking: 8-Queen Problem, Sum of subsets, graph coloring, Hamiltonian cycles.

Branch and bound: its application for some of above problems; Computational Complexity: Complexity measures, Polynomial vs non-polynomial time complexity; NP-hard and NP-complete classes and examples.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand asymptotic notations to analyse the performance of algorithms.

CO2: Understand and apply various problem solving techniques such as divide and conquer, greedy algorithm, dynamic programming, etc.

CO3: Solve given problem by selecting the appropriate algorithm design technique and justify the selection.

CO4: Know the concepts of P, NP, NP-hard and NP-complete problems.

Books and References

1. Introduction to Algorithms by T. Cormen, C. Leiserson, R. Rivest, and C. Stein, MIT Press / McGraw-Hill.
2. Algorithm Design: Foundations, Analysis, and Internet Examples by Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons.
3. Algorithm Design by J. Kleinberg and É. Tardos, Addison-Wesley.

Course Name: **Advanced Operating Systems** Course Code: **CS-622** Course Type: **Core**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To learn the advancement in Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication.
- To learn the mechanisms involved in memory management in contemporary OS.
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
- To know the components and management aspects of concurrency management.
- To learn programmatically to implement simple OS mechanisms.

Course Content

Overview of Advanced Operating Systems: Introduction, Functions of OS, Evolution, Types of advance operating systems, Issues and challenges in the design of distributed Operating Systems. Models of Distributed Operating Systems: Introduction, Architectural models, Fundamental models. Inter-process Communication: APIs for Internet Protocols, External Data Representations, Client-Server Communication, Group Communication. Distributed File Systems: Introduction, Architecture, Design Issues, Case Studies: Sun Network File System, Andrew File System.

Time and Global State: Physical and Logical Time, Internal and External Synchronization protocols viz Cristian's Algorithm, Berkeley Algorithm, Network Time Protocol, Lamport's Logical Clocks, Vector Clocks, Global State, Cuts, and Distributed Debugging. Distributed Mutual Exclusion: Simple and Multicast based Mutual Exclusion Algorithms viz Centralized, Ring based, Ricart Agrawala's Algorithm, Maekawa's Algorithm. Distributed Election Algorithms: Ring based and Bully's Algorithm. Multicast Communication: Basic and Reliable Multicast and Ordered Multicast communication.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Outline the potential benefits of distributed systems.

CO2: Apply standard design principles in the construction of distributed systems.

CO3: Select appropriate approaches for building a range of distributed systems.

Books and References

1. Distributed Systems: Concepts and Design by G. Coulouris, J. Dollimore, and T. Kindberg, Pearson Education.
2. Advanced Concepts in Operating Systems: Distributed, Database and Multiprocessor Operating Systems by M. Singhal and N. Shivaratri, McGraw Hill International Edition.
3. Distributed Operating Systems by R. K. Sinha, Prentice Hall.

Course Name: **Advanced DBMS** Course Code: **CS-623** Course Type: **Core**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- List and explain the fundamental concepts of a database system.
- Utilize a wide range of features available in a DBMS package.
- Analyze database requirements and determine the entities involved in the system and their relationship to one another.
- Develop the logical design of the database using data modeling concepts such as entity-relationship diagrams.
- Create a relational database using a relational database package.
- Manipulate a database using SQL.
- Assess the quality and ease of use of data modeling and diagramming tools.

Course Content

Database Management systems concepts, Conceptual Database Design, Logical Database Design Physical Database Design, Query Processing, Transaction processing, Crash recovery, Concurrency control, Distributed Database, client/server database, Integrity security and repositories. Emerging Database trends, Design and database administration skills based on near-real life applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Master the basic concepts and appreciate the applications of database systems.

CO2: Master the basics of SQL and construct queries using SQL.

CO3: Be familiar with a commercial relational database system (Oracle) by writing SQL using the system.

CO4: Be familiar with the relational database theory, and be able to write relational algebra expressions for queries.

Books and References

1. Fundamentals of Database Systems by R. Elmasri and S. Navathe, Benjamin Cummings.
2. An introduction to Data Base systems by C. J. Date, Addison Wesley.
3. Database System Concepts by Silberschatz, Korth and Sudarshan, Tata McGraw Hill.
4. Client/Server Strategies by Vaakevitch David, Galgotia Publications.

Course Name: **Computational Lab–II** Course Code: **CS-624**	
Contact Hours/Week: **2P**	Course Credits: **01**

Course Objectives

- To provide exposure of working on hardware software platforms.
- To design and implement solutions for real life problems.

