SEMESTER I

CS420: Industrial Training /Mining Project /Technical Training (CR-Audit, L-0, P-0)

Students are supposed to submit the detail report covering the aspects related to computer engineering projects that are relevant to industry in which they receive training.

CS421: Computer Graphics (CR-3, L-3, P-0)

Prerequisites: Data Structures

Course objective:

- Students will demonstrate an understanding of contemporary graphics hardware.
- Students will create interactive graphics applications in C++ using one or more graphics application programming interfaces.
- Students will write program functions to implement graphics primitives.
- Students will write programs that demonstrate geometrical transformations.
- Students will demonstrate an understanding of the use of object hierarchy in graphics applications.
- Students will write program functions to implement visibility detection.
- Students will write programs that demonstrate computer graphics animation.
- Students will write programs that demonstrate 2D image processing techniques

Course outline:

Introduction to Computer Graphics: Overview of computer graphics, graphics displays, output devices and physical interactive devices, graphical user interfaces, Graphics image file formats.
Raster Scan Graphics: Line drawing algorithms, DDA, Bresenham's algorithm, circle generations, scan conversion-generation of displays, image compression, displaying lines, characters and polygons, polygon filling algorithms, fundamentals of antialiasing, halftoning
Geometrical Transformations: 2-D Transformations, Linear Transformations, other transformations, combined Transformation, Coordinate systems, 3-D Transformations, rotation, scaling and translation, reflection about any arbitrary axis.
Windowing and Clipping: Viewing transformations, Parallel projections, Perspective Projection, Perspective Transform, Two dimensional clipping, simple visibility algorithm, polygon clipping, 3-Dimensional clipping
Hidden surface elimination: Floating horizon, Back face removal algorithms, Z-Buffer algorithm, painter's algorithm, Warnock algorithm, BSP Tree methods
Rendering: Introduction, illumination models, transparency, shadows, phongs and gouraud shading
Curve Design: Properties of curves, Bezier and b-splines.

References:

- David f. Rogers, "Procedural elements of computer graphics", TMH.
- Foley, Van dam, feiner hughes, "Computer graphics principles and practice", Addison Wesley Indian Edition.
- Newman sproull, "Principles of Interactive computer graphics", McGraw Hill Company.

CS422: Advanced Database Management Systems (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to:

- Write complex queries, including full outer joins, self-joins, sub queries, and set theoretic queries.
- Write stored procedures and triggers.
- Apply the principles of query optimization to a database schema.
- Explain the various types of locking mechanisms utilized within database management systems.
- Explain the different types of database failures as well as the methods used to recover from these failures.
- Design queries against a distributed database management system.
- Perform queries against database designed with object-relational extensions.
- Explain the Database Security and Authorization.
- Describe the design of Distributed Databases.
- Describe the basic concept of Data warehousing and Data mining.
- Discuss the emerging Database Models Technologies and Applications.
- Develop and query XML files.

Course outline:

Distributed Databases: Introduction, Promises of DDBSs, Complicating factors, problem areas of DDBSs, Architectural models for Distributed DBMS, Distributed DBMS architecture, Distributed database Design: Alternative Design Strategies, Distribution Design issues.
Distributed Query Processing: Query processing problem, objectives of Query processing, Complexity of Relational Algebra operation, Characterization of Query processors, Layers of Query processing. Distributed Transactions**,** Commit protocols, Concurrency control in Distributed Databases, Failures and fault Tolerance in Distributed databases.
Parallel Databases: Database servers, Parallel architectures, parallel DBMS techniques, parallel execution problems, parallel execution for Hierarchical architecture.
Application development and administration: Web interfaces to databases, performance tuning, performance benchmarks, standardization, e-commerce, and legacy systems
Advanced Querying and Information Retrieval: Decision support systems, Information Integration: modes of information, wrappers in mediator based systems, data analysis and OLAP, data mining, data warehousing, and information retrieval systems, and applications
Advanced Data Types and New Applications: Motivation, time in databases, spatial and geo-graphic data, multimedia databases, mobility and personal databases
Advanced Transaction Processing: Transaction processing monitors, transactional workflows, main memory databases, real time transaction systems, long duration transactions, transaction management in multidatabases
Multidimensional Indexes: Application needing multiple dimensions, hash like structures for multidimensional data, tree like structures for multidimensional data, bitmap indexes
XML: Background, Structure of XML Data, XML Document Schema, Querying and Trans-formation, API, Storage of XML Data, XML Applications.

