# FOURTH YEAR

# COMPUTER ENGINEERING

# Pattern - D

# SEMESTER I

# THEORY

# CS1204 : Operating Systems

**Prerequisites :**

1. Data Structures and Algorithms
2. Computer Organization

**Aim :**

This course provides an understanding of the standard problems and their solutions in the area of operating systems (including process management, storage management, I/O systems design) and with practical experience with one or more modern operating systems.

**Objectives:**

1. Identify the role of operating systems and explain the different structures of operating systems.
2. Describe OS support for processes/threads, and virtual memory, I/O and file systems.
3. Evaluate processes and/or threads synchronization mechanisms and explain deadlock conditions and ways to resolve them.
4. Identify the different design and implementation concepts for Unix/Linux
5. Use Inter-Process Communication techniques under Unix/Linux.

**Unit 1 : Introduction to OS ( 7 Hrs )**

Architecture, Goals & Structures of O.S., Hardware Abstraction layer, Basic functions, Interaction of OS and hardware architecture, System Calls, Batch, multiprogramming, multitasking, time sharing, parallel, distributed & real-time OS.

Unix OS: Architecture, Kernel Data Structures, Buffer Headers, Structure of Buffer Pool, Scenarios for retrieval of a Buffer, Reading and Writing Disk Blocks, Advantages and Disadvantages of the Buffer Cache

MS-Windows 2000: Architecture

**Unit 2 : Process Management ( 7 Hrs )**

Process description & control: Process Concept, Process states, Process description, Process control, Threads, SMP & Microkernels: Processes and Threads, Symmetric Multiprocessing, Microkernels

Concurrency: Principles of Concurrency, Mutual Exclusion: Semaphores, Message Passing, Monitors, Classical Problems of Synchronization: Readers-Writers problem, Producer Consumer Problem, Dining Philosopher problem

**Unit 3 : Deadlock and Linux Programming ( 7 Hrs )**

Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery
Shell and Command Programming, AWK Programming

**Unit 4 : Memory Management ( 7 Hrs )**

Memory management requirements, Fixed and Variable Partitioning, Allocation Strategies (First Fit, Best Fit, Worst Fit), Fragmentation, Swapping.
Virtual Memory: Concepts, Segmentation, Paging, Address Translation, Demand paging, Page Replacement Policies (FIFO, LRU, Optimal), Thrashing, Working Set Model

**Unit 5 : UNIX Representation of Files ( 7 Hrs )**

Internal Representation of files: Inodes, Structure of a regular file, Directories, Conversion of a path name to an Inode, Super Block, Inode assignment to a new file, Allocation of Disk Blocks, Other file types
System Calls for the file system: open, read, write, file & record locking, lseek, Close, File Creation, Creation of Special Files, Change Directory and Change Root, Change Owner and Change Mode, stat and fstat, pipes, dup, mounting and unmounting file systems, link, unlink, file system abstractions, file system maintenance

**Unit 6 : Process Management in UNIX ( 7 Hrs )**

Structures of processes: Process States and Transitions, Layout of System Memory, The Context of a process, Saving the context of a process, Manipulation of the process address space, Sleep
Process control: Process Creation, Signals, Process Termination, Awaiting Process Termination, Invoking other programs, The User ID of a process, Changing the size of a process, The shell, System boot and the init process

Process Scheduling and Time: Process Scheduling, System calls for time & clock

**Outcomes:**

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

1. Summarize the principles underlying the design and construction of a typical operating system, giving particular recognition to the wider applicability of the ideas and the influences from such developments as high-level languages, networking, multimedia, and security concerns.
2. Describe the concept of a process and how processes deal with scheduling, cooperation, and communication with other processes.
3. Explain the classical problems in process synchronization and know several different ways to solve such problems, including semaphores, critical regions, and monitors.
4. Describe several different schemes for managing main memory, including swapping, virtual memory, paging, and segmentation.
5. Compare and contrast several schemes for file allocation and file management.
6. Describe the characteristics of an I/O system and explain how the user, the operating system, and the hardware interact with I/O.

**Text Books:**


1. Stalling William, "Operating Systems", Pearson Education,4th Edition ,2001.
2. Silberschatz A., Galvin P., Gagne G., "Operating System Concepts", John Wiley and Sons, 7th Edition, 2003.


**Reference Books:**


1. Tanenbaum Andrew S., "Modern Operating Systems" PHI, 2nd Edition, 2001.
2. Bach Maurice, "Design of the Unix Operating System", Pearson Education, 1st Edition, 1990.
3. Das Sumitabha, "Unix Concepts and Applications", Tata McGraw Hill, 3rd Edition, 2003.