List of Experiments

1. Hands-on experience on Bluetooth, GSM, GPS, Wifi, WAP and Zigbee kits.
2. Hands-on experience with WSN Memsic/Crossbow Classroom Kit for study of various parameters/protocols viz. effect of transmission range, number of nodes, routing, etc.
3. Implementation of ad hoc networks.
4. Implementation/simulation of time synchronization protocols in wired, wireless and heterogeneous environments.
5. Development of mobile applications for collection of users data based on Bluetooth, Wifi and other traffics.
6. Knowledge extraction from given data.
7. Use of machine learning techniques for solving problems related to wired/wireless networks.
8. Comparative evaluation of intrusion detection techniques.

Note: *The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Simulate/implement given problem scenario and analyze its performance.

CO2: Design solutions for real life problems.

Course Name: **Artificial Intelligence** Course Code: **CS-711** Course Type: **Programme Elective-I**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To provide a strong foundation of fundamental concepts in Artificial Intelligence.
- To provide a basic exposition to the goals and methods of Artificial Intelligence.
- To enable the student to apply these techniques in applications which involve perception, reasoning and learning.

Course Content

Introduction: Introduction to AI, AI techniques, level of model, criteria for success. Function and Recursion, Tuples, patterns, Lists, concrete data types, Inductive definitions, Induction on concrete data, Formal syntax, Operational semantics, operational reasoning, references, the environmental model, exceptions, polymorphism, typing subtyping, inference. AI application development using SML. The major methods of representing knowledge in AI: Rule based representations, declarative or logical formalisms, Prolog programming, frames or objects oriented systems, network based approaches, semantic network, and finally mixed representations. For each of them the emphasis will be along three dimensions: The building of tools in which all knowledge is to be represented. The use of these tools to actually represent knowledge, and the use of this represented knowledge to solve problems (i.e., essentially control issues). Case study of one or more examples from natural language processing, question answering, speech, expert systems, etc.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the various searching techniques and constraint satisfaction problem.

CO2: Apply these techniques in applications which involve perception, reasoning and learning.

CO3: Acquire the knowledge of real world Knowledge representation.

CO4: Use different machine learning techniques to design AI machine and enveloping applications for real world problems.

Books and References

1. ML for the Working Programmer by Larry Paulson, Cambridge University Press.
2. The SML Basis Library, including information on useful common functions included with most implementations of SML.
3. A Gentle Introduction to ML by Andrew Cumming, Napier University, Edinburgh.
4. Tips for Computer Scientists on Standard ML by Mads Tofte, Department of CSE, Copenhagen University.
5. Art of Prolog by Ehud Shapiro, MIT Press.
6. Artificial Intelligence by Ritche and Knight, Tata McGraw-Hill.

Course Name: **Soft Computing**
Course Code: **CS-712**
Course Type: **Programme Elective-I**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To impart knowledge about the basic principles, techniques, and applications of soft computing.
- Provide the mathematical background for carrying out the optimization associated with neural network learning.
- Develop the skills to gain basic understanding of the areas of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.

Course Content

Introduction to Neuro fuzzy and Soft Computing, Fuzzy set theory, Fuzzy Rules, Fuzzy Reasoning, Fuzzy inference System, Neural Networks; Radial basis and recurrent neural networks, Hopfield Networks, Comparison of RBF and MLP Network, Running Algorithms, Neuro Fuzzy Modeling, Applications of Soft Computing to Signal Processing, Image Processing, Forecasting, XOR Problem traveling salesman problem, Image compression suing MLPs character retrieval using Hopfield networks, Introduction to Genetic Algorithm hybrid systems etc. Recent advances in soft computing applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.

CO2: Understand the concepts of fuzzy sets, knowledge representation using fuzzy rules, approximate reasoning, fuzzy inference systems, and fuzzy logic.

CO3: Understand the fundamental theory and concepts of neural networks, Identify different neural network architectures, algorithms, applications and their limitations.

CO4: Understand appropriate learning rules for each of the architectures and learns everal neural network paradigms and its applications.

Books and References

1. Learning and Soft Computing by V. Kecman, Pearson.
2. Genetic Algorithms in Search Optimization and Machine Learning by D. E. Goldberg, Addison Wesley.
3. Neural Network and fuzzy systems by B. Kosko, Prentice Hall of India.
4. Intelligent Hybrid Systems by S. Goonatilake and S. Khebbal, Wiley.

Course Name: **Speech and Natural Language Processing**
Course Code: **CS-713**
Course Type: **Programme Elective-I**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To develop an in-depth understanding of both the algorithms available for the processing of linguistic information and the underlying computational properties of natural languages.
- To make students focus on the computational properties of natural languages and of the algorithms used to process them.
- To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.

Course Content

Introduction, Brief Review of Regular Expressions and Automata, Finite State Transducers, Word level Morphology and Computational Phonology, Basic Text to Speech; Introduction to HMMs and Speech Recognition. Indian language case studies, Part of Speech Tagging; Parsing with CFGs, Probabilistic Parsing. Representation of Meaning, Semantic Analysis, Lexical Semantics, Word Sense, Disambiguation, Discourse understanding, Natural Language Generation, Techniques of Machine Translation, Indian Language case studies.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Get acquainted with natural language processing and learn how to apply basic algorithms in this field.