References:

- Naveen Prakash, "Introduction to database management", *TMH*
- Rob and Coronel, "Database Systems", Fifth Edition, *Thomson*
- Molino, Ullman and Widom, "Database System Implementation", *Pearson Education Asia*
- Ozsu and Valduriez, "Principles of Distributed Database Systems", *Pearson Education Asia*
- Database management, Objectives, system functions and administration, Gordon Everest,
- Ramkrishnan and Gehrke, "Database Management Systems", *MGH International Edition*
- Silberchatz, Korth and Sudarshan, "Data base systems concepts", *MGH*, 4th edition

List of Experiments:

1. ER Diagram and Relational Model,
2. Creation of Database from ER Model,
3. Advanced SQL Queries,
4. Implementation of views,
5. Implementation of triggers,
6. Implementation of procedures,
7. Implementation of ORDBMS Concept.

CS423 TCP/IP Networking (CR-5, L-4, P-2)

Course objective:

- Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies;
- Have a basic knowledge of the use of cryptography and network security;
- Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols;
- Analyze, specify and design the topological and routing strategies for an IP based networking infrastructure
- Have a working knowledge of datagram and internet socket programming**.**

Course outline:

Introduction and Underlying Technologies: Introduction, the OSI Model and the TCP/IP Protocol Suite, Underlying Technologies
Network Layer: Introduction to Network Layer, IPv4 Addresses, Delivery and Forwarding of IP Packets, Internet Protocol Version 4 (IPv4), Address Resolution Protocol (ARP), Internet Control Message Protocol Version 4 (ICMPv4), Unicast Routing Protocols (RIP, OSPF, and BGP), Multicasting and Multicast Routing Protocols
Transport Layer: Introduction to the Transport Layer, User Datagram Protocol (UDP), Trans-mission Control Protocol (TCP), and Stream Control Transmission

Protocol (SCTP)
Application Layer: Introduction to the Application Layer, Host Configuration: DHCP, Do-main Name System (DNS), Remote Login: TELNET and SSH, File Transfer: FTP and TFTP, World Wide Web and HTTP, Electronic Mail: SMTP, POP, IMAP, and MIME, Network Management: SNMP, Multimedia
IPv6 Addressing

References:

- Internetworking with TCP/IP(5th Edition), Douglas E. Comer
- TCP/IP Protocol Suite, 4/e, Forouzan
- Computer Networks, 4/e, Andrew S. Tanenbaum

List of Experiments:

1. Write a program to determine Address Classes of an IP address ,
2. Write a program for Addresses subnetting in IP address
3. Write a program for Variable length subnet masking
4. Study of network monitoring and analysis
5. Program for header checksum calculation
6. Socket programming
   a. TCP
   b. UDP
7. Configuration
   a. ARP server
   b. DHCP server
   c. Proxy server
   d. DNS server
   e. Web server
8. Protocol Analysis of TCP, UDP, and IP using Wireshark tool

CS424: Cryptography And Network Security (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Have a fundamental understanding of the objectives of cryptography and network security.
- Become familiar with the cryptographic techniques that provide information and network security.
- Be able to evaluate the security of communication systems, networks and protocols based on a multitude of security metrics
- Understand a variety of generic security threats and vulnerabilities, and identify and analyze particular security problems for a given application.
- Understand the design of security protocols and mechanisms for the provision of security services needed for secure networked applications.
- Choose a suitable ciphering algorithm according to the required security level.
- Understand a given ciphering algorithm and to analyze it.

Course outline:

Introduction to cryptography: What is Cryptography, Encryption Schemes, Functions, Secret Key Cryptography, Public Key Cryptography, Hash Algorithms

Mathematical Background for Cryptography: Modulo arithmetic, Euclid's algorithm, alge-braic structures- groups, rings, fields-Polynomial fields, prime numbers, Fermat's theorem, Eyler's totient function, Euler's theorem, testing for primality-Probabilistic Considerations, Chinese reminder theorem, Discrete Logarithms – the powers of an integer, Modulo n, Indices, calculation of Discrete Logarithms
Conventional Encryption: Classical techniques, Modern Techniques, Algorithms, Confidentiality using conventional encryption
Public Key encryption and Hash Function: Public Key Cryptography, Message authentication and hash function, Digital Signatures and authentication protocols
System Security: Kerberos, Web security SSL, TSL, Firewalls

References:

- Bernard Menezes, "Network Security and Cryptography", Cengage Learning,
- William Stalling, "Cryptography and Network and Network security-Principals and practices", *Pearson Education*
- King, Dalton, and Osmanoglu, "Security Architecture", *TMH edition*
- Kaufman, Perlman, and Spenciner, "Network Security", *PHI*

List of Experiments:

1. Write program for Mono alphabetic cipher
2. Implementation of Play Fair cipher
3. Implementation of Vigenere cipher (Polyalphabetic substitution)
4. Implementation of Hill cipher.
5. Implementation of S-DES algorithm for data encryption.
6. Implement RSA asymmetric (public key and private key)-Encryption. Encryption key (e, n) & (d, n)
7. Implement RC4 encryption algorithm
8. Generate digital signature using Hash code.
9. Generate digital signature using MAC code.

