



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SR ENGINEERING COLLEGE

Approved by AICTE, New Delhi | Affiliated to JNTUH | Accredited by NAAC 'A' Grade
Ananthasagar (V), Hasanparthy (M), WARANGAL - 506 371
(*An Autonomous Institution*)

**COURSE STRUCTURE
AND
DETAILED SYLLABUS**

FOR

**I - M.TECH(CSE)
(COMPUTER SCIENCE AND
ENGINEERING)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SR ENGINEERING COLLEGE
(RA15) M.Tech (CSE)
COURSE STRUCTURE AND SYLLABUS

BRANCH: C S E

YEAR:I SEMESTER: I

S.No .	Subject Code	Subject	L	P	C	CIE	SEE
1	CS201	Advanced Data Structures and Algorithms	4	0	4	40	60
2	CS202	Big data and Analytics	4	0	4	40	60
3	CS203	Machine Learning	4	0	4	40	60
4	CS204	Distributed Systems	4	0	4	40	60
5	Elective - I		4	0	4	40	60
	CS205	Software Process and Project Management					
	CS206	Digital Image Processing					
	CS207	Pattern Recognition					
6	Elective - II		4	0	4	40	60
	CS208	Natural Language Processing					
	CS209	Game theory					
	CS210	Software Architecture and Design Patterns					
7	CS221	Advanced Algorithms Lab	0	3	2	40	60
8	CS222	Data Analytics Lab	0	3	2	40	60
Total					28	320	480

BRANCH: CSE**YEAR: I SEMESTER: II**

S. No.	Subject Code	Subject	L	P	C	CIE	SEE
1	CS211	Advanced Network Programming	4	0	4	40	60
2	CS212	High Performance Computing	4	0	4	40	60
3	CS213	Web Services and Service Oriented Architecture	4	0	4	40	60
4	CS214	Number theory and Cryptography	4	0	4	40	60
5	Elective - III		4	0	4	40	60
	CS215	Adhoc and Sensor Networks					
	CS216	Cloud Computing					
	CS217	Database Security					
6	Elective - IV		4	0	4	40	60
	CS218	Semantic Web and Social Networks					
	CS219	Software Quality Assurance & Testing					
	CS220	Computational Astronomy					
7							
	CS223	High Performance Computing Lab	0	3	2	40	60
	CS224	Network Programming Lab	0	3	2	40	60
Total :					28	320	480

BRANCH: CSE**YEAR: II SEMESTER: I**

S. No.	Subject Code	Subject	L	P	C	CIE	SEE
1	CS225	Comprehensive Viva	-	-	4	-	100
2	CS226	Project Work Review - I	-	24	12	50	-
Total:					16	50	100

BRANCH: CSE**YEAR: II SEMESTER: II**

S. No.	Subject Code	Subject	L	P	C	CIE	SEE
1	CS227	Project Work Review - II	-	8	4	50	-
2	CS228	Project Evaluation (Viva - Voce)	-	16	12	-	150
Total:					16	50	150

(CS201)ADVANCED DATA STRUCTURES AND ALGORITHMS

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURES OBJECTIVES:

1. The fundamental design, analysis, and implementation of basic data structures.
2. Basic concepts in the specification and analysis of programs.
3. Principles for good program design, especially the uses of data abstraction.
4. Significance of algorithms in the computer field
5. Various aspects of algorithm development
6. Qualities of a good solution

COURSE OUTCOMES:

1. Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Master a variety of advanced abstract data type (ADT) and data structures and their implementations.
3. Master different algorithm design techniques (brute-force, divide and conquer, greedy, etc)
4. Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

UNIT I

Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big Oh, Omega and Theta notations, Complexity Analysis Examples. Data structures-Linear and non linear data structures, ADT concept, Linear List ADT, Array representation, Linked representation, Vector representation, singly linked lists -insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, circular lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

UNIT II

Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack,implementation of recursion, Circular queue-insertion and deletion, Dequeue ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap, java.util package-ArrayList, Linked List, Vector classes, Stacks and Queues in java.util, Iterators in java.util.

UNIT III

Searching–Linear and binary search methods, Hashing-Hash functions, Collision Resolution methods-Open Addressing, Chaining, Hashing in java.util-HashMap, HashSet, Hashtable. Sorting –Bubble sort, Insertion sort, Quick sort, Merge sort, Heap sort, Radix sort, comparison of sorting methods.

UNIT IV

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree ADT, representations, recursive and non recursive traversals, Java code for traversals, Threaded binary trees. Graphs- Graphs terminology, Graph ADT, representations, graph traversals/search methods-dfs and bfs, Java code for graph traversals, Applications of Graphs-Minimum cost spanning tree using Kruskal's algorithm, Dijkstra's algorithm for Single Source Shortest Path Problem.

UNIT V

Search trees- Binary search tree-Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees-Definition and examples only, Red Black trees – Definition and examples only, B-Trees-definition, insertion and searching operations, Trees in java.util- TreeSet, Tree Map Classes, Tries(examples only), Comparison of Search trees. Text compression-Huffman coding and decoding, Pattern matching-KMP algorithm.

TEXT BOOKS:

1. S. Sahni, "Data structures, Algorithms and Applications in Java", Universities Press. [ISBN:0-07-109217-x]
2. Adam Drozdek, "Data structures and Algorithms in Java", 3rd edition, Cengage Learning. [ISBN:978-9814239233]

REFERENCE BOOKS:

1. R.Lafore "Data structures and Algorithms in Java", Pearson education. ISBN: 9788 131718124.
- 2.. P.Tremblay and G.A.Cheston "Data structures and Software Development in an Object-Oriented Domain", Java edition, Pearson Education.

(CS202) BIG DATA & ANALYTICS

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURES OBJECTIVES:

1. To understand the 5V paradigm of the big data
2. To apply analytics to gain insights into the big data
3. To comprehend the hadoop architecture for storage & processing the big data
4. To acquire programming skills in hadoop echo system
5. To explore NoSQL databases

COURSE OUTCOMES:

1. To Understand big data for business intelligence
2. To acquire skills in MapReduce analytics using Hadoop and related tools
3. To manage Big data without SQL
4. To Understand Nosql big data management

UNIT I

Types of digital data: structured, semi-structured & unstructured. Big data definition: Volume, Velocity, Variety, veracity& value. Big data analytics skills: analytics classification, In-memory analytics, parallel processing & distributed systems. CAP theorem. The Big data Technology landscape: SQL vs NoSQL. Introduction to Hadoop.

UNIT II

Exploring Hadoop architecture: HDFS, The Hadoop Distributed File System. Basic concepts, namenode, datanodes, The design of HDFS, Interfaces basic commands for file system operations, Hadoop I/O.

UNIT III

Introduction to MapReduce programming: Mapper, Reducer, Combiner, Partitioner, Searching, Sorting, Compression. Developing a MapReduce Application (*programming language is not a binding*), MapReduce workflows, MapReduce Types & Formats, Features of MapReduce.

UNIT IV

Introduction to HIVE: Architecture, File format, Hive Query Language (HQL). Programming with Hive.

Introduction to Pig: Pig Latin Overview, data types, running Pig, Execution models, Pig Commands, Application development using Pig. Pig vs Hive

UNIT V

Introduction to HBase : NoSQL a column oriented database. Example schemas & comparisons with RDBMS.

Introduction to ZooKeeper : Installing & running zookeeper. Zookeeper service.

Introduction to sqoop: A sample import using sqoop. Importing large objects. Working with imported data. A deeper look at sqoop export.

TEXT BOOKS:

1. Seema Acharya & Subhashini Chellappan “BIG data and Analytics” Wiley India Publishers, ISBN:978-81-265-5478-2
2. Tom White “Hadoop the Definitive Guide” O’Reilly Publishers, ISBN:978-1-449-31152-0

REFERENCE BOOKS:

1. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.
2. Vignesh Prajapati, Big data analytics with R and Hadoop, SPD 2013.
3. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
4. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011
5. Alan Gates, "Programming Pig", O'Reilley, 2011.

WEB LINKS:

1. <https://hadoop.apache.org/>
2. www.cloudera.com
3. <http://hortonworks.com/>
4. <http://www.edureka.co/>

(CS203) MACHINE LEARNING

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To be able to formulate machine learning problems corresponding to different applications.
2. To understand a range of machine learning algorithms along with their strengths and weaknesses.
3. To understand the basic theory underlying machine learning.
4. To be able to apply machine learning algorithms to solve problems of moderate complexity.
5. To be able to read current research papers and understands the issues raised by current research.
6. to understand inductive and analytical learning

COURSE OUTCOMES:

1. Apply machine learning: data, model selection, model complexity, etc.
2. Apply design and analyze the popular machine learning approaches.
3. Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
4. Be able to design and implement various machine learning algorithms in a range of real-world applications.