CO2: Grasp basics of knowledge representation, inference, and relations to the artificial intelligence.

CO3: Improve their programming skills and their knowledge of development tools.

Books and References

1. Speech and Language Processing by Daniel Jurafsky and James H. Martin, Prentice Hall.
2. NLP: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, Prentice Hall, New Delhi.
3. Language as a Cognitive Process by T. Winograd, Addison-Wesley.

Course Name: **Data Mining**
Course Code: **CS-714**
Course Type: **Programme Elective-I**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To introduce students to the basic applications, concepts, and techniques of data mining.
- To develop skills for using recent data mining software (example R) to solve practical problems in a variety of disciplines.
- To gain experience doing independent study and research.

Course Content

Types of data mining problems. The process of data mining. Statistical evaluation of big data: statistical prediction, performance measures, pitfalls in data-mining evaluation. Data preparation: data models, data transformations, handling of missing data, time-dependent data, and textual data. Data reduction: feature selection, principal components, smoothing data, case subsampling. Predictive modeling: mathematical models, linear models, neural nets, advanced statistical models, distance solutions, logic solutions, decision trees, decision rules, model combination. Solution analyses: graphical trend analyses, comparison of methods. Future trends: text mining, visualization, distributed data. Case studies.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Categorize and carefully differentiate between situations for applying different data-mining techniques: frequent pattern mining, association, correlation, classification, prediction, and cluster and outlier analysis.
CO2: Design and implement systems for data mining.
CO3: Evaluate the performance of different data-mining algorithms.
CO4: Propose data-mining solutions for different applications.

Books and References

1. Data Mining Concepts and Techniques by Jiawei Han and Michelien Kamber, Morgan Kaufmann.
2. Data Warehousing, Data mining and OLAP by Alex Berson and Stephen Smith, McGraw Hill.

Course Name: **Bioinformatics** Course Code: **CS-715** Course Type: **Programme Elective-I**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To introduce the fundamental concepts of Bioinformatics.
- To develop skills in designing biological database and retrieving.
- To explore the different paradigms in Data mining, Pattern Recognition, Soft computing.
- To apply appropriate sequence analysis methods for analyzing bio-molecular sequences.

Course Content

Introduction; Databases mapping, sequence, structure, non-redundant, Sequence alignment pair wise and multiple, phylogenetic, Structure prediction methods, homology, threading, abinitio, Sequence analysis class and secondary structure prediction, motifs PROSITE, detecting functional sites in DNA, OR Finder, Computer science perspective pattern recognition, hidden Markov models, Data Mining using Soft Computing Techniques.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the basic terminologies used in the field of Bioinformatics.

CO2: Understand the databases related to Bioinformatics and will be able to comprehend data in these databases.

CO3: Perform sequence alignment and analysis.

CO4: Apply computational techniques and prediction algorithms to solve problems related to the domain of Bioinformatics.

Books and References

1. Bioinformatics by D. Baxevanis and B. F. F. Ouellette, Wiley Interscience.
2. Introduction to bioinformatics by M. Lesk, Oxford University Press.
3. Computational methods in molecular biology by S. L. Salzberg, D. B. Searls and S.Kasif (eds.), Elsevier.
4. Computer methods for macromolecular sequence analysis by R. F. Doolittle, Academic Press.
5. Guide to human genome computing by M. Bishop, Academic Press.

Course Name: **Formal Languages and Automata Theory**
Course Code: **CS-721**
Course Type: **Programme Elective-II**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To identify different formal language classes and their relationships.
- Design grammars and recognizers for different formal languages.
- To understand basic properties of Turing machines and computing with Turing machines.
- Know the concepts of decidability and undecidability, the concepts of NP-completeness and NP-hard problem.

Course Content

Propositional calculus and Predicate Calculus, Satisfiability and validity, Notions of soundness and completeness, Chomsky Hierarchy of Grammars and the corresponding acceptors, Turing Machines, Recursive and Recursively Enumerable Languages; Operations on Languages, closures with respect to the operations. Church-Turing Thesis, Decision Problems, Decidability and Undecidability, Halting Problem of Turing Machines; Problem reduction (Turing and mapping reduction).Time Complexity and Space Complexity.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Relate practical problems to languages, automata, computability, and complexity.

CO2: Students will be able to explain the relationship among language classes and grammars with the help of Chomsky Hierarchy.

CO3: Define the notions of computability and decidability.

Books and References

1. Introduction to Automata theory by John E. Hopcroft, Rajeev Motwani and Jeffery Ullman, Pearson Education.
2. Languages and Machines: An Introduction to the Theory of Computer Science by Thomas A. Sudkamp, Pearson Education.
3. Introduction to the Theory of computation by Michael Sipser, Thomson Publication.
4. Introduction to Formal Languages, Automata Theory and Computation by Kamala Krithivasan and Rama R, Pearson Education.