CS425: Mini-Project (CR-1, L-0, P-3)

Course objective: At the end of this course the student should be able to

- Preparedness to study independently any other Domain, Technology and
  programming languages and apply to variety of real time problem scenarios

Mini-Project shall be based on any recent topic selected by the students working in a group. In any group more than two students are not allowed. Teaching load of three hours per week per batch of six students shall be allotted to the teacher. The guide shall give the term-work marks by assessing the work done and the submitted bound report by the students in the group. External practical examination shall be based on the work demonstrated by the group, followed by the oral examination conducted by the panel of examiners, consisting of guide working as a senior examiner and other external examiner(s), appointed by the Institute.
.

Elective Subject:

CS426: Artificial Intelligence & Expert Systems (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Understanding of both the achievements of AI and the theory underlying those achievements.
- Distinguish the engineering issues underlying the design of AI Systems.
- Use traditional AI language including an ability to write simple to intermediate programs and an ability to understand code written in that language.
- Describe the key components of the artificial intelligence (AI) field
- Describe search strategies and solve problems by applying a suitable search method
- Describe minimax search and alpha-beta pruning in game playing.
- Describe and apply knowledge representation
- Describe and apply probability theorem and Bayesian networks.
- Describe the key aspects of intelligent agents
- Describe the key aspects of Evolutionary computation, including genetic algorithms and genetic programming.
- Describe the key aspects of Machine learning

Course outline:

Introduction: General, Developments in Artificial Intelligence, Developments in Expert Systems, Role of AI and Expert Systems in Engineering.

Search Techniques: Problem Definition and Solution Process, Production Systems, Search Techniques, Problem Decomposition and AND-OR Graphs.
Knowledge-Based Expert System: What is KBES?, Architecture of KBES.
Engineering Design Synthesis: Synthesis, Decomposition Model for Synthesis, Role of Synthesizer in KBES Environment, An Architecture for a Synthesizer-A Generic Tool Generic Synthesis Tool-GENSYNT
Criticism and Evaluation: Methodologies Used in Knowledge Based Environment, A Frame-work for Critiquing and Evaluation, Generic Critiquing Tool-GENCRIT
Case-Based Reasoning: Applications of Case-Based Reasoning, Case-Based Reasoning Process, A Framework for CBR in Engineering Design (CASETOOL), Architecture of CASETOOL Application Example
Process Models and Knowledge-Based System: Expert Systems for Diagnosis, Blackboard Model of Problem Solving, ODESSY-An Integrated System for Preliminary Design of Rein-forced Concrete Multistory Office Buildings, Conceptual Design of a Car Body Shape, SETHU-An Integrated KBES for Concrete Road Bridge Design

References:

- C.S. Krishnamoorthy, S. Rajeev: Artificial Intelligence and Expert Systems for Engineers,
- Stuart E. Savory, Artificial Intelligence & Expert Systems, Ellis Horwood Ltd

List of Experiments:

1. Turbo Prolog features and format.
2. Program using variables in Prolog
3. Program for usage of rules in Prolog
4. Program for using Input, Output and fails predicates in prolog.
5. Program for studying Usage of Arithmetic operators in Prolog.
6. Program to study usage of Cut, Not, Fail predicates in Prolog
7. Program to study usage of Recursion in prolog.

8. Programs to study usage of Logical, Arithmetic, String operators in Prolog
9. Usage of Compound Object and List in prolog
10. Program for studying usage of Dynamic Database in prolog

CS427: Cloud Computing (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to:

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
- Identify the architecture and infrastructure of cloud computing, including saas, paas, iaas, public cloud, private cloud, hybrid cloud, etc.
- Explain the core issues of cloud computing such as security, privacy, and interoperability.
- Choose the appropriate technologies, algorithms, and approaches for the related issues.
- Identify problems, and explain, analyze, and evaluate various cloud computing solutions.
- Provide the appropriate cloud computing solutions and recommendations according to the applications used.

Attempt to generate new ideas and innovations in cloud computing.