UNIT I

Introduction: Well-posed learning problems, Designing a learning system, Perspectives and issues in machine learning **Concept learning and the general to specific ordering** – Introduction, A concept learning task, Concept learning as search, Find-S: finding a maximally specific hypothesis, Version spaces and the candidate elimination algorithm, Remarks on version spaces and candidate elimination, Inductive bias

UNIT II

Decision Tree learning: Introduction, Decision tree representation, Appropriate problems for decision tree learning, The basic decision tree learning algorithm, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning **Artificial Neural Networks** – Introduction, Neural network representation, Appropriate problems for neural network learning, Perceptions, Multilayer networks and the back propagation algorithm, Remarks on the back propagation algorithm, An illustrative example face recognition Advanced topics in artificial neural networks. **Evaluation Hypotheses** – Motivation, Estimation hypothesis accuracy, Basics of sampling theory, A general approach for deriving confidence intervals, Difference in error of two hypotheses, Comparing learning algorithms.

UNIT III

Bayesian learning: Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum likelihood and least squared error hypotheses, Maximum likelihood hypotheses for predicting probabilities, Minimum description length principle, Bayes optimal classifier, Gibbs algorithm, Naïve Bayes classifier, An example learning to classify text, Bayesian belief networks The EM algorithm **Computational learning theory** – Introduction, Probability learning an approximately correct hypothesis, Sample complexity for Finite Hypothesis Space, Sample Complexity for infinite Hypothesis Spaces, The mistake bound model of learning **Instance-Based Learning-** Introduction, k -Nearest Neighbour Learning, Locally Weighted Regression, Radial Basis Functions, Case-Based Reasoning, Remarks on Lazy and Eager Learning **Genetic Algorithms** – Motivation, Genetic Algorithms, An illustrative Example, Hypothesis Space Search, Genetic Programming, Models of Evolution and Learning, Parallelizing Genetic Algorithms.

UNIT IV

Learning Sets of Rules: Introduction, Sequential Covering Algorithms, Learning Rule Sets: Summary, Learning First Order Rules, Learning Sets of First Order Rules: FOIL, Induction as Inverted Deduction, Inverting Resolution. **Analytical Learning** - Introduction, Learning with Perfect Domain Theories: Prolog-EBG Remarks on Explanation-Based Learning, Explanation-Based Learning of Search Control Knowledge

UNIT V

Combining Inductive and Analytical Learning: Motivation, Inductive-Analytical Approaches to Learning, Using Prior Knowledge to Initialize the Hypothesis, Using Prior Knowledge to Alter the Search Objective, Using Prior Knowledge to Augment Search Operators, **Reinforcement Learning** – Introduction, The Learning Task, Q Learning, Non-Deterministic, Rewards and Actions, Temporal Difference Learning, Generalizing from Examples, Relationship to Dynamic Programming.

TEXT BOOKS:

1. Tom M. Mitchell, "Machine Learning", MGH, ISBN-13:9780071154673, 978-1449303716
2. Stephen Marsland, Taylor & Francis (CRC), "Machine Learning: An Algorithmic Perspective". [ISBN-13:978-1-4200-6718-7]

REFERENCE BOOKS:

1. William W Hsieh, "Machine Learning Methods in the Environmental Sciences, Neural Networks", Cambridge Univ Press. [ISBN:9780521181914]
2. Richard o. Duda, Peter E. Hart and David G. Stork, "pattern classification", John Wiley & Sons Inc., 2001. [ISBN:978-0-471-05669-0]

(CS204) DISTRIBUTED SYSTEMS

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are;
2. To list the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions;
3. To recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems;
4. To design a distributed system that fulfills requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognize when this is not possible, and explain why;
5. To build distributed system software using basic OS mechanisms as well as higher-level middleware and languages.
6. To understand the middleware technologies

COURSE OUTCOMES:

1. To specify the properties of distributed algorithms, so called liveness and safety Properties.
2. Models of distributed systems, including failure and timing model.
3. Master basic algorithms for failure detection, leader elections, broadcast and multicast, basic shared memory in distributed systems, agreement protocols, and group communication.
4. Practice in design and implementation of selected distributed algorithms in middleware designed for group communication.

UNIT I

Characterization of Distributed Systems- Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models- Introduction, Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication. Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

UNIT II

Operating System Support- Introduction, OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture, Distributed File Systems- Introduction, File Service architecture, case study- SUN network file systems. Name Services-Introduction, Name Services and the Domain Name System, Case study of the Global Name Service, Case study of the X.500 Directory Service.

UNIT III

Peer to Peer Systems-Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies-Squirrel, OceanStore. Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging. Coordination and Agreement - Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT IV

Transactions and Concurrency control - Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency Controls. Distributed Transactions - Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery, Replication-Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data.

UNIT V

Security - Introduction, Overview of Security techniques, Cryptographic algorithms, Digital signatures, Case studies-Kerberos, TLS, 802.11 WiFi. Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy case study, Release consistency and Munin case study, other consistency models, CORBA case study- Introduction, CORBA RMI, CORBA Services.

TEXT BOOKS:

1. G Coulouris, J Dollimore and T Kindberg “Distributed Systems Concepts and Design”, Fourth Edition, Pearson Education.[ISBN:0-13-101621-0]
2. S. Ghosh, Chapman & Hall/CRC, Taylor & Francis Group “Distributed Systems, 2010. [ISBN-13:978-1-58488-564-1]

REFERENCE BOOKS:

1. S.Mahajan and S.Shah, “Distributed Computing”,Oxford University Press.[ISBN-13:978-0198061861]
2. Pradeep K.Sinha,” Distributed Operating Systems Concepts and Design”,PHI.[ISBN:7-302-02411-1]

**(CS205) SOFTWARE PROCESS AND PROJECT
MANAGEMENT
(ELECTIVE – I)**

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
2. Compare and differentiate organization structures and project structures.
3. Implement a project to manage project schedule, expenses and resources with the application of suitable project management tools.
4. To understand the future software project management practices
5. To learn the different process models
6. To understand workflows, check points of process

COURSE OUTCOMES:

1. Appreciate the importance of software process and management;
2. Apply project management techniques for information systems development;
3. Apply the management skills to monitor and control a software project;
4. Work together as a team in preparing a report

UNIT I

Software Process Maturity: Software maturity Framework, Principles of Software Process Change, Software Process Assessment, The Initial Process, The Repeatable Process, The Defined Process, The Managed Process, The optimizing Process.

Process Reference Models: Capability Maturity Model (CMM), CMMI, PCMM, PSP, TSP.

UNIT II

Software Project Management Renaissance: Conventional Software Management, Evolution of Software Economics, Improving Software Economics, The old way and the new way.

Life-Cycle Phases and Process artifacts: Engineering and Production stages, inception phase, elaboration phase, construction phase, transition phase, artifact sets, management artifacts, engineering artifacts and pragmatic artifacts, model based software architectures.

UNIT III

Workflows and Checkpoints of process: Software process workflows, Iteration workflows, Major milestones, Minor milestones, Periodic status assessments.

Process Planning: Work breakdown structures, Planning guidelines, cost and schedule estimating process, iteration planning process, Pragmatic planning.

UNIT IV

Project Organizations: Line-of- business organizations, project organizations, evolution of organizations, process automation.

Project Control and process instrumentation: The seven core metrics, management indicators, quality indicators, life-cycle expectations, Pragmatic Software metrics and metrics automation.

UNIT V

CCPDS-R Case Study and Future Software Project Management Practices

Modern Project Profiles, Next-Generation software Economics, Modern Process Transitions.

TEXT BOOKS:

1. Watts S.Humphrey,“Managing the Software Process” , Pearson Education.[ISBN-13:978-0201180954]
2. Walker Royce“Software Project Management”, Pearson Education. [ISBN: 9788177583786]

REFERENCE BOOKS:

1. Agile, Extreme, Robert Wysocki, “Effective Project Management: Traditional”, Sixth edition, Wiley India, rp2011.[ISBN:978-1-118-01619-0]
2. Bob Hughes & Mike Cotterell ,“Software Project Management”, fourth edition, TMH, 2006

(CS206) DIGITAL IMAGE PROCESSING (ELECTIVE – I)

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES

1. To learn about the fundamental aspects of digital image processing and the relationship among the pixels
2. To know about the techniques involved in enhancing an image
3. To learn about the image degradation restoration process
4. To know about the processing of colour images
5. To learn about the image segmentation principles.
6. To know about stastical and structural methods

COURSE OUTCOMES

1. Apply the basic digital image fundamentals and the relationship among the pixels for an image.
2. Accomplish various procedures, enhancement of an image in the spatial domain.
3. Apply the wireless fundamentals to model, design the image techniques.
4. Apply pattern and segmentation, design optimised format of an image.