# CS1024: Object Oriented Modeling and Design

**Prerequisites :**
1. Software Engineering
2. Object Oriented Programming

**Aim :**
1. The major emphasis of this course is on using object-oriented modeling to define a system specification. A study of object-oriented techniques using Unified Modeling Language (UML) for the analysis and design of software systems will be performed.
2. Techniques for designing both the structural and behavioral aspects of software systems are emphasized. This course will cover object-oriented approaches to system analysis, data modeling and design that combine both process and data views of systems.

**Objectives:**
1. To experience the insights necessary to obtain maximum benefit from object technology
2. To understand the need for, the place of, and aims of, requirements, analysis and design
3. To thoroughly understand the practices of analysis and design (OOA and OOD)
4. To understand the practical connections between the theory of object-oriented design and the object-oriented programming languages
5. To become familiar with the unified modelling language (UML 1.x or UML 2.0)
6. To understand the relative merits of the different UML diagrams, distinguishing those diagrams most likely always to be useful to typical projects from those diagrams more likely to be of interest to more specialized projects
7. Transforming analysis behavioral models into design sequence diagrams

**Unit 1 : Introduction to Object Orientation ( 7 Hrs )**

**Elements of UML:** The importance of modeling, enabling concise communication, Building blocks: things, relationships and diagrams, Architectural views: use case, design, implementation, process and deployment, Levels of detail: visualization, specification and construction

**Object-oriented concepts:** Objects and classes, Links and relationships, Inheritance and polymorphism

**The Unified Process:** The object-oriented software life cycle, Use case-driven and architecture-centric features, Iterative and incremental development, performing requirements analysis

**Unit 2 : : UML 2.0 ( 7 Hrs )**

Programming In Small Versus Programming In Large, UML 2.0 History/ New Features MDA/ MOF/ XMI/ CORBA, Introduction to UML Metamodel, Extensibility Mechanisms and its usage, Introduction to OCL ,Behavioral Diagrams in UML ,Structural Diagrams in UML, Specification techniques of diagrams in UML

**Unit 3 : The Behavioral Model** **( 7 Hrs )**

**Use Cases:** Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Using Use Cases **Domain Analysis:** Top View - The Domain Perspective, Data Dictionary: Finding the Objects, Responsibilities, Collaborators, and Attributes, CRC Cards, Class Models and Use Case Models, Judging the Domain Model

**Producing Requirements Models**

Capturing system behavior in use cases: Finding primary and secondary use cases

Include and Extend dependencies, Use case generalization relationships, Refining use cases: rapid prototypes

Creating the domain object model: Building a class description database, Finding analysis classes, Managing analysis complexity with packages and subsystems

**Unit 4 : Object Analysis** **( 7 Hrs )**

**Use case realization:** Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Tying object and behavioral models with collaboration diagrams

**Implementing memory in objects using state machines:** States, events and actions

Nested machines and concurrency, Converting sequence diagrams into communicating state machines, Modifying the object model to facilitate states

**Analyzing object behavior:** Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Drilling Down, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization

Other Behavioral Diagrams: Communication Diagram, Interaction Overview Diagrams

Timing Diagrams

**Unit 5 : Object Design** **( 7 Hrs )**

**Design of Objects:** Design and Factoring, Design of Software Objects, Features and Methods, Cohesion of Objects, Coupling between Objects Coupling and Visibility,

Inheritance

**Advanced Objects:** Constructors & Destructors, Instance Creation, Abstract Classes

Polymorphism, Multiple Inheritance and associated Problems, Interfaces, Interfaces with Ball and Socket Notation, Templates

**Establishing The Object Model:** Refining classes and associations, Analysis model vs. design model classes, Categorizing classes: entity, boundary and control , Modeling associations and collections, Preserving referential integrity , Achieving reusability

Isolating reusable base classes, Reuse through delegation, Identifying and using service packages, Improving reuse with design patterns

**Unit 6 : Interfaces and Application of UML ( 7 Hrs )**

**Packages and interfaces:** Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces

**Component and deployment diagrams:** Describing dependencies, Deploying components across threads, processes and processors

**UML 2.0 in Application Engineering:** Application of UML in Embedded System, Application of UML in Web Engineering, Forward Engineering and Reverse Engineering Concepts

**Outcomes:**

Upon successfully completing this course the student will:

1. Understand different perspectives about the systems development process
2. Understand the role and importance of requirements analysis and specification
3. Understand the basic principles of object-orientation
4. Acquire a working knowledge of system modeling techniques
5. Become aware of the emerging ideas relevant to object-oriented systems development.
6. Create commonly expected "deliverables" of systems design including models of structure, behavior and dynamics

**Text Books**


1. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object Oriented Analysis and Design. ", 2nd Edition, Addison- Wesley, ISBN – 0321321278.
2. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049.