Course outline:

Overview of Distributed: Computing trends of computing, Introduction to distributed computing, Next big thing: cloud computing
Introduction to Cloud Computing: What's cloud computing, Properties & Characteristics, Service models, Deployment models
Infrastructure as a Service (IaaS): Introduction to IaaS, Resource Virtualization, Server, Storage, Network, Case studies
Platform as a Service (PaaS): Introduction to PaaS**,** Cloud platform & Management**,** Computation, Storage, Case studies
Software as a Service (SaaS): Introduction to SaaS**,** Web services**,** Web 2.0, Web OS, Case studies
Cloud issues and challenges: Cloud provider Lock-in, Security

References:

- Executive's Guide to Cloud Computing by Eric A. Marks.
- Cloud Computing by Anthony T. Velte.

List of Experiments:

1. Program for Large data processing in the cloud
2. Study Virtualization techniques and security
3. Configure Program Resource and power management in the cloud
4. Monitoring and SLA assurance
5. Configure Semantic cloud and SaaS

CS428: Artificial Neural Networks (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to:

- Understand and explain strengths and weaknesses of the neural-network algorithms.
- Explain the function of artificial neural networks of the Back-prop, Hopfield, RBF and SOM type.
- Explain the difference between supervised and unsupervised learning.
- Describe the assumptions behind, and the derivations of the ANN algorithms dealt with in the course.
- Efficiently and reliably implement the algorithms introduced in class on a computer, interpret the results of computer simulations.
- Give example of design and implementation for small problems
- Implement ANN algorithms to achieve signal processing, optimization, classification and process modeling**.**

Course outline:

Feedforward networks: Fundamental concepts- Models of artificial neural network (ANN); Learning and adaption; Learning rules, Classification model, Features and decision regions, Perceptron networks, Delta learning rules for multi-perceptron layer, Generalized learning rule, Error backpropagation training, Learning factors.
Recurrent networks: Mathematical foundation of discrete time and gradient type Hopefield networks, Transient response and relaxation modeling.
Self-organizing networks: Hamming net and MAXNET, Unsupervised learning of clusters, Counterpropagation network, Feature mapping, Self organizing feature maps, Cluster discovery network (ART1).
Fuzzy Neural Networks: Fuzzy set theory, Operations on fuzzy sets, Fuzzy neural networks, Fuzzy min-max neural networks, General fuzzy min-max neural network **Applications**: Handwritten character recognition, Face recognition, Image compression

References:

- Jacek Zurada, "Introduction to ANN", *Jaico Publishing House*
- Bose and Liang, "Neural network fundamentals with Graphs, Algorithms, and Applications", *TMH edition*
- Ham and Kostanic, "Principles of Neurocomputing for Science and Engineerin", *TMH edition*

List of Experiments:

1. Introduction to Neural Applications
2. Classification of Linearly Separable Objects
3. Classification of Non-Linearly Separable Objects (XOR Problem)
4. Visual Understanding of Error Minimization, Creating Perceptrons
5. Preparing Input Data and Target Outputs Character Recognition Using a BPNN Generalizing Random Initial Weights for Hidden and Output Layers

CS429: Information Theory & Coding (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to:

- Define and apply the basic concepts of information theory (entropy, etc.).
- Differentiate between lossy and lossless data compression methods, and describe the most common such methods;
- Design an efficient data compression scheme for a given information source.
- Calculate the capacity of communication channels;
- Sketch Shannon's proof regarding the limits of error-free Communication.
- Explain the impact of feedback and/or many senders or Receivers on the communication problem.

Course outline

Information Theory: Information – Entropy, Information rate, classification of codes, Kraft McMillan inequality, Source coding theorem, Shannon-Fano coding, Huffman coding, Extend-ed Huffman coding - Joint and conditional entropies, Mutual information - Discrete memory less channels – BSC, BEC – Channel capacity, Shannon limit.
Source Coding: Text, Audio and Speech: Text: Adaptive Huffman Coding, Arithmetic Coding, LZW algorithm – Audio: Perceptual coding, Masking techniques, Psychoacoustic model, MEG Audio layers I,II,III, Dolby AC3 - Speech: Channel decoder, Linear Predictive Coding
Source Coding: Image and Video: Image and Video Formats – GIF, TIFF, SIF, CIF, QCIF – Image compression: READ, JPEG – Video Compression: Principles-I,B,P frames, Motion estimation, Motion compensation, H.261, MPEG standard
Error Control Coding: Block codes: Definitions and Principles: Hamming weight, Hamming distance, Minimum distance decoding - Single parity codes, Hamming codes, Repetition codes - Linear block codes, Cyclic codes - Syndrome calculation, Encoder and decoder – CRC
Error Control Coding: Convolution codes: Convolution codes – code tree, trellis, state diagram - Encoding – Decoding: Sequential search and Viterbi algorithm – Principle of Turbo coding