UNIT I

Introduction: Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system.. Digital Image Fundamentals: A simple image formation model, image sampling and quantization, basic relationships between pixels.

UNIT II

Image Enhancement in the Spatial Domain: Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpeningspatial filters, combining the spatial enhancement methods.

UNIT III

Image Restoration: A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function.

UNIT IV

Color Image Processing: Color fundamentals, color models, pseudo color image processing, basics of full–color image processing, color transforms, smoothing and sharpening, color segmentation.

Image Compression: Fundamentals, image compression models, error-free compression, loss predictive coding, image compression standards. **Morphological Image Processing:**

Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms.

UNIT V

Image Segmentation: Detection of discontinuous, edge linking and boundary detection, thresholding, region-based segmentation. Object Recognition : Patterns and patterns classes, recognition based on decision-theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods – matching shape numbers, string matching.

TEXT BOOKS:

1. Rafeal C.Gonzalez, Richard E.Woods, “Digital Image Processing”, Second Edition, Pearson Education/PHI,ISBN:10:0982085400
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image Processing, Analysis and Machine Vision”, Second Edition, Thomson Learning.ISBN:10:1133593607

REFERENCE BOOKS:

1. Alasdair McAndrew,” Introduction to Digital Image Processing with Matlab”, Thomson Course Tec, ISBN:0534400116
2. Rafeal C.Gonzalez, Richard E.Woods, Steven L. Eddins, “Digital Image Processing using Matlab”, Pearson Education, ISBN:10:0130085197

(CS207) PATTERN RECOGNITION
(ELECTIVE – I)

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To implement pattern recognition and machine learning theories
2. To design and implement certain important pattern recognition techniques
3. To apply the pattern recognition theories to applications of interest
4. To implement the entropy minimization, clustering transformation and feature ordering
5. Understand and analyse methods for automatic training of classification systems
6. to understand the applications of pattern recognition

COURSE OUTCOMES:

1. Design systems and algorithms for pattern recognition (signal classification), with focus on sequences of patterns that are analyzed using, e.g., hidden Markov models (HMM),
2. Analyse classification problems probabilistically and estimate classifier performance,
3. Apply Maximum-likelihood parameter estimation in relatively complex probabilistic models, such as mixture density models and hidden Markov models.
4. Understand the principles of Bayesian parameter estimation and apply them in relatively simple probabilistic models.

UNIT I

Introduction: Basic concepts, Applications, Fundamental problems in pattern Recognition system design, Design concepts and methodologies, Examples of Automatic Pattern recognition systems, Simple pattern recognition model. **Decision And Distance Functions:** Linear and generalized decision functions, Pattern space and weight space, Geometrical properties, implementations of decision functions, Minimum-distance pattern classifications.

UNIT II

Probability: Probability of events: Random variables, Joint distributions and densities, Movements of random variables, Estimation of parameter from samples. **Statistical Decision Making-** Introduction, Baye's theorem, Multiple features, Conditionally independent features, ecision boundaries, Unequal cost of error, estimation of error rates, the leaving-one-out-techniques,characteristic curves, estimating the composition of populations. Baye's classifier for normal patterns.

UNIT III

Non Parametric Decision Making: Introduction, histogram, kernel and window estimation, Nearest neighbour classification techniques. Adaptive decision boundaries, adaptive discriminate functions, Minimum squared error discriminate functions, choosing a decision making techniques. **Clustering and Partitioning** - Hierarchical Clustering: Introduction, agglomerative clustering algorithm, the single-linkage, complete-linkage and average-linkage algorithm. Ward's method Partition clustering-Forg's algorithm, K-means's algorithm, Isodata algorithm.

UNIT IV

Pattern Preprocessing and Feature Selection: Introduction, distance measures, clustering transformation and feature ordering, clustering in feature selection through entropy minimization, features selection through orthogonal expansion, binary feature selection.

UNIT V

Syntactic Pattern Recognition and Application of Pattern Recognition: Introduction, concepts from formal language theory, formulation of syntactic pattern recognition problem, syntactic pattern description, recognition grammars, automata as pattern recognizers, Application of pattern recognition techniques in bio-metric, facial recognition, IRIS scan, Finger prints, etc.,

TEXT BOOKS:

1. Gose. Johnsonbaugh. Jost. "Pattern recognition and Image Analysis", PHI. [ISBN-10:8120314840]
2. Tou. Rafael. Gonzalez. "Pattern Recognition Principle", Pearson Education [ISBN:9788436814897]

REFERENCE BOOK:

1. Richard Duda, Hart, David Stork, "Pattern Classification", John Wiley.[ISBN:0-471-05669-3]
2. M.Anji Reddy, Y.Hari Shankar "Digital Image Processing", BS Publications.[ISBN-13:9788178001227]

(CS208) NATURAL LANGUAGE PROCESSING (ELECTIVE-II)

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To acquire basic understanding of linguistic concepts and natural language complexity, variability.
2. To acquire basic understanding of machine learning techniques as applied to language.
3. To implement N-grams Models.
4. to understand language modeling
5. to learn about the CFG
6. to information extraction and retrieval

COURSE OUTCOMES:

1. Given an appropriate NLP problem, students should be able to select a corpus and an annotation scheme for the problem and justify the choice over other candidates.
2. Given one of the main linguistic issues relevant to NLP (including the representation and induction of syntactic knowledge, and the modelling of lexical and semantic information, and the syntax-semantics interface), students should be able to construct an example of the issue and provide an explanation of how their example illustrates the issue in general.
3. Given an example of one of the main linguistic issues identified above, students should be able to classify it as belonging to that issue and relate the example to the issue in general.
4. Given an NLP problem, students should be able to analyse, assess and justify which algorithms are most appropriate for solving the problem, based on an understanding of fundamental algorithms such as Viterbi algorithm, inside-outside, chart-based parsing and generation.

UNIT I

Introduction and Overview: What is Natural Language Processing, hands-on demonstrations. Ambiguity and uncertainty in language. The Turing test. **Regular Expressions:** Chomsky hierarchy, regular languages, and their limitations. Finite-state automata. Practical regular expressions for finding and counting language phenomena. A little morphology. Exploring a large corpus with regex tools. **Programming in Python:** An introduction to programming in Python. Variables, numbers, strings, arrays, dictionaries, conditionals, iteration. The NLTK (Natural Language Toolkit). **String Edit Distance and Alignment:** Key algorithmic tool: dynamic programming, a simple example, use in optimal alignment of sequences. String edit operations, edit distance, and examples of use in spelling correction, and machine translation.

UNIT II

Context Free Grammars: Constituency, CFG definition, use and limitations. Chomsky Normal Form. Top-down parsing, bottom-up parsing, and the problems with each. The desirability of combining evidence from both directions **Non-probabilistic Parsing** Efficient CFG parsing with CYK, another dynamic programming algorithms. Early parser. Designing a

little grammar, and parsing with it on some test data. **Probability** Introduction to probability theory Joint and conditional probability, marginals, independence, Bayes rule, combining evidence. Examples of applications in natural language. **Information Theory** The "Shannon game"--motivated by language! Entropy, crossentropy, information gain. Its application to some language phenomena.

UNIT III

Language modeling and Naive Bayes: Probabilistic language modeling and its applications. Markov models. N-grams. Estimating the probability of a word, and smoothing. Generative models of language. Part of Speech Tagging and Hidden Markov Models, Viterbi Algorithm for Finding Most Likely HMM Path Dynamic programming with Hidden Markov Models, and its use for part-of-speech tagging, Chinese word segmentation, prosody, information extraction, etc.

UNIT IV

Probabilistic Context Free Grammars: Weighted context free grammars. Weighted CYK. Pruning and beam search. **Parsing with PCFGs** A tree bank and what it takes to create one. The probabilistic version of CYK. Also: How do humans parse? Experiments with eye-tracking. Modern parsers. **Maximum Entropy Classifiers** The maximum entropy principle and its relation to maximum likelihood. Maximum entropy classifiers and their application to document classification, sentence segmentation, and other language tasks

UNIT V

Maximum Entropy Markov Models and Conditional Random Fields Part-of-speech tagging, noun-phrase segmentation and information extraction models that combine maximum entropy and finite-state machines. State-of-the-art models for NLP. **Lexical Semantics** Mathematics of Multinomial and Dirichlet distributions, Dirichlet as a smoothing for multinomial's. **Information Extraction** and deference Resolution- Various methods, including HMMs. Models of anaphora resolution. Machine learning methods for co reference.