Course Name: **Parallel Algorithms** Course Code: **CS-722** Course Type: **Programme Elective-II**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To introduce principles and design techniques of parallel algorithms and data structures for various parallel architectures.
- To emphasize on theoretical aspect as well as empirical development of algorithms.

Course Content

Introduction to parallel algorithms: EREW, CREW, CRCW PRAMs and interconnection network models, Need for Parallel Processing , Data and Temporal Parallelism, Models of Computation, RAM and PRAM Model, Shared Memory and Message Passing Models, Processor Organizations, PRAM Algorithms, Analysis of PRAM Algorithms Parallelization of Algorithm: Different Parallel Programming Models, Brent's Theorem, Message passing algorithm, Load balancing and termination detection, programming with shared memory, programming for Distributed shared memory. Basic Parallel Algorithmic Techniques: Divide-and-Conquer, Partitioning, pipelining, Accelerated Cascading, Symmetry Breaking, Synchronization (Locked, Lock-free) Parallel Algorithms and Data organization for shared/distributed memory, Min/Max, Sum Searching, Merging, Sorting, Various Parallel Sorting and Sorting Networks, Introduction to Graphics Processing Units (GPUs), CUDA Programming Model, Various operation in CUDA, Optimization Techniques: Understanding thread and blocks execution, Memory Bank Conflicts, Parallel Thread Execution, Control Flow, Precision, Optimizing CPU-GPU usage.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Design efficient parallel algorithms and applications.

CO2: To use large scale parallel machines to solve problems.

Books and References

1. Parallel Programming Techniques and Applications by B. Wilkinson and M. Allen, Pearson.
2. An Introduction to Parallel Algorithms by Joseph Jaja, Addison-Wesley Professional.
3. CUDA by Example: An Introduction to General-Purpose GPU Programming by Jason Sanders and Edward Kandrot, Nvidia.
4. Parallel Computers- Architecture and Programming by V. Rajaraman and C.S.R Murthy, PHI.
5. Programming Massively Parallel Processors: A Hands-on Approach by David Kirk and Wen-mei Hwu, Elsevier.
6. Parallel Programming in C with MPI and OpenMP by Michael J Quinn, McGraw Hill.

Course Name: **Information Theory and Coding**
Course Code: **CS-723**
Course Type: **Programme Elective-II**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To learn concepts in information theory, and the performance characteristics of an ideal communications system.
- To know about the fundamentals in Information coding and its applications.

Course Content

Introduction to information Theory, Information and entropy, properties of entropy of a binary memory less source, Measure of Information, Source Coding, Shannon Fano coding, Huffman coding, Lempel Ziv coding, channel coding, Channel capacity, noisy channel. Coding theorem for DMC. Linear block codes, generator matrices, parity check matrices, encoder syndrome and error detection minimum distance, error correction and error detection capabilities, cyclic codes, coding and decoding. Coding convolutional codes, encoder, generator matrix, transform domain representation state diagram, distance properties, maximum likelihood decoding, Viterbi decoding, sequential decoding, interleaved convolutional codes.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understood that how the quantity of information could be measure.

CO2: Understood the concept and properties of entropy and mutual information as applied to information.

CO3: Construct compact and non-compact codes for a given data ensemble.

Books and References

1. Information Theory Coding and Cryptography by R. Bose, Tata McGraw Hill.
2. The Theory of Error Correcting Codes by F. J. MacWilliams and N. J. A. Sloane, Elsevier.
3. Coding and Information Theory by S. Roman, Springer.
4. The Theory of Information and Coding by R. J. McEliece, Cambridge University Press.
5. Elements of Information Theory by T. M. Cover and J. A. Thomas, Wiley.

Course Name: **Game Theory** Course Code: **CS-724** Course Type: **Programme Elective-II**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To teach students some strategic considerations to take into account making their choices.
- To learn basic concepts of game theory.

Course Content

Basic Solution concepts and computational issues: Games, Old and New; Games Strategies, Costs and Payoff, Basic Solution Concepts; Finding Equilibria and Learning in Games. Refinement of Nash: Games with Turns and Sub game, Perfect Equilibrium: Cooperative games, markets and their Algorithmic Issues. The Complexity of finding Nash Equilibria: Introduction, Lemke Howson algorithm, succinct representation of games. Graphical Games: Computing Nash Equilibria in Tree Graphical Games, Graphical Games and correlated Equilibria, Cryptography and Game theory: Cryptographic notation and settings, game theory notation and settings, cryptographic influence on game theory and Game theoretic influence on cryptography. Distributed algorithmic mechanism design: two examples of DAMD, Interdomain routing Cost sharing. Incentive and Pricing in Communication Networks Large network Competitive model, Pricing and Resource allocation Game theoretic model Incentive and Information security: Misaligned incentive. Informational Asymmetries, Complex network and topology.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Solve strategic games between two and more agents in non-cooperative scenario.