References:

- R Bose, "Information Theory, Coding and Cryptography", TMH 2007
- Fred Halsall, "Multimedia Communications: Applications, Networks, Protcols and Standards", Perason Education Asia, 2002
- K Sayood, "Introduction to Data Compression" 3/e, Elsevier 2006
- S Gravano, "Introduction to Error Control Codes", Oxford University Press 2007
- Amitabha Bhattacharya, "Digital Communication", TMH 2006

List of Experiments:

1. Review of probability theory and Entropy
2. Mutual information
3. Data compression
4. Huffman coding
5. Asymptotic equipartition property
6. Universal source coding
7. Channel capacity
8. Differential entropy
9. Block codes and Convolutional codes

SEMESTER-II

CS431: Distributed Computing (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Build formal models of distributed systems;
- Differentiate between synchronous, asynchronous and hybrid models;
- Understand the assumptions and limitations underlying models of distributed systems;
- Describe the more relevant problems in distributed systems;
- Design new distributed algorithms;
- Invoke impossibility results to avoid wasting time trying to solve an unsolvable problem;
- Prove impossibility results.
- Distinguish between parallel and distributed computers.

Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library.

Course outline:

Introduction Distributed Computing: Definition of a distributed system, goals, architecture.,
Processes: Threads, virtualization, clients, servers, code migration Case study: Mobile Agents**.**
Communication: Remote procedure call, message-oriented communication, stream-oriented communication, Case, Study: RMI, MPI.
Naming: flat naming, structured naming, attribute-based naming Case study: LDAP
Synchronization: clock synchronization, mutual exclusion, election algorithms
Consistency and replication: Data-centric consistency models, client-centric consistency models, consistency protocols.
Distributed object-based systems: Distributed Objects, Object Servers, Binding a Client to an Object, Java RMI, Common Object Request, Broker Architecture (CORBA) Case Studies: Enterprise Java Beans.
Distributed file systems: architecture, NFS, synchronization, consistency and replication, Case Studies: Hadoop Distributed File System (HDFS) .
Distributed web-based systems: Web-Based Systems ,Web Services, Web Server /Client, Web Server Clusters ,Web Proxy Caching, Case Study: Apache Web Server, Squid.
Programming Models for Distributed Systems: MapReduce: Simplified data processing on large clusters.

References:

- Distributed Systems. Principles and Paradigms. (2nd Ed.) Prentice Hall, 2007. Andrew S. Tanenbaum, Vrije University, Amsterdam, The Netherlands, Maarten Van Steen
- Distributed System Concepts and Design(5th Ed.), George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair
- Hadoop, The Definitive Guide, Oreilly, 2010. Tom White

List of Experiments:

1. Implement concurrent echo client-server application, Implement concurrent day-time client-server application, Configure following options on server socket and tests them: SO_KEEPALIVE, SO_LINGER, SO_SNDBUF, SO_RCVBUF, TCP_NODELAY
2. Incrementing a counter in shared memory
3. Create CORBA based server-client application
4. Design XML Schema and XML instance document
5. WSDL based: Implement ArithmeticService that implements add, and subtract operations
6. Java based: Implement TrigonometricService that implements sin, and cos operations, Monitor SOAP request and response packets. Analyze parts of it and compare them with the operations (java functions) headers
7. Design and test BPEL module that composes ArithmeticService and TrignometricService
8. Test open source ESB using web service, Implementing Publish/Subscribe Paradigm using Web Services, ESB and JMS
9. Implementing Stateful grid services using Globus WS-Core-4.0.3
10. HADOOP

CS432: ADVANCED ALGORITHMS (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Have a thorough understanding of a variety of algorithms with practical applications and the resource requirements of each.
- Have a good idea as to where these algorithms can be used.
- Describe the following classes of algorithms and design principles associated with them: recursive algorithms, graph (search-based) algorithms, greedy algorithms, algorithms based on dynamic programming, network flow (optimization) algorithms, approximation algorithms, randomized algorithms, distributed and parallel algorithms.
- Apply the studied design principles to produce efficient algorithmic solutions to a given problem taking account of the strengths and weaknesses of the applicable principles.

Outline methods of analyzing correctness and asymptotic performance of the studied classes of algorithms, and apply them to analyze correctness and asymptotic performance of a given algorithm.

Course outline:

Algorithmic paradigms: Dynamic Programming, Greedy, Branch-and-bound; Asymptotic Complexity, Amortized analysis; Graph Algorithms: Shortest paths, Flow networks; NP-completeness; Approximation algorithms; Randomized algorithms and advanced data-structures, topics of current Research.