TEXT BOOKS:

1. Jurafsky and Martin, "Speech and Language Processing", Prentice Hall[ISBN-13:978-0131873216]
2. Manning and Schutze, "Statistical Natural Language Processing" ,MIT Press[ISBN:0-262-13360-1]

REFERENCES BOOKS:

1. Cover, T.M. and J.A. Thomas "Elements of Information Theory", Wiley.[ISBN-13:978-0-471-24195-9]
2. James Allen, "Natural Language Understanding", The Benajmins/Cummings Publishing Company

(CS209) GAME THEORY

(ELECTIVE-II)

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. to train students in the logic and strategic decision making involved in the theory of game
2. To teach students to solve strategic games between two and more agents in noncooperative scenario.
3. Students will learn to analyze and solve both simultaneous-moves and sequential-moves games and will be familiarized with different solution concepts like minimax, Nash equilibrium etc.
4. to learn different methods to solve games, like backward induction, rationalizability, best responses, dominance and elimination of strategies, cellby-cell outcome examination, etc.
5. Applications of game theory in voting, bargaining, and collective action will make students recognize and understand game theory in the world around them.
6. Students will be expected to absorb and apply the concepts and ideas that constitute these various game types and their solutions, and apply them to the problems at hand.

UNIT I

Introduction

What is game theory?

An outline of the history of game theory. The theory of rational choice.

Introduction to Games in Strategic Form and Iterated Strict Dominance: Strategic-Form Games, Dominated Strategies, Applications of the Elimination of Dominated Strategies

UNIT II

Games with Perfect Information

Strategic games: examples

Nash Equilibrium: Theory, Concepts and examples.

Best response functions Dominated Actions

Equilibrium in a single population:symmetric games and symmetric equilibria

UNIT III

Nash Equilibrium : Illustrations

Cournot's model of oligopoly

Bertrand's model of oligopoly

Electoral competition

The War of Attrition Auctions Accident law

UNIT IV

Mixed Strategy Nash Equilibrium

Introduction

Strategic games with randomization

Mixed strategy Nash equilibrium: concept and examples

Dominated Actions

Formation of Players' beliefs

UNIT V

Extensive Games with Perfect Information

Introduction to extensive games

Strategies and outcomes

Nash equilibrium

Sub game perfect Nash equilibrium

Backward induction

Illustrations of Extensive Games and Nash Equilibrium:

Stackelberg model of duopoly markets

Ultimatum game.

TEXT BOOKS:

1. Games of Strategy, A. Dixit, S. Skeath & D. Reiley, 3rd edition (Norton & Co. 2009)
2. An Introduction to Game Theory, Osborne, Martin J. (2004), Oxford University Press.

REFERENCES BOOKS:

1. Game Theory, Fudenberg, D. and J. Tirole, 1991, MIT Press.
2. H. Peters; "Game theory: a multi-leveled approach"; 2008; Berlin, Springer.
3. J. Weibull, "Evolutionary Game Theory"; 1995; MIT Press.
4. Y. Narahari. Game Theory and Mechanism Design (March 2014). IISc Press and World Scientific.
5. A. Dixit and B. Nalebuff. *Thinking Strategically*, Norton 1991

**(CS210) SOFTWARE ARCHITECTURE AND DESIGN
PATTERNS
(ELECTIVE –II)**

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To understand the concept of patterns and the Catalog.
2. To discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
3. To understand the variety of implemented bad practices related to the Business and Integration tiers.
4. To highlight the evolution of patterns.
5. To how to add functionality to designs while minimizing complexity
6. To understand what design patterns really are, and are not

COURSE OUTCOMES:

The student will be able to

1. Argue the importance and role of software architecture in large scale software systems
2. Design and motivate software architecture for large scale software systems
3. Recognise major software architectural styles, design patterns, and frameworks
4. Discuss and evaluate the current trends and technologies such as model-driven, service-oriented, and aspect-oriented architectures and Evaluate the coming attractions in software architecture research and practice

UNIT I

Envisioning Architecture: The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views.**Creating an Architecture** Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT II

Analyzing Architectures Architecture Evaluation, Architecture design decision making, ATAM, CBAM. **Moving from one system to many** Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT III

Patterns Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage.

Creational and Structural patterns: Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight.

UNIT- IV

Behavioral patterns: Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

UNIT - V

Case Studies: A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development,

TEXT BOOKS:

1. Len Bass, Paul Clements & Rick Kazman, “Software Architecture in Practice”, second edition, Pearson Education, 2003.
2. Erich Gamma, “Design Patterns”, Pearson Education, 1995.

REFERENCE BOOKS:

1. David M. Dikel, David Kane and James R. Wilson, “Software architecture”, Prentice Hall PTR, 2001.
2. F.Buschmann & others, “Pattern Oriented Software Architecture”, John Wiley & Sons

(CS221) ADVANCED ALGORITHMS LAB

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	-	-	3	2	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. The fundamental design, analysis, and implementation of basic data structures.
2. Basic concepts in the specification and analysis of programs.
3. Principles for good program design, especially the uses of data abstraction.
4. to understand the sorting techniques
5. to understand the non linear data structures
6. to learn about the pattern matching

COURSE OUTCOMES:

1. Basic ability to analyze algorithms and to determine algorithm correctness and time Efficiency class.
2. Master a variety of advanced abstract data type (ADT) and data structures and their Implementations.
3. Master different algorithm design techniques (brute-force, divide and conquer, greedy, etc.)
4. Ability to apply and implement learned algorithm design techniques and data structures to solve problems

Sample Problems on Data structures:

1. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:
a) Linear search b) Binary search
2. Write Java programs to implement the following using arrays and linked lists
a) List ADT
3. Write Java programs to implement the following using an array.
a) Stack ADT b) Queue ADT
4. Write a Java program that reads an infix expression and converts the expression to postfix form. (Use stack ADT).
5. Write a Java program to implement circular queue ADT using an array.
6. Write a Java program that uses both a stack and a queue to test whether the given string is a palindrome or not.
7. Write Java programs to implement the following using a singly linked list.
a) Stack ADT b) Queue ADT
8. Write Java programs to implement the deque (double ended queue) ADT using
a) Array b) Singly linked list c) Doubly linked list.
9. Write a Java program to implement priority queue ADT.
10. Write a Java program to perform the following operations:
a) Construct a binary search tree of elements.
b) Search for a key element in the above binary search tree.
c) Delete an element from the above binary search tree.

11. Write a Java program to implement all the functions of a dictionary (ADT) using Hashing.
12. Write a Java program to implement Dijkstra's algorithm for Single source shortest path problem.
13. Write Java programs that use recursive and non-recursive functions to traverse the given binary tree in
 - a) Preorder b) Inorder c) Postorder.
14. Write Java programs for the implementation of bfs and dfs for a given graph.
15. Write Java programs for implementing the following sorting methods:
 - a) Bubble sort d) Merge sort g) Binary tree sort
 - b) Insertion sort e) Heap sort
 - c) Quick sort f) Radix sort
16. Write a Java program to perform the following operations:
 - a) Insertion into a B-tree b) Searching in a B-tree
17. Write a Java program that implements Kruskal's algorithm to generate minimum cost spanning tree.
18. Write a Java program that implements KMP algorithm for pattern matching.

REFERENCE BOOKS:

1. A.Drozdek "Data Structures and Algorithms in java", 3rd edition, Cengage Learning.
 2. J.R.Hubbard, "Data Structures with Java", 2nd edition, Schaum's Outlines, TMH.
- (Note: Use packages like java.io, java.util, etc)**

(CS222) DATA ANALYTICS LAB

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	-	-	3	2	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. Basic Concepts of data analysis and importance of Analytic Techniques.
2. Introduction Big data analytics and Tools
3. Installation of Hadoop and its Ecosystem
4. Hadoop Data Management
5. Performing Analytics on Hadoop

COURSE OUTCOMES:

1. Ability to select the appropriate technique for data analysis.
2. Understanding concepts of big data and different handling tools.
3. Ability to install hadoop and its ecosystem in physical and virtual environment.
4. Understanding hadoop ecosystem and usage for data analytics.
5. Performing data analysis using hadoop.