CO2: Analyze and solve both simultaneous-moves and sequential-moves games.

CO3: Learn different methods to solve games.

Books and References

1. A Course in Game Theory by M. J. Osborne and A. Rubinstein, MIT Press.
2. An Introduction to Game Theory by M. J. Osborne, Oxford University Press.
3. Algorithmic Game Theory by N. Nisan, T. Rougharden, E. Tardos and V. V. Vazirani, Cambridge University Press.
4. Fun and Games: A Text on Game theory by K. Binmore, AIBS publisher.

Course Name: **Combinatorial Optimization** Course Code: **CS-725** Course Type: **Programme Elective-II**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- Introducing the fundamentals of combinatorial optimization techniques.
- Learning how to model problems using mathematical programs.
- Learn to formulate and solve traditional problems in combinatorial optimization using combinatorial algorithm.
- Introducing the concept of linear programming and matching algorithm.

Course Content

Optimization Problem: Global and Local Optima; Convex sets and functions; Convex Programming Problem; Simplex Algorithm: Forms of linear programming problem; Geometry of linear program; Duality: Dual of a linear program in general form; shortest path problem and its dual; Dual simplex algorithm; Primal dual algorithm: Shortest Path Problem, Max Flow; Algorithms and complexity: Computability; time bound; analysis of algorithm; polynomial time algorithm; Algorithm for matching; weighted matching. Special topics in Combinatorial Optimization.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Identify and classify combinatorial optimization problems with real-world problems.

CO2: Identify, classify and implement algorithm to solve combinatorial optimization problems.

CO3: Model problems using linear programming.

CO4: An understanding of the inherent complexity of problems: Polynomial time, NP-completeness.

Books and References

1. Combinatorial optimization: algorithm and Complexity by C.H. Papadimitriou and K. Steiglitz, Prentice Hall of India.
2. Art of Computer Programming by D. Knuth, Vol. IV, Addison Wesley.
3. Computational Complexity by C.H. Papadimitriou, Addison Wesley.

Course Name: **Computer Vision and Image Processing**
Course Code: **CS-731**
Course Type: **Programme Elective-III**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To study the image enhancement and compression techniques.
- To study image restoration and segmentation procedures.
- Assess the performance of image processing algorithms and systems.

Course Content

Digital Image Fundamentals; Image Enhancement in Spatial Domain; Gray Level Transformation, Histogram Processing, Spatial Filters; Image Transforms; Fourier Transform and their properties, Fast Fourier Transform, Other Transforms; Image Enhancement in Frequency Domain; Color Image Processing; Image warping and restoration; Image Compression; Image Segmentation; edge detection, Hough transform, region based segmentation; Morphological operators; Representation and Description; Features based matching and Bayes' classification, Imaging geometry, shape from shading.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Review the fundamental concepts of a digital image processing system.
CO2: Analyze images in the frequency domain using various transforms.
CO3: Examine various types of images, intensity transformations and spatial filtering.
CO4: Evaluate the techniques for image enhancement, compression and image restoration.

Books and References

1. Digital Image Processing by R. Gonzalez and R. E. Wood, Prentice Hall of India.
2. Introductory Computer Vision and Image Procession by Andrian low, McGraw Hill.
3. Digital Image Processing by W.K. Pratt, McGraw Hill.
4. Fundamentals of Digital Image Processing by A.K. Jain, Pearson.

Course Name: **Performance Evaluation of Computer System**
Course Code: **CS-732**
Course Type: **Programme Elective-III**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To provide an introduction to the tools and techniques needed to construct and analyze performance models of computer systems and communication networks.
- To provide a working knowledge of computer performance evaluation and fundamental techniques such as measurement and mathematical modeling.
- To cover discrete and continuous time Markov chain models, queues in isolation and queuing networks.

Course Content

Introduction to Probability Refresher: Bayes' theorem, Conditional probability, Total probability, Discrete and Continuous Random variables, Common distributions, Probability Generating Functions (PGF) and Laplace Transforms (LST), Numerous examples from computer networking, Stochastic processes, Discrete time Markov chains (DTMC). Continuous time Markov chains (CTMC), Queuing systems (M/M/1, M/M/c/k, M/G/1), Queuing networks, Statistical analysis of simulations, Specific topics: Introduction to performance measures, basic probability review, Markov chains, basic queuing models, introduction to simulation modeling, some advanced queuing models, basic queuing networks.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Get acquainted with tools and techniques used for evaluation of different types of systems.

CO2: Grasp working knowledge of basics of computer performance evaluation and modeling techniques.

CO3: Select appropriate evaluation techniques, performance metrics and workloads for a system.

CO4: Design measurement and simulation experiments to provide the most information with the least effort.