References:

- T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, McGraw Hill, 1994.
- Jon Kleinberg, Eva Tardos, Algorithm Design, Pearson Addison-Wesley, 2006
- Dan Gusfield, Algorithms on Strings, trees and Sequences, Cambridge, 2005.
- Sara Baase, Computer Algorithms: Introduction to Design and Analysis, Addison Wesley, 1998.
- Michael T Goodrich & Roberto Tamassia, Algorithm Design: Foundations, Analysis & Internet Exam-ples, John Wiley, 2002

List of Experiments:

1. Write a program that implements tower of hanoii.
2. Write a program that implements heap sort.
3. Write a program that implements quick sort.
4. Write a program that implements merge sort.
5. Write a program that implements Prim's algorithm.
6. Write a program that implements Kruskal's algorithm.
7. Write a program that implements String editing.
8. Write a program that implements knapsack using greedy.
9. Write a program that implements Dijkstra's aalgorithm. .
10. Write a program that implements Longest Common Subsequence.
11. Write a program that implements Nqueen Problem.
12. Write a program that implements knapsack using backtracking

CS433: MOBILE COMPUTING (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Learn state-of-the-art wireless technologies;
- Obtain background for original research in wireless networking and mobile computing field;
- Have a good understanding of how the underlying wireless and mobile communication networks work, their technical features, and what kinds of applications they can support;
- Understand the characteristics and limitations of mobile hardware devices including their user-interface modalities
- Explain the structure and components for Mobile IP and Mobility management;
- Understand positioning techniques and location based services and applications;
- Describe the important issues and concerns on security and privacy;
- Develop mobile computing applications by analyzing their characteristics and requirements, selecting the appropriate computing models and software architectures, and applying standard programming languages and tools.
- Design and development of context-aware solutions for mobile devices

Course outline:

Wireless Transmission: Signals, propagation, signal encoding, multiplexing, modulation and spread spectrum.**Wireless LANS:** IEEE 802.11, Bluetooth and Hiperlan.**Mobile Network Layer:** IP packet delivery, agent discovery, registration, tunneling and encapsulation, optimization, reverse tunneling, mobile ad-hoc networks **Mobile Transport Layer:** Indirect TCP, snooping TCP, mobile TCP, transaction oriented TCP, TCP over 3G wireless networks **Wireless WANS:** Cellular network, GSM, GPRS, UMTS, CDPD and CDMA **Other Topics:** Operating Systems for mobile devices, wireless application protocol, WML and WML Scripts

References:

- Mobile Communications, Jochen Schiller
- Wireless Communications and Networks, William Stallings
- Mobile Computing, Talukder and Yavagal
- The Wireless Appilcation Protocol, Singhal, Bridgman, Mauney, Alvinen, Bevis, Chan and Hild

List of Experiments:

1. To check orthogonality of two codes.
2. Generation of Walsh codes.
3. To implement Code Division Multiple Access (CDMA).
4. To study frequency reuse.
5. To study Choice Group class and its implementation in J2ME.
6. To study Canvas class and its implementation in J2ME.
7. WML programs, WML script programs
8. Write a WML page to display an image and to accept input from the user.
9. Study Assignment 1: Detailed study of Bluetooth.

CS434: Project –II (CR-4, L-0, P-8)

Course objective: At the end of this course the student should be able to

- Preparedness to study independently any other Domain, Technology and

Programming languages and apply to variety of real time problem scenarios

Project shall be based on any recent topic selected by the students working in a group. In any group more than two students are not allowed. Teaching load of eight hours per week per batch of six students shall be allotted to the teacher. The guide shall give the term-work marks by assessing the work done and the submitted bound report by the students in the group. External practical examination shall be based on the work demonstrated by the group, followed by the oral examination conducted by the panel of examiners, consisting of guide working as a senior examiner and other external examiner(s), appointed by the Institute.

Elective Subject:

CS436: Data Intensive computing (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to:

- Deal with information management, business intelligence and predictive analysis for big data;
- Use large-scale computing systems to solve data-intensive real-world problems
- Transform huge amounts of data into knowledge and intelligence for human understanding and decision making.
- Introduce the fundamentals in data intensive computing and its enabling systems architectures such as MapReduce, cloud computing and storage, with a focus on system architecture, middleware and building blocks, programming models, algorithmic design, and application development.

Harness the capabilities of the cloud computing infrastructures.