SAMPLE EXPERIMENTS:

1. Installation of hadoop framework(HDFS ,Map Reduce)on physical environment and creating hadoop cluster.
2. Performing HDFS related operations (loading data into HDFS, replicating data blocks, searching for Files in HDFS)
3. Performing MapReduce functions on hadoop cluster differentiating JobTrackers and Task Trackers.
4. Installation of Eclipse for MapReduce programming on hadoop cluster.
5. Executing mapreduce program to find the word count of a input file.
6. Executing Mapareduce program to find the maximum and minimum temperature from a given Weather data file.
7. Creating database table in HBASE and performing queries using NOSQL.
8. Pig demonstration and Data Analysis using Pig
9. Hive demonstration and Processing of a text using Hive of hadoop ecosystem.
10. Impala demonstration and Data processing using Impala
11. Cloudera latest version demonstration
12. Big Data Case Studies

REFERENCE BOOKS:

1. "[Hadoop Releases](http://apache.org)". apache.org. Apache Software Foundation. Retrieved 2014-12-06.
2. "[Hadoop Releases](http://Hadoop.apache.org)". Hadoop.apache.org. Retrieved 2015-07-29.
3. "[Welcome to Apache™ Hadoop®!](http://hadoop.apache.org)". hadoop.apache.org. Retrieved 2015-09-20.
4. "[What is the Hadoop Distributed File System \(HDFS\)?](http://ibm.com)". ibm.com. [IBM](http://ibm.com). Retrieved 2014-10-30.
5. Malak, Michael (2014-09-19). "[Data Locality: HPC vs. Hadoop vs. Spark](http://datascienceassn.org)". datascienceassn.org. Data Science Association. Retrieved 2014-10-30.

6. ["Resource \(Apache Hadoop Main 2.5.1 API\)".](#) apache.org. Apache Software Foundation. 2014-09-12. Retrieved 2014-09-30.
7. Murthy, Arun (2012-08-15). ["Apache Hadoop YARN – Concepts and Applications".](#)hortonworks.com. Hortonworks. Retrieved 2014-09-30.
8. ["Continuity Raises \\$10 Million Series A Round to Ignite Big Data Application Development Within the Hadoop Ecosystem".](#) finance.yahoo.com. [Marketwired.](#) 2012-11-14. Retrieved 2014-10-30.
9. ["Hadoop-related projects at".](#) Hadoop.apache.org. Retrieved 2013-10-17.
10. [Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data.](#) John Wiley & Sons. 2014-12-19. p. 300. [ISBN 9781118876220.](#) Retrieved 2015-01-29.

(CS211)ADVANCED NETWORK PROGRAMMING

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. Computer network programming involves writing computer programs that enable processes to communicate with each other across a computer network
2. Network programming to build effective client-server systems
3. To know about the Interprocess communication
4. To learn about peer-to-peer communication, the program can act both as a client and a server.
5. to learn about the networking programming
6. to understand about client - server applications

COURSE OUTCOMES:

1. To use Java Application Programming Interface and windowing toolkits (AWT, Swing)
2. To design GUI clients for network servers
3. To develop concurrent programs with threads, in particular, multithreaded servers
4. To develop and design distributed applications with sockets, JSF, EJB.

UNIT – I

Linux Utilities- File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking utilities, Filters, Text processing utilities and Backup utilities. Bourne again shell(bash) - Introduction, pipes and redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples. Review of C programming concepts-arrays, strings (library functions), pointers, function pointers, structures, unions, libraries in C.

UNIT - II

Files- File Concept, File types File System Structure, Inodes, File Attributes, file I/O in C using system calls, kernel support for files, file status information-stat family, file and record locking-lockf and fcntl functions, file permissions- chmod, fchmod, file ownership-chown, lchown , fchown, linksoft links and hard links – symlink, link, unlink. File and Directory management – Directory contents, Scanning Directories- Directory file APIs. Process-Process concept, Kernel support for process, process attributes, process control – process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process.

UNIT - III

Signals- Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions. Interprocess Communication - Introduction to IPC mechanisms, Pipes- creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs(Named pipes), differences between unnamed and named pipes, popen and pclose library functions, Introduction to message queues, semaphores and shared memory.Message Queues- Kernel support for messages, UNIX system V APIs for messages, client/server example. Semaphores-Kernel support for semaphores, UNIX system V APIs for semaphores.

UNIT – IV

Shared Memory- Kernel support for shared memory, UNIX system V APIs for shared memory, client/server example. Network IPC - Introduction to Unix Sockets, IPC over a network, Client-Server model ,Address formats(Unix domain and Internet domain), Socket system calls for Connection Oriented -Communication, Socket system calls for Connectionless-Communication, Example-Client/Server Programs- Single Server-Client connection, Multiple simultaneous clients, Socket options – setsockopt, getsockopt,fcntl.

UNIT-V

Network Programming in Java-Network basics, TCP sockets, UDP sockets (datagram sockets), Server programs that can handle one connection at a time and multiple connections (using multithreaded server), Remote Method Invocation (Java RMI)-Basic RMI Process, Implementation details-Client-Server Application.

TEXT BOOKS:

1. T.Chan, “Unix System Programming using C++”, PHI.
2. Sumitabha Das, “Unix Concepts and Applications”, 4th Edition, TMH.

REFERENCE BOOKS:

1. Robert Love, “Linux System Programming”, O’Reilly, SPD.
2. Jan Graba, “An Introduction to Network Programming with Java”, Springer, rp 2010.

(CS212) HIGH PERFORMANCE COMPUTING

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To know how modern high performance processors are organized their strengths and weaknesses.
2. To study about the architecture of parallel systems.
3. To gain depth knowledge about the analytical parallel algorithms.
4. Understand hardware design of modern HPC platforms and the parallel programming models that they support
5. Understand the mapping of applications to high-performance computing systems

COURSE OUTCOMES:

1. To design architectures to perform parallel processing.
2. To design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.
3. To optimize the performance of a parallel program to suit a particular hardware and software environment.
4. To analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

UNIT – I

Introduction

Need of high speed computing – increase the speed of computers – history of parallel computers and recent parallel computers; solving problems in parallel –temporal parallelism– data parallelism–comparison oftemporal and data parallel processing – data parallel processing with specialized processors inter-task dependency.

The need for parallel computers - models of computation - analyzing algorithms -expressing algorithms.

UNIT - II

Parallel Programming Platforms: Trends in microprocessor architectures - imitations of memory system performance – parallel computing platforms – communication costs in parallel machines – routing mechanisms for interconnection networks.

Principles of Parallel Algorithm Design: Preliminaries– decomposition techniques – characteristics of tasks and interactions – mapping techniques for load balancing – methods for containing interaction overheads – parallel algorithm models.

Basic Communication Operations: One-to-all broadcast and all-to-one education – all-to-all broadcast reduction – all-reduce and prefix-sum operations – scatter and gather – all-to-all personalized communication – circular shift – improving the speed of some communication operations

UNIT - III

Analytical Modeling of Parallel Programs: Sources of overhead in parallel programs – performance metrics for parallel systems – scalability of parallel systems – minimum execution time and minimum cost-optimal execution time

Programming using the Message-Passing Paradigm: principles of Message - assign programming – the building blocks – MP I – topologies and embedding – overlapping communication with computation – collective communication and computation operations – groups and communicators.

Programming Shared Address Space Platforms: Thread basics – synchronization primitives in P threads – controlling thread and synchronization attributes – composite synchronization constructs – tips for designing asynchronous programs – Open MP

UNIT – IV

Dense Matrix Algorithms: Matrix-vector multiplication – matrix-matrix multiplication – solving a system of linear equations – FFT.

Sorting: Issues in sorting on parallel computers – sorting networks – bubble sort and its variants – Quick sort – bucket and sample sort – other sorting algorithms

Graph Algorithms: Definitions and representation – minimum spanning tree single-source shortest paths – all-pairs shortest paths

UNIT-V

Search Algorithms for Discrete for Discrete Optimization Problems Definitions and examples, sequential search algorithms, search over head f actor, parallel depth-first search, parallel best-first search, speedup anomalies in parallel search algorithms.

Dynamic Programming: Overview

TEXT BOOKS:

1. Ananth Grama, Anshul gupta, George Karypis and Vipin Kumar, “Introduction to Parallel Computing”, Pearson Education, Second edition, 2004.
2. V. Rajaraman and C. Siva RamMurthy, “Parallel Computers – Architecture and Programming”, Prentice-Hall of India, 2003.