Books and References

1. Simulation Modeling and Analysis by Law and Kelton, McGraw Hill.
2. Fundamentals of Queueing Theory by D. Gross and C. M. Harris, John Wiley and Sons.
3. Queueing Systems: Vol. I & II by L. Kleinrock, John Wiley and Sons.
4. A Gentle Introduction to Some Basic Queuing Concepts by William Stallings, PHI.
5. Quantitative System Performance - Computer System Analysis Using Queueing Network Models by Edward D. Lazowska, John Zahorjan, G. Scott Graham, Kenneth C. Sevcik, Prentice-Hall.

Course Name: **Fault Tolerant Computing**
Course Code: **CS-733**
Course Type: **Programme Elective-III**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To know the different advantages and limits of fault avoidance and fault tolerance techniques.
- To impart the knowledge about different types of redundancy and its application for the design of computer system being able to function correctly even under presence of faults and data errors.
- To understand the relevant factors in evaluating alternative system designs for a specific set of requirement.
- To understand the subtle failure modes of "fault-tolerant" distributed systems.

Course Content

Introduction to Fault Tolerant Computing. Basic concepts and overview of the course; Faults and their manifestations, Fault/error modeling, Reliability, availability and maintainability analysis, System evaluation, performance reliability tradeoffs. System level fault diagnosis, Hardware and software redundancy techniques. Fault tolerant system design methods, Mobile computing and Mobile communication environment, Fault injection methods, Software fault tolerance, Design and test of defect free integrated circuits, fault modeling, built in self-test, data compression, error correcting codes, simulation software/hardware, fault tolerant system design, CAD tools for design for testability. Information Redundancy and Error Correcting Codes, Software Problem. Software Reliability Models and Robust Coding Techniques, Reliability in Computer Networks Time redundancy. Re execution in SMT, CMP Architectures, Fault Tolerant Distributed Systems, Data replication. Case Studies in FTC: ROC, HP Non Stop Server. Case studies of fault tolerant systems and current research issues.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Become familiar with general and state of the art techniques used in design and analysis of fault-tolerant digital systems.

CO2: Be familiar with making system fault tolerant, modeling and testing, and benchmarking to evaluate and compare systems.

Books and References

1. Fault Tolerant Computer System Design by D. K. Pradhan, Prentice Hall.
2. Fault Tolerant Systems by I. Koren, Morgan Kauffman.
3. Software Fault Tolerance Techniques and Implementation by L. L. Pullum, Artech House Computer Security Series.
4. Reliability of Computer Systems and Networks: Fault Tolerance Analysis and Design by M. L. Shooman, Wiley.

Course Name: **Embedded Systems** Course Code: **CS-734** Course Type: **Programme Elective-III**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To discuss the major components that constitutes an embedded system.
- To implement small programs to solve well-defined problems on an embedded platform.
- To develop familiarity with tools used in an embedded environment.

Course Content

Introduction: Embedded system, Processor, hardware units, software embedding, SOC, NOC, VLSI circuit; Device and Device drivers, I/O devices, timer and counting devices, serial communication using IC, LAN and advanced I/O buses between the networked multiple devices, Host system, parallel communication using ISA, PCI, PCI X, and advanced buses, device drivers, parallel port device drivers in a system, serial port device drivers. Interrupt service handling mechanism; Software and programming concepts: processor and memory selection for embedded system, embedded programming in C++, Java and UML, multiple processes and applications, problem of sharing data by multiple tasks and routines, interprocess communication; Real time OS: OS services, I/O subsystem, Network OS, Real time Embedded system, Need of well tested and debugged RTOS, Introduction to C/OS II. Case Studies of programming with RTOS: Smart card embedded system, Hardware and Software co design: specification and design of an embedded system, use of software tools for development of an embedded system. Recent advances in embedded applications.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Become familiar with programming environment used to develop embedded systems.

CO2: Understand key concepts of embedded systems like IO, timers, interrupts, interaction with peripheral devices.

CO3: Learn debugging techniques for an embedded system.

Books and References

1. Embedded System Architecture, Programming and Design by R. Kamal, Tata McGraw Hill.
2. Hardware Software Codesign of Embedded System by R. Niemann, Kulwer Academic.
3. Embedded Real Time System Programming by Sriram. V. Iyer and P. Gupta, Tata McGraw Hill.
4. Computer as Components: Principles of Embedded Computer System Design by W. Wolf, Elsevier.
5. Embedded System Design by S. Heath, Elsevier.
6. Real Time Systems Theory and Practice by R. Mall, Pearson.
7. Embedded System Design: A Unified Hardware/Software Approach by F. Vahid & T. Givargis, Wiley.
8. Network on Chips: Technology and Tools by G. D. Michelli and Luca Benin, Morgan & Kaufman Publication.

Course Name:	**Cryptography and Computer Security**
Course Code:	**CS-741**
Course Type:	**Programme Elective-IV**
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To understand basics of Cryptography and Network Security
- To learn about how to maintain the Confidentiality, Integrity and Availability of a data.
- Encrypt and decrypt messages using block ciphers, sign and verify messages using well known signature generation and verification algorithms.
- To understand various protocols for network security to protect against the threats in the networks.