**Reference Books**


1. Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974.
2. Martin Flower, "UML Distilled: A Brief Guide to The Standard Object Modeling Language ", 3rd Edition, Addison- Wesley, ISBN – 0321193687.
3. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, "UML 2 Tool Kit", John Wiley & sons, ISBN – 0471463612.

# CS1194: Design and Analysis of Algorithms

**Prerequisites**: Data Structures and Algorithms.

**Aim:**
This course introduces basic algorithmic techniques, time requirements of an algorithm and mathematical techniques used in analysis of algorithms. The emphasis will be learning analysis of algorithms for a wide variety of foundational problems occurring in computer science applications with discussions on complexity and NP-completeness.

**Objectives:**
1. Fundamental understanding of the mathematics used to analyze, evaluate, and design algorithms
2. Develop the ability to assess the advantages and disadvantages of different types of algorithms.
3. Understand methods for designing time and space efficient algorithms.
4. Increased ability to design and implement efficient solutions to problems.

**Unit 1 : Introduction ( 6 Hrs )**

'O','Ω' and 'Θ' asymptotic notations, Average, Best and Worst case analysis of algorithms for Time and Space complexity, Amortized Analysis, Solving Recurrence Equations, Proof Techniques: by Contradiction, by Mathematical Induction

**Priority Queues** : Heaps & Heap sort.

**Unit 2 : Divide And Conquer And Greedy Strategy ( 9 Hrs )**

Divide and Conquer: General Strategy, Exponentiation. Binary Search, Quick Sort and Merge Sort. Greedy Method ,General Strategy, Knapsack problem, Job sequencing with Deadlines, Optimal merge patterns, Minimal Spanning Trees and Dijkstra's algorithm

**Unit 3 : Dynamic Programming ( 7 Hrs )**

General Strategy, Multistage graphs, OBST, 0/1 Knapsack, Traveling Salesperson Problem, Flow Shop Scheduling.

**Unit 4 : Backtracking & Branch And Bound ( 7 Hrs )**

**Backtracking:** General Strategy, 8 Queen's problem, Graph Coloring, Hamiltonian Cycles, 0/1 Knapsack.

**Branch and Bound:** General Strategy, 0/1 Knapsack, Traveling Salesperson Problem

**Unit 5 : Parallel Algorithms ( 7 Hrs )**

Computational Model, Basic Techniques and Algorithms (Complete Binary Tree, Pointer Doubling, Prefix Computation), Selection, Merging, Sorting Networks, Parallel Sorting, Graph Problems (Alternate Algorithm for Transitive Closure, All pairs shortest path)

**Unit 6 : NP-Hard And NP-Complete Problems ( 6 Hrs )**

**NP-Hard And NP-Complete Problems:**
Algorithms, Complexity-intractability, Non-Deterministic Polynomial time (NP) Decision problems, Cooks Theorem.
**NP-Complete problems**- Statisfiability problem, vertex cover problem.

**NP-Hard problems**-graph, scheduling, code generation problems, Simplified NP Hard Problems

**Outcomes:**

On successful completion of this course, the student will be able to

1. analyze the average- and worst-case performance of algorithms,
2. use the various strategies effectively,
3. apply the concept of NP-completeness and be familiar with approximation algorithms
4. implement the families of algorithms in the appropriate high-level language.

**Text Books**


1. Horowitz, Sahani, "Fundamentals of computer Algorithms", Galgotia. 2nd Edition, 1998.
2. Bressard, Bratley "Fundamentals of Algorithmics." ,PHI, 2nd Edition,1996.


**Reference Books**


1. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm" ,PHI 2nd edition, 2001.
2. A. V. Aho and J.D. Ullman, "Design and Analysis of Algorithms", Addison Wesley. 2nd edition.

# FOURTH YEAR

# COMPUTER ENGINEERING

# SEMESTER 1

# LABORATORY

# CS 6204 : Operating Systems

**Objectives:**

1. To study the operations performed by Operating Systems as a resource manager.
2. To learn the internal working of the UNIX operating system.

## List of Practicals

1. Execution of basic & advanced Unix commands.
2. Implement a shell program to find out if a given string is a palindrome or not.
3. Generate a student report using AWK programming.
4. Solve the Readers-Writers problem using threads and semaphores.
5. Solve the Producers-Consumers problem using threads and mutex.
6. Implement the Dining Philosopher's problem using Multithreading.
7. Write a program to simulate the following page replacement algorithm
    a. FIFO
    b. Optimal
8. Write a program using fork system call to create child process, suspend it using wait system call and transfer it into the zombie state.
9. Perform client-server communication using following IPC mechanism
    a. Unnamed pipe
    b. Named pipe
10. Write a program for File management using file access system call such as open, read, write, lseek and stat
11. Write a program for implementation of an alarm clock application using signals
12. Write and insert a module in Linux kernel

**Outcomes:**

Student will be able to program on the LINUX platform in industry. Since industry has specialized system oriented projects, this course will equip the student in handling those projects.