REFERENCE BOOKS:

1. SelimG. Akl, “The Design and Analysis of Parallel Algorithms”, Prentice-Hall of India, 1999. (UNITs I, IV & V - as Reference)
2. M.J. Quinn, “Parallel Computing – Theory and Practice”, McGraw-Hill, 1994
3. Michael Jay Quinn, “Parallel Programming in C with MPI and OpenMP”, McGraw-Hill, 2003

(CS213) WEB SERVICES AND SERVICE ORIENTED ARCHITECTURE

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To Understand Web Services and implementation model for SOA
2. To Understand the SOA, its Principles and Benefits, XML concepts
3. To Understand paradigms needed for testing Web Services
4. To explore different Test Strategies for SOA-based applications
5. To implement functional testing, compliance testing and load testing of Web Services
6. To Identify bug-finding ideas in testing Web Services

COURSE OUTCOMES:

1. Understand and describe the principles of service oriented architecture
2. Understand and describe the standards and technologies of modern web services implementations
3. Effectively use market-leading development tools to create and consume web services
4. Analyze the requirements of a medium-difficulty programming task, and create software that meets the requirements

UNIT - I

Evolution and Emergence of Web Services – Evolution of distributed computing. Core distributed computing technologies – client/server, CORBA, JAVA RMI, Micro Soft DCOM, MOM, Challenges in Distributed Computing, role of J2EE and XML in distributed computing, emergence of Web Services and Service Oriented Architecture (SOA). Introduction to Web Services – The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services.

UNIT - II

Web Service Architecture – Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication, basic steps of implementing web services. Describing Web Services – WSDL introduction, non functional service description, WSDL1.1 Vs WSDL 2.0, WSDL document, WSDL elements, WSDL binding, WSDL tools, WSDL port type, limitations of WSDL.

UNIT - III

Brief Over View of XML – XML Document structure, XML namespaces, Defining structure in XML documents, Reuse of XML schemes, Document navigation and transformation. SOAP : Simple Object Access Protocol, Inter-application communication and wire protocols, SOAP as a messaging protocol, Structure of a SOAP message, SOAP envelope, Encoding, Service Oriented Architectures, SOA revisited, Service roles in a SOA, Reliable messaging, The enterprise Service Bus, SOA Development Lifecycle, SOAP HTTP binding, SOAP communication model, Error handling in SOAP.

UNIT - IV

Registering and Discovering Services : The role of service registries, Service discovery, Universal Description, Discovery, and Integration, UDDI Architecture, UDDI Data Model, Interfaces, UDDI Implementation, UDDI with WSDL, UDDI specification, Service Addressing and Notification, Referencing and addressing Web Services, Web Services Notification.

UNIT- V

SOA and web services security considerations, Network-level security mechanisms, Application-level security topologies, XML security standards, Semantics and Web Services, The semantic interoperability problem, The role of metadata, Service metadata, Overview of .NET and J2EE, SOA and Web Service Management, Managing Distributed System, Enterprise management Framework, Standard distributed management frameworks, Web service management, Richer schema languages, WS-Metadata Exchange.

TEXT BOOKS:

1. Michael P. Papazoglou, "Web Services & SOA Principles and Technology", Second Edition.
2. R. Nagappan, R. Skoczylas, R.P. Sriganesh "Developing Java Web Services", Wiley India.

REFERENCE BOOKS:

1. S. Chatterjee, J. Webber, "Developing Enterprise Web Services", Pearson Education.
2. S. Graham and others, "Building web Services with Java", 2nd Edition, Pearson Education.

(CS214) NUMBER THEORY & CRYPTOGRAPHY

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. Understand the basic concepts of number theory
2. Communicate and solve problems mathematically
3. Understand the basic methods of secure communication and cryptanalysis,
4. Understand the concepts of Public vs. Private Key,
5. Have a basic knowledge of the current cryptographic protocols and the issues involved.

COURSE OUTCOMES:

1. Choose a crypt method for secure communication
2. Have sound foundations of number theory for designing secure crypt systems
3. Apply elliptic curve cryptography as a modern crypt methodology
4. Analyze crypt systems.

UNIT I

Divisibility Theory in the Integers, Primes and Their Distribution, The Theory of Congruence's, Fermats Theorem. Number-Theoretic Functions, Eulers Generalization of Fermats Theorem, Primitive Roots and Indices, The Quadratic Reciprocity Law.

UNIT II

Fibonacci Numbers: Fibonacci, The Fibonacci Sequence, Certain Identities Involving Fibonacci Numbers Continued Fractions: Srinivasa Ramanujan, Finite Continued Fractions, Infinite Continued Fractions Farey Fractions, Pells Equation.

UNIT III

Introduction to Cryptography: From Caesar Cipher to Public Key Cryptography, The Knapsack Cryptosystem An Application of Primitive Roots to Cryptography.

UNIT IV

Numbers of Special Form: Marin Mersenne, Perfect Numbers, Mersenne Primes and Amicable Numbers, Fermat Numbers. Certain Nonlinear Diophantine Equations: The Equation, Fermats Last Theorem Representation of Integers as Sums of Squares: Joseph Louis Lagrange Sums of Two Squares, Sums of More Than Two Squares.

UNIT V

Introduction to the arithmetic of elliptic curves. Elliptic curve cryptography & applications

TEXT BOOKS:

1. [James S. Kraft](#) (Author), [Lawrence C. Washington](#) (Author) “An Introduction to Number Theory with Cryptography” CRC Press. ISBN: 978-1482214413
2. David M Burton “Elementary Number Theory” 7th Edition, McGraw Hill Education, ISBN: 9781259025761

REFERENCE BOOKS:

1. [Lawrence C. Washington](#) “Elliptic Curves: Number Theory and Cryptography”, Second Edition, CRC Press. ISBN:978-142007146
2. [Fernando Gouvea](#) “p-adic Numbers: An Introduction” 2nd Edition, Springer, ISBN: 978-3540629115
3. Wenbo Mao “Modern Cryptography : Theory and Practice”, Pearson Publishers, ISBN: 813170212X
4. Neil Koblitz “A Course In Number Theory And Cryptography” Springer, ISBN: 9788181282309
5. [Song Y. Yan](#) “Computational Number Theory and Modern Cryptography” Wiley Publishers,ISBN: 978-1-118-18858-3

(CS215) ADHOC AND SENSOR NETWORKS

(Elective-III)

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To understand the fundamentals of ADHOC Networks.
2. To know the Transport Protocols of ADHOC and sensor Networks
3. To understand the Wireless Sensor Networks.
4. To understand the communication protocols of wireless system.
5. To know about the QoS of ADHOC Networks.
6. To learn about the design issues of wireless sensor networks.

COURSE OUTCOMES:

1. Apply the basic fundamentals to design ADHOC wireless system.
2. Develop and design security protocols of ADHOC wireless network.
3. Model an architecture for wireless sensor networks
4. Model and design WSNS and QoS of wireless sensor networking systems.

UNIT - I

Ad Hoc Wireless: Introduction, Mobile Ad Hoc Networks, Technologies for Ad Hoc Network, Issues in Ad hoc wireless Networks, IEEE 802.11 Architecture and protocols. Protocol for AD HOC Wireless Networks Issues and classification of MAC protocol, other MAC protocols, Dynamic Source Routing (DBR), Adhoc Distance Vector (AoDV) routing, Routing Protocols, Multicasting Routing issues

UNIT –II

Transport layer and Security protocols Issues in designing transport layer protocols, TCP over Ad Hoc Wireless Networks, Network Security Attacks, and Key management.

UNIT - III

Wire Sensor Networks Basic Sensor Network Architectural Elements, Applications of Sensor Networks, Comparison with Ad Hoc Wireless Networks, Challenges and Hurdles. Architecture of WSNs Hardware components, Operating systems and execution environments, some examples of sensor nodes, Network Architecture, Sensor networks scenarios, Optimization goals and figures of merit, Design principles for WSNs.

UNIT –IV

Communication Protocols Physical Layer and Transceiver design considerations in WSNs, Fundamentals of (wireless) MAC protocol, Address and name management in wireless sensor networks, Localization and positioning

UNIT –V

Routing protocols Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless. Routing Strategies in Wireless Sensor Networks, QoS in wireless sensor networks, Coverage and deployment.