Course outline:

Distributed Systems, Supercomputing, Grid Computing, Cloud Computing, Many-core Computing, Data Intensive Computing, Storage Systems, Distributed and Parallel File Systems, Parallel I/O, Local Resource Management, Scientific Computing and Applications, Parallel Programming Systems and Models, MapReduce, Data-Intensive Computing with GPUs, Data-Intensive Computing with Databases, Introduction data warehousing and data mining, Data Warehouse and OLAP Technology, Data Warehouse Architecture, Data Preprocessing, Data Mining, Classification and Prediction, Cluster Analysis

References:

- Transactional Information Systems by Gerhard WEIKUM and Gottfried VOSSEN
- Handbook of Cloud Computing, "Data-Intensive Technologies for Cloud Computing," by A.M. Middleton. Handbook of Cloud Computing. Springer, 2010
- Jiawei Han and Miceline Kamber, “Data Mining: Concepts and Techniques”, Elsevier publication
- George M. Marakas, “Modern Data Warehousing, Mining & Visualization”, Pearson Education, 2003
- Margaret H. Dunham, “Data mining: Introductory & Advanced Concepts”, Pearson Education, 2003

List of Experiments:

1. Distributed file systems,
2. Data aware scheduling algorithms,
3. Distributed operating systems,
4. Distributed job management systems,
5. Parallel programming languages,
6. Distributed workflow systems,
7. Distributed monitoring systems,
8. Scientific computing with GPUs,
9. Scientific computing with MapReduce,
10. Distributed caching strategies,
11. Distributed cache eviction policies,
12. Distributed hash tables,
13. Virtualization impact for data-intensive computing.

CS437: Multimedia Systems (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Develop a solid understanding of Interactive Multimedia Design principles;
- Develop expertise in Interactive Multimedia Design practices;
- Understand the interplay between design and technology in interactive multimedia;
- Students will be capable of understanding different realizations of multimedia tools and their usage;
- Implementing various multimedia standards and compression technologies;
- Become proficient in the use of core interactive multimedia tools and technologies;
- Develop their ability to work as part of a team and also as an individual;
- Become conversant with contemporary and emerging issues in multimedia and the Web;
- Learn how to find, synthesize and apply data from a range of sources

Course outline:

Introduction: Multimedia elements and applications, Architecture, Evolving technologies, De-fining objects, Data interface standards, Multimedia databases
Compression and decompression: Types of compression, Binary image compression schemes, Color, gray scale and still video image compression, Video image and audio compression
Data and file format standards: RTF, TIFF, BMP, RIFF, MIDI, JPEG DIB, AVI and MPEG
Multimedia input/output technologies: Issues, pen input, video and image display systems, print output technologies, image scanners, digital voice and audio, digital camera, video images and animation, and full motion video
Storage and retrieval technologies: Magnetic media technologies, optical media, hierarchical storage and cache management
Architectural and telecommunications considerations: Specialized computational processors, memory systems, multimedia board solutions, LAN/WAN connectivity, Distributed object model
Multimedia application design: Application classes, types of multimedia systems, virtual reality design, components of multimedia systems, organizing multimedia databases, application workflow and distributed application design issues
Multimedia authoring and user interface: Multimedia authoring systems, hypermedia application design considerations, user interface design, information access, and object display/ playback issues
Multimedia messaging: Mobile messaging, hypermedia message components, hypermedia linking and embedding, creating hypermedia messages, integrated multimedia message standards and document management
Distributed multimedia systems: Components, distributed client-server operation, multimedia object severs, multi-server network topologies, distributed multimedia databases and managing distributed objects
Multimedia Database Systems: Multimedia database management system, characteristics of an MDBMS, data analysis, data structure, operations on data, integration in a database model, relational database model, object-oriented database model

References:

- Prabhat K. Andleigh and Kiran Thakrarar, “Multimedia systems design”, PHI, 2002
- John F. Koegel Buford, “Multimedia systems”, Pearson Education, 2002
- Steinmetz and Nahrstedt, “Multimedia: Computing, Communications and Applications”.
- Tay Vaughan, “Multimedia Making it work”, Fifth Edition, TMH
- Chapman,” Digital Multimedia” Wiley India.
- Ranajan Parekh, “Principles of Multimedia”, Tata McGraw Hill
- Buford – “Multimedia Systems”, Pearson.

List of Experiments:

1. Study of MAYA software.
2. Study of FLASH software.

3. Creating a Flash Banner.
4. Creating animation with Flash:
   a. Creating a ghost (unshaped) 2D object.
   b. Creating animation with moving objects.
   c. Adding sound to the animation (with play button).
5. Creating a game using action script.
6. Creating a flash based presentation (4/5 frames) with UI controls.
7. Study of VLC player, its setting, streaming and non-streaming techniques.
8. Study of VRML.
9. Creating a 3D object using 2D and show special effects for the same.