Course Content

Introduction, need and basic goals for computer security, security threats etc.Cryptography, elementary number theory, finite fields, arithmetic and algebraic algorithms, secret key and public key cryptography, Pseudo random bit generators, Block and stream ciphers, Hash functions and message digests, Public key encryption, Probabilistic encryption, Authentication, Digital signatures, Zero knowledge interactive protocols, Elliptic curve cryptosystems, Formal verification, Cryptanalyses, Hard problems. Network Security: problems in network security; kinds of attacks, PKI, key exchange protocols, example protocols such as PGP, Kerberos, IPSEC/VPN, SSL, S/MIME, etc. Protocol vulnerabilities: examples of protocol vulnerabilities such as in TCP/IP, denial of service attacks etc. Tools for network security such as firewalls and intrusion detection systems.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Understand the most common type of cryptography algorithms and network security concepts and application.

CO2: Understand security protocols for protecting data on networks.

CO3: Develop an understanding of security policies as well as protocols to implement such policies in the form of message exchanges.

CO4: Understand vulnerability assessments and the weakness of using passwords for authentication.

Books and References

1. Cryptography, Theory and Practice by Douglas R. Stinson, CRC Press.
2. Cryptography and Network Security: Principles and Practices by W. Stallings, Prentice Hall.
3. Applied cryptography by B. Schneier, John Wiley & Sons.
4. Handbook of Applied Cryptography by A. Menezes, P. Van Oorschot, S. Vanstone, CRC Press.
5. Network Security by C. Kaufman, R. Perlman and M. Speciner, Prentice Hall.
6. Introduction to Cryptography with coding theory by Wade Trappe and Lawrence C. Washington, Pearson Education.

Course Name: **Intrusion Detection System**
Course Code: **CS-742**
Course Type: **Programme Elective-IV**
Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To understand when, where, how, and why to apply Intrusion Detection tools and techniques in order to improve the security posture of an enterprise.
- To apply knowledge of the fundamentals and history of Intrusion Detection in order to avoid common pitfalls in the creation and evaluation of new Intrusion Detection Systems.
- To analyze intrusion detection alerts and logs to distinguish attack types from false alarms.

Course Content

Introduction: IDS, Types of IDS, host based IDS, Network based IDS, Stack based IDS, signature Based IDS, anomaly based IDS, TCP/IP and security concerns, DNS and security concerns, Mail server and security concerns, Web Server and security concern, firewall, Types of Intrusion, Symptoms that help in intrusion detection, statistical pattern recognition for detection and classification of attacks, vulnerabilities and Threats; Trojan Remote Access Trojan RAT, Virus, Worms and Malwares. Data Collection Mechanism: Data Sampling, Packet Sampling, Flow Sampling, techniques for visualizing network data, Packet Sampling tools, Tcpdump windump, Wireshark tool, Writing Tcpdump/Windump Filters, libcap/winpcap libraries, pcap file, sniffing and spoofing tools, data and methodologies of computer intrusion detection, statistical & machine approaches to detection of attacks on computers.Attacks and Packet analysis: network based attacks such as probes & denial of service attacks, host based attacks such as buffer overflows and race conditions, malicious codes, Examining Packet Header Fields, normal and abnormal values in IP, TCP, UDP, and ICMP header fields, Fragmentation theory, packet capture examples, fragmentation-based attacks, ICMP protocol, ICMP based attacks, Network Traffic Analysis: malicious, normal and application traffic; discern malicious traffic from false positives. IDS Patterns, DoS attacks, network mapping, and coordinated attacks, Indications & Warnings and Traffic Correlation, Network correlation, Network Situational Awareness, anomaly detection, signature based analysis, Semantic aware signature, policy based analysis, and host based analysis. IDS infrastructure: IDS Architecture, IDS/IPS Management and Architecture Issues with regard to deploying IDS/IPS systems, end point approach to security, system approach to security, IDS Interoperability models: CIDF (Common Intrusion Detection Framework), IDMEF (Intrusion Detection Message Exchange Format), IODEF (Incident Object Description Exchange Format), CVE (Common Vulnerabilities and Exposures), OVAL (Open Vulnerability and Assessment Language). Protocol Analysis: Microsoft Protocols, SMB/CIFS, RPC, and Active Directory protocols, SIP protocol, Chat protocol, the key differences between IPv4 and IPv6, IPv6 based attacks. IDS tools: Snort and Bro IDS tools, NIDS Evasion, Insertion, and Checksums to confuse NID systems, Snort Fundamentals and Configuration, Snort GUIs & Sensor Management, Snort Performance, Active Response & Tagging, Snort Rules, Stimulus Response, hosts response to both normal and abnormal traffic, Advanced Snort Concepts as rule ordering and reduction of false negatives and positives. Evaluation and tuning of IDS, Cross over Rate (CER) of IDS. Advanced topics: honeypots, shadow honeypots.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Explain the fundamental concepts of Network Protocol Analysis and demonstrate the skill to capture and analyze network packets.