TEXT BOOKS:

1. C Siva Ram Murty and BS Manoj, “Ad HOC Wireless Networks: Architectures and Protocols “, 2nd Ed, Pearson Education,ISBN:8131706885
2. Adleshein and Gupta, “Fundamentals of Mobile and Pervasive Computing”, TMH, 2005, ISBN:978-0-7645-4887-1

REFERENCE BOOKS:

1. Mohamed Illayas, “Handbook of Ad Hoc wireless network”, CRC press,ISBN:0-8493-1332-5
2. Holger Karl, “Protocols and Architectures for Wireless Sensor Networks”, John Wiley and Sons, ISBN:0470519231
3. Kazem Sohraby, Daniel Minoli, Taieb Znati, “Wireless Sensor Networks Technology, Protocols, and applications”, John Wiley and Sons,ISBN:978-0-471-74300-2

WEB LINKS:

1. <http://nptel.iitm.ac.in>
2. www.pearsonhighered.com > ... > Wireless Communications

(CS216) CLOUD COMPUTING

(ELECTIVE-III)

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100

Pre-requisite	
----------------------	--

Program: M.Tech
Year: I
Sem : II
CIE: 40
SEE: 60

Lecture : 4/Week
Tutorial : -
Drawing : -
Credits : 4

COURSE OBJECTIVES:

1. To aware about computer networks and Operating Systems
2. To learn about shared resources to be provisioned over the network as services in an on-demand Manner.
3. To know about computing model, which enables information, software
4. To know about various services and technologies
5. To learn about architecture for cloud computing
6. To know about the legal issues in cloud computing

COURSE OUTCOMES:

1. Describe the hardware and software concepts and architecture of cloud computing.
2. Be able to evaluate Infrastructure as a Service (IAAS) & Platform and Software as a Service (PAAS / SAAS) for building computer clusters.
3. Be able to utilise clustering middleware, such as parallel virtual machine (pvm), message passing interface (mpi), hpc portals, and peer-to-peer networks for implementing virtual super computing resources
4. Contrast the key technical and commercial issues concerning cloud computing versus traditional software models.

UNIT I

Systems Modeling, Clustering and Virtualization: Distributed System Models and Enabling Technologies, Computer Clusters for Scalable Parallel Computing, Virtual Machines and Virtualization of Clusters and Data centers.

UNIT II

Foundations: Introduction to Cloud Computing, Migrating into a Cloud, Enriching the 'Integration as a Service' Paradigm for the Cloud Era, the Enterprise Cloud Computing Paradigm.

UNIT III

Infrastructure as a Service (IAAS) & Platform and Software as a Service (PAAS / SAAS):

Virtual machines provisioning and Migration services, On the Management of Virtual machines for Cloud Infrastructures, Enhancing Cloud Computing Environments using a cluster as a Service, Secure Distributed Data Storage in Cloud Computing. Aneka, Comet Cloud, T-Systems, Workflow Engine for Clouds, Understanding Scientific Applications for Cloud Environments.

UNIT IV

Monitoring, Management and Applications : An Architecture for Federated Cloud Computing, SLA Management in Cloud Computing, Performance Prediction for HPC on Clouds, Best Practices in architecting Cloud Applications in the AWS cloud, Building Content Delivery networks using Clouds, Resource Cloud Mashups.

UNIT V

Governance and Case Studies: Organizational Readiness and Change management in the Cloud age, data Security in the Cloud, Legal Issues in Cloud computing, Achieving Production Readiness for Cloud Services.

TEXT BOOKS:

1. Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms” Wiley, 2011.
2. Kai Hwang, Geoffery C.Fox, Jack J.Dongarra, “Distributed and Cloud Computing”, Elsevier, 2012.

REFERENCE BOOKS:

1. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, Tata McGraw Hill, rp2011.
2. Gautam Shroff, “Enterprise Cloud Computing”, Cambridge University Press, 2010.

(CS217) DATABASE SECURITY

ELECTIVE-III

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To learn the security of databases
2. To learn the design techniques of database security
3. To learn the secure software design
4. To know about stastical DB Protection
5. To understand the IDS
6. To know about the models and protection of new generation DBs

COURSE OUTCOMES:

1. Carry out a risk analysis for a large database.
2. Implement identification and authentication procedures, fine-grained access control and data encryption techniques.
3. Set up accounts with privileges and roles.
4. Audit accounts and the database system.
5. Back-up and Restore a database.

UNIT I

Introduction: Introduction to Databases Security Problems in Databases Security Controls Conclusions **Security Models -1** Introduction Access Matrix Model Take-Grant Model Acten Model PN Model Hartson and Hsiao's Model Fernandez's Model Bussolati and Martella's Model for Distributed databases

UNIT II

Security Models -2: Bell and LaPadula's Model Biba's Model Dion's Model Sea View Model Jajodia and Sandhu's Model The Lattice Model for the Flow Control conclusion. **Security Mechanisms** Introduction User Identification/Authentication Memory Protection Resource Protection Control Flow Mechanisms Isolation Security Functionalities in Some Operating Systems Trusted Computer System Evaluation Criteria

UNIT III

Security Software Design: Introduction A Methodological Approach to Security Software Design Secure Operating System Design Secure DBMS Design Security Packages Database Security Design

UNIT IV

Statistical Database Protection & Intrusion Detection Systems: Introduction Statistics Concepts and Definitions Types of Attacks Inference Controls evaluation Criteria for Control Comparison .Introduction IDES System RETISS System ASES System Discovery

UNIT V

Models for The Protection Of New Generation Database Systems -1: Introduction A Model for the Protection of Frame Based Systems A Model for the Protection of Object-Oriented Systems SORION Model for the Protection of Object-Oriented Databases **Models For The Protection Of New Generation Database Systems -2** A Model for the Protection of New Generation Database Systems: the Orion Model Jajodia and Kogan's Model A Model for the Protection of Active Databases Conclusions.

TEXT BOOKS:

1. Hassan A. Afyouni “Database Security and Auditing”, India Edition, CENGAGE Learning, 2009.
2. Castano, “Database Security”, Second edition, Pearson Education.

REFERENCE BOOK:

1. Alfred basta, melissa zgola, “Database security” CENGAGE learning.

(CS218) SEMANTIC WEB AND SOCIAL NETWORKS

ELECTIVE-IV

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

Program: M.Tech
Year: I
Sem : II
CIE: 40
SEE: 60

Lecture : 4/Week
Tutorial : -
Drawing : -
Credits : 4

COURSE OBJECTIVES:

1. To learn Web Intelligence
2. To learn Knowledge Representation for the Semantic Web
3. To learn Ontology Engineering
4. To learn Semantic Web Applications, Services and Technology
5. To learn Social Network Analysis and semantic web
6. To know about the different social networking technologies

COURSE OUTCOMES:

1. Describe the web intelligence and Knowledge Representation for the Semantic Web.
2. Discuss the principles of Social Network Analysis and semantic web.
3. Ability to work on Ontology Engineering.
4. Explain the problems and solutions introduced by Social Network Analysis and semantic web.

UNIT –I

Web Intelligence Thinking and Intelligent Web Applications, The Information Age ,The World Wide Web, Limitations of Today’s Web, The Next Generation Web, Machine Intelligence, Artificial Intelligence, Ontology, Inference engines, Software Agents, Berners-Lee www, Semantic Road Map, Logic on the semantic Web.

UNIT -II

Knowledge Representation for the Semantic Web Ontologies and their role in the semantic web, Ontologies Languages for the Semantic Web –Resource Description Framework(RDF) / RDF Schema, Ontology Web Language(OWL), UML,XML/XML Schema.

UNIT-III

Ontology Engineering Ontology Engineering, Constructing Ontology, Ontology Development Tools, Ontology Methods, Ontology Sharing and Merging, Ontology Libraries and Ontology Mapping, Logic, Rule and Inference Engines.

UNIT-IV

Semantic Web Applications, Services and Technology Semantic Web applications and services, Semantic Search, e-learning, Semantic Bioinformatics, Knowledge Base ,XML Based Web Services, Creating an OWL-S Ontology for Web Services, Semantic Search Technology, Web Search Agents and Semantic Methods

UNIT-V

Social Network Analysis and semantic web What is social Networks analysis, development of the social networks analysis, Electronic Sources for Network Analysis – Electronic Discussion networks, Blogs and Online Communities, Web Based Networks. Building Semantic Web Applications with social network features.

TEXT BOOKS:

1. Berners Lee, Godel and Turing, “Thinking on the Web” ,Wiley inter science, 2008.
2. Peter Mika, “Social Networks and the Semantic Web”, Springer, 2007.

REFERENCE BOOKS:

1. “J.Davies, R.Studer, P.Warren, Semantic Web Technologies, Trends and Research in Ontology Based Systems”, John Wiley & Sons.
2. Liyang Lu Chapman and Hall/CRC Publishers, (Taylor & Francis Group), “Semantic Web and Semantic Web Services”.