CS438: Computer Vision (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Learn digital image fundamentals: visual perception, digital image pixels, image features.
- Use statistical methods to disentangle data using models constructed with the aid of geometry, physics and learning;
- Obtain simple inferences from individual pixel values;
- Combines information from multiple images into a coherent whole;
- Imposes order on groups of pixels to segment them or infer shape information;
- Recognize objects using geometric information or probabilistic techniques.
- Learning and applying knowledge in analyzing image segmentation, representation, description, and recognition techniques.

Course outline:

Introduction: Image formation-image model, imaging devices

Early processing: Recovering intrinsic structure, Filtering Image, finding local edges, Range information from geometry, Surface orientation, Optical flow, Resolution pyramids.
Boundary detection: Searching near and approximate location, Hough method for curve detection, Edge following as graph searching, Edge following as dynamic programming, Contour following
Region growing: Regions, local technique, Blob coloring, Global techniques, Splitting and merging
Texture: Structural models, Texture as a pattern recognition problem, Texture gradients
Motion: Motion understanding, Optical flow, Image sequences
Representation of 2-D geometrical structure: Boundary representation, Region representation, Simple shape properties, Representation of 3-D structures, Solids and their representation, Surface representation, Generalized cylinder representation, Volumetric representation, Understanding line drawings
Knowledge representation and use: Knowledge base models and processes, Semantic nets, Control issues in vision systems
Matching: Aspects, Graph theoretic algorithms, Implementation, Matching in practice
Inference: First order predicate Calculus, computer reasoning, Production systems, Scene labeling, Active knowledge

References:

- Ballard and Brown, "Computer Vision", Prentice Hall publication
- Jain, Kasturi and Schunck, "Machine Vision", McGraw-Hill International Editions

List of Experiments:

1. Introduction: OpenGL -- The Industry's Foundation for High Performance Graphics.
2. Program to recursively subdivide a tetrahedron to from 3D Sierpinski gasket. The number of recursive steps is to be specified by the user.
3. Program to draw a color cube and spin it using OpenGL transformation matrices.
4. Program to create a house like figure and rotate it about a given fixed point using OpenGL functions.
5. Program to create a cylinder and a parallelepiped by extruding a circle and quadrilateral respectively. Allow the user to specify the circle and the quadrilateral.
6. Program to draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing. Use OpenGL functions.
7. Program to display a set of values { fij } as a rectangular mesh.
8. Program, using OpenGL functions, to draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the properties of the surfaces of the solid object used in the scene.

CS439: High Performance Computing (CR-5, L-4, P-2)

Course objective: At the end of this course the student should be able to

- Describe the basic features of a modern CPU and analyze how these affect the performance of a code;
- Identify and correct common inefficiencies in both serial and parallel scientific computer codes;
- Choose an appropriate programming paradigm for a particular problem or hardware architecture;
- Write a parallel program using shared memory or message passing constructs in a Physics context. Ability to write a simple GPU accelerated program;
- Detect source of performance bottlenecks in parallel computer programs and how these relate to basics of computing architecture;
- Use batch systems to access parallel computing hardware. Validate the correctness of a parallel computer program v/s equivalent serial software.

Course outline:

Introduction to parallel computing: Motivating parallelism, scope of parallel computing, Parallel programming platforms
Principles of parallel algorithm design: preliminaries, decomposition techniques, Mapping techniques and load balancing, Parallel algorithms models
Parallel programming: message passing: Introduction to MPI, using clusters of computers, evaluating parallel programs, debugging
Parallel programming: shared-memory: Thread basics, programming with pthreads, java threads, openMp
Load balancing and termination detection: Dynamic load Balancing, Distributed termination detection algorithms

References:

- An Introduction to Parallel Computing: Design and Analysis of Algorithms, Second Edition - A.Grama, A. Gupta, G. Karypis and V. Kumar, Pearson
- Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers" (2nd ed.) by B. Wilkinson and M. Allen, Prentice Hall.
- Parallel Programming: for Multicore and Cluster Systems, Thomas Rauber, Gudula Rünger, Springer.

List of Experiments:

1. Compiling and running a MPI Program
2. Running a parallel MPI program to solve a problem.
3. Computational modeling -- the modeling of real-world processes or phenomena in software.
4. Solve a problem using *domain decomposition*.
5. Solve the same problem using *processor farm* model.
6. Introduce *MPE*, the MPI graphics library.
7. Practice tuning an algorithm, and measuring the resulting changes in speedup and efficiency.
8. Practice using some of the MPI collective communication operations, and introduce the use of OpenMP.
9. Computing PI via Posix Threads.