CO2: Use various protocol analyzers and Network Intrusion Detection Systems as security tools to detect network attacks and troubleshoot network problems.

Books and References

1. Network Intrusion Detection by Stephen Northcutt and Judy Novak, Sams Publishing.
2. Extrusion Detection: Security Monitoring for Internal Intrusions by Bejtlich, Pearson Education.
3. Guide to Intrusion Detection and Prevention Systems (IDPS) by Karen Scarfone and Peter Mell, National Institute of Standards and Technology (NIST).
4. CCNP Security: Intrusion Prevention and Intrusion Detection Systems by David Burns, Odunayo Adesina and Keith Barker, Cisco Press.
5. Intrusion Detection and Correlation: Challenges and Solutions by Christopher Kruegel, Fredrik Valeur and Giovanni Vigana, Springer.

Course Name: **Biometric Security**
Course Code: **CS-743**
Course Type: **Programme Elective-IV**

Contact Hours/Week: **4L** Course Credits: **04**

Course Objectives

- To understand fundamentals of biometrics.
- To gain a broader knowledge and understanding of the different Biometric techniques.
- To learn about biometrics for security.

Course Content

Security via biometrics, space domain based biometrics and recognition techniques. Correlation based biometric filters, Basic theory of correlation filters, Design of advanced correlation filters that offer tolerance to expected impairments, methods to implement digital correlation, applications of correlation filters. Special topics in biometric security.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Have a good understanding of the various modules constituting a bio-metric system.
CO2: Get familiar with different bio-metric traits and to appreciate their relative significance.
CO3: Evaluate and design security systems incorporating bio-metrics.
CO4: Identify the security issues in various fields and able to resolve it via biometrics.

Books and References

1. Biometrics for Network Security by P. Reid, Pearson Press.
2. Biometrics by J. D. Woodward, N. M. Orlans and P. T. Higgins, Dreamtech Publishers.
3. Biometrics by S. Nanavati, M. Thieme and R. Nanavati, Wiley Publishers.

Course Name: **Cluster and Grid Computing** Course Code: **CS-744** Course Type: **Programme Elective-IV**	
Contact Hours/Week: **4L**	Course Credits: **04**

Course Objectives

- To provide an insight for achieving cost efficient high performance system.
- To provide understanding of the design and architecture of grid and cluster computing.

Course Content

Introduction: High Performance Computing (HPC), Grand Challenge Problems Computational and communication intensive, Parallel Architectures Classifications SMP, MPP, NUMA, Clusters and Components of a Parallel Machine, Conventional Supercomputers and its limitations, Multi-processor and Multi Computer based Distributed Systems. Cluster and Grids: Cluster Components Processor/machine, High Speed Interconnections goals, topology, latency, bandwidth, Example Interconnect: Myrinet, Inifiniband, QsNet, Fast Ethernet, Gigabit Ethernet, Light weight Messaging system/Light weight communication Protocols, Cluster Middleware Job/Resource Management System, Load balancing, Scheduling of parallel processes, Enforcing policies, GUI, Introduction to programming tools such as PVM, MPI, Cluster Operating Systems Examples: Linux, MOSIX, CONDOR, Characteristics of Grid, Computational services, Computational Grids, Data grids/Storage grids, management and applications, Different components of Grid fabric, Grid middleware, Grid applications and portal, Globus toolkit Ver.2.4, web services, MDS,GRAM, Grid Security Cryptography, Authentication, Integrity, Digital Signature, Digital Certificates, Certificate Authority, MD 5, RSA, GSI,GSSAPI, Directory Service, LDAP,GRID FTP,GASS Fault Tolerance: Fault detection and diagnosis of Clusters and Grids. Recent advances in cluster and grid computing.

Course Outcomes

Upon successful completion of the course, the students will be able to

CO1: Have knowledge of Grid Computing, Web Services, and Service-oriented architecture, Architecture for grid computing, Cluster Computing, Process scheduling and load balancing.

CO2: Identify the resource selection for Grid environment.

CO3: Understand the data management and transfer in Grid environments.

Books and References

1. Grid Computing by D. Janakiram, Tata Mcgraw Hill.
2. High Performance Cluster Computing, Volume 1 and 2 by R. K. Buyya, Prentice Hall.
3. Fault Tolerance in Distributed Systems by P. Jalote, Prentice Hall.
4. Cluster Computing by R. K. Buyya and C. Szyperski, Nova Science, New York, USA.
5. Market oriented Grid and Utility Computing by R. K. Buyya and K. Bubendorfer, Wiley.
6. Grid Computing by J. Jaseph and C. Fellenstein, Pearson.