(CS219) SOFTWARE QUALITY ASSURANCE & TESTING

ELECTIVE-IV

Program: M.Tech
Year: I
Sem : II
CIE: 40
SEE: 60

Lecture : 4/Week
Tutorial : -
Drawing : -
Credits : 4

COURSE OBJECTIVES:

1. To understand software testing and quality assurance as a fundamental component of software life cycle
2. To define the scope of SW T&QA projects
3. To efficiently perform T&QA activities using modern software tools
4. To estimate cost of a T&QA project and manage budgets
5. To prepare test plans and schedules for a T&QA project
6. To develop T&QA project staffing requirements

COURSE OUTCOMES:

1. Argue the importance and role of software quality assurance and testing in large scale software systems
2. Design and motivate software testing and quality assurance in large.
3. Recognize major software test plans and schedules for a T&QA project
4. Discuss and evaluate the T&QA project staffing requirements

UNIT I

Software Quality Assurance and Standards: The Software Quality challenge, What is Software Quality, Software Quality factors, The components of Software Quality Assurance system, Software Quality Metrics, Costs of Software Quality, Quality Management Standards, Management and its role in Software Quality Assurance, SQA unit and other actors in SQA system. **Quality Standards:** ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma and other latest quality standards.

UNIT II

Software Testing Strategy and Environment: Minimizing Risks, Writing a Policy for Software Testing, Economics of Testing, Testing-an organizational issue, Management Support for Software Testing, Building a Structured Approach to Software Testing, Developing a Test Strategy **Building Software Testing Process:** Software Testing Guidelines, workbench concept, Customizing the Software Testing Process, Process Preparation checklist **Software Testing Techniques:** Dynamic Testing – Black Box testing techniques, White Box testing techniques, Static testing, Validation Activities, Regression testing

UNIT III

Software Testing Tools Selecting and Installing Software Testing tools Automation and Testing Tools Load Runner, Win runner and Rational Testing Tools, Silk test, Java Testing Tools, JMetra, JUNIT and Cactus.

UNIT IV

Testing Process Seven Step Testing Process – I: Overview of the Software Testing Process, Organizing of Testing, Developing the Test Plan, Verification Testing, Validation Testing.

UNIT V

Seven Step Testing Process – II: Analyzing and Reporting Test results, Acceptance and Operational Testing, Post-Implementation Analysis **Specialized Testing Responsibilities:** Software Development Methodologies, Testing Client/Server Systems

TEXT BOOKS:

1. William E. Perry, "Effective Methods for Software Testing", Third edition, Wiley India, 2009 [ISBN:978-0-7645-9837-1]
2. Naresh Chauhan, "Software Testing – Principles and Practices", Oxford University Press, 2010.[ISBN:97080198061847]

REFERENCE BOOKS:

1. Daniel Galin, "Software Quality Assurance – From Theory to Implementation", Pearson Edn,2009.[ISBN:978-81-317-2395-1]
2. M.G.Limaye, "Software Testing – Principles, Techniques and Tools", Tata McGraw-Hill, 2009.[ISBN:0070139903]

(CS220) COMPUTATIONAL ASTRONOMY

ELECTIVE-IV

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

Program: M.Tech
Year: I
Sem : II
CIE: 40
SEE: 60

Lecture : 4/Week
Tutorial : -
Drawing : -
Credits : 4

COURSE OBJECTIVES:

This course will enable the students

1. To Appreciate the majesty of skies & take a scientific look.
2. To explore computer graphics & animation skills
3. To learn java through astronomy and create applications
4. To Explore Jet Propulsion Lab data

COURSE OUTCOMES:

The student will be able to

1. Get insights into the domain of basic astronomy
2. Sharpen programming skills in Java through graphics & animation progra
3. Have proficiency in Computer graphics
4. Comprehend geometrical formulae for astronomy

UNIT I

Introduction to astronomy, The Earth's globe, Distance conversions, Transformation of coordinates, The parallactic angle, Planetary conjunctions, Equation of Time, Apparent place of stars, Nutation and the Obliquity of the Ecliptic. Solar coordinates, Position of the moon, passages of the moon, Eclipses.

UNIT II

Introduction to 3D graphics, rotation, reflection, 3D rendering on 2D, java APIs for 3D graphics, static image of a shaded sphere, surface details of a sphere

UNIT III

Keplers's laws of planetary motion, Kepler's equation, Equation of time, Geometric formulas used in astronomy & implementations in java. JPL data & format conversions, data-jockeying.

UNIT IV

Exploring JPL data for different animations of the solar system. View of the sky- perspectives from earth with respect to time and space, animations. Stars and motion of stars an animation study.

UNIT V

Useful Astronomical aids-star maps, star catalogues, star charts, star rising & star Setting times, Ephemerides & Almanacs, Astronomical calendars,

TEXT BOOKS:

1. Jeen Meeus, "Astronomical Algorithms" Willmann-Bell, Inc
2. Randall S. Fairman "3D Astronomy with Java, An introduction to Computer Graphics", Willmann-Bell, Inc, Publishers, ISBN: 9780943396934

REFERENCE BOOKS:

1. Lawrence "Introduction to basic Astronomy with a PC A Practical Introduction to Astronomical Computing" Willmann-Bell, Inc Publishers
2. Jean Meeus "Astronomical Formulae for Calculators" Willmann-Bell, Inc, Publishers
3. Simon Portegies Zwart "Computational Astrophysics & Cosmology" Springer ISSN:2197-7909

WEB LINKS:

1. <http://www.jpl.nasa.gov/>
2. <http://academicearth.org/astronomy/>
3. <https://www.coursera.org/course/introastro>

(CS223) HIGH PERFORMANCE COMPUTING LAB

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

COURSE OBJECTIVES:

1. To introduce the high performance domain on computing.
2. To extend C/C++ programming skills on extended hard ware
3. To introduce Compute Unified Device Architecture.
4. To familiarize with openCL programming.
5. To appreciate computing with graphic processing unit.

COURSE OUTCOMES:

The student will be able to:

1. Apply C/C++ skills in GPU computing.
2. Comprehend the advanced computing features with extended hard wares.
3. Recognize the “power wall” limitations of CPU computing.
4. Solve scientific problems in data intensive and high speed requirements.

Week -1:

Preparing eclipse for openCL development

Week -2:

Installation of openCL in Windows 7 and configuration in visual studio

Week -3:

OpenCL GPU Matrix multiplication.

Week -4:

OpenCL vector addition

Week -5:

OpenCL Dot product

Week -6:

Parallel pre fix sum using OpenCL

Week -7:

OpenCL Matrix Transpose

Week -8:

OpenCL Quasi random generation

Week -9:

N-body problem using OpenCL

Week -10:

Particle collision simulation

Week -11:

CUDA API's – GPUs

Week -12:

CUDA on Hadoop - GPUs

(CS224) NETWORK PROGRAMMING LAB

Year	Semester	Hours / Week			C	Marks		
		L	T	P/D		CIE	SEE	Total
I	I	4	-	-	4	40	60	100
Pre-requisite								

Program: M.Tech

Year: I

Sem : II

CIE: 40

SEE: 60

Lecture : -

Tutorial : -

Practical : 3/Week

Credits : 2

Course Objectives:

The students will be able to:

1. Implement the IPC Programmes
2. Implement the Shared Memory Concepts
3. Design and Implement the TCP Client/Server Programme
4. Design and Implement the UDP Client/Server Programme

Recommended Systems/Software Requirements:

- Intel based desktop PC with minimum of 166 MHZ or faster processor with atleast 64 MB RAM and 100 MB free disk space LAN Connected
- Any flavour of Unix / Linux

Week -1:

Implement the following forms of IPC.

a)Pipes b)FIFO

Week- 2:

Implement file transfer using Message Queue form of IPC

Week 3:

Write a programme to create an integer variable using shared memory concept and increment the variable simultaneously by two processes. Use semaphores to avoid race conditions

Week- 4:

Design TCP iterative Client and server application to reverse the given input sentence

Week- 5:

Design TCP iterative Client and server application to count the number of vowels present in given input sentence

Week 6:

Design TCP client and server application to transfer file

Week- 7:

Design a TCP concurrent server to convert a given text into upper case using multiplexing system call “select”

Week- 8:

Design a TCP concurrent server to echo given set of sentences using poll functions

Week -9:

Design UDP Client and server application to reverse the given input sentence

Week -10:

Design UDP Client server to transfer a file

Week- 11:

Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.

Week -12:

Design a RPC application to add and subtract a given pair of integers

REFERENCE BOOKS:

1. Advance Unix Programming Richard Stevens, Second Edition Pearson Education
2. Advance Unix Programming, N.B. Venkateswarlu, BS Publication.

WEB LINKS:

1. np-material.blogspot.com/.../unix-network-programming-1st-Edition..
<http://thebookslibrary.blogspot.in/2008/12/unix-network-programming-vol-i-sockets.html>