**Text Books:**


1. Stalling William, "Operating Systems", Pearson Education,4th Edition ,2001.
2. Silberschatz A., Galvin P., Gagne G., "Operating System Concepts", John Wiley and Sons, 7th Edition, 2003.


**Reference Books:**


1. Tanenbaum Andrew S., "Modern Operating Systems" PHI, 2nd Edition, 2001.
2. Bach Maurice, "Design of the Unix Operating System", Pearson Education, 1st Edition, 1990.
3. Das Sumitabha, "Unix Concepts and Applications", Tata McGraw Hill, 3rd Edition, 2003.

# CS1024 : Object Oriented Modeling and Design

**Prerequisites :**

1. Software Engineering
2. Object Oriented Programming

**Objectives:**

1. To experience the insights necessary to obtain maximum benefit from object technology
2. To understand the need for, the place of, and aims of, requirements, analysis and design
3. To thoroughly understand the practices of analysis and design (OOA and OOD)
4. To understand the practical connections between the theory of object-oriented design and the object-oriented programming languages
5. To become familiar with the unified modelling language (UML 1.x or UML 2.0)
6. To understand the relative merits of the different UML diagrams, distinguishing those diagrams most likely always to be useful to typical projects from those diagrams more likely to be of interest to more specialized projects
7. Transforming analysis behavioral models into design sequence diagrams

## List of Practicals

1. To narrate Requirement Definition Document for the target system with following three areas:
    a. Problem Identification
    b. Problem Definition
    c. Problem Statement
2. To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 std guidelines.
3. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases. The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
    a. Actors (External Users)
    b. Transactions (Use Cases)
    c. Event responses related to transactions with external agents.
    d. Detection of System boundaries indicating scope of system.

4. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object communication. The model should depict:
    a. Discrete, distinguishable entities (class).
    b. Events (Individual stimulus from one object to another).
    c. Conditional events and relationship representation.
5. To depict the state transition with the life history of objects of a given class model. The model should depict:
    a. Possible ways the object can respond to events from other objects.
    b. Determine of start, end, and transition states.
6. To depict the dynamic behavior using detailed Activity diagram.
7. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase.
8. To develop logical static structure of target system with Class diagram. The model should depict
    a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
    b. Identification of objects and their purpose.
    c. Roles / responsibilities entities that determine system behavior.
9. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules.
10. To represent deployment view of the system through Architecture Diagram.
11. To narrate the Program Design Language Constructs for the target system and implement the system according to specification.

**Outcomes:**

Upon successfully completing this course the student will:

1. Understand different perspectives about the systems development process
2. Understand the role and importance of requirements analysis and specification
3. Understand the basic principles of object-orientation
4. Acquire a working knowledge of system modeling techniques
5. Become aware of the emerging ideas relevant to object-oriented systems development.
6. Create commonly expected "deliverables" of systems design including models of structure, behavior and dynamics

**Text Books**


1. Jim Arlow, Ila Neustadt, "UML 2 and Unified Process: Practical Object Oriented Analysis and Design. ", 2nd Edition, Addison- Wesley, ISBN – 0321321278.
2. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049.


**Reference Books**


1. Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974.
2. Martin Flower, "UML Distilled: A Brief Guide to The Standard Object Modeling Language ", 3rd Edition, Addison- Wesley, ISBN – 0321193687.
3. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, "UML 2 Tool Kit", John Wiley & sons, ISBN – 0471463612.

# CS6194: Design And Analysis of Algorithms

**Objectives:**

1. Fundamental understanding of the mathematics used to analyze, evaluate, and design algorithms
2. Develop the ability to assess the advantages and disadvantages of different types of algorithms.
3. Understand methods for designing time and space efficient algorithms.
4. Increased ability to design and implement efficient solutions to problems.

## List of Practicals

1. Recursive and iterative( non recursive) algorithm for specific problem and there complexity measures(comparison expected).
2. Quick Sort/ Merge Sort implementations using divide and conquer approach. Time complexity measure is to be obtained.
3. Minimal spanning Trees/ Job scheduling as an example of Greedy approach
4. Finding shortest path for multistage graph problem. (single source shortest path and all pairs shortest path.)
5. OBST/Flow Shop Scheduling as an example of dynamic programming.
6. 0/1 knapsack's problem using Dynamic Programming, Backtracking and Branch & Bound Strategies.
7. 8-Queen problem/ Graph coloring problem : general backtracking method and recursive back tracking method and their comparison for space and time complexity.
8. A complete LC branch and bound algorithm for job sequencing with dead lines problem. Use fixed tuple size formulation.
9. Algorithm implementation for `Traveling salesman' problem using -
   (a) Dynamic programming approach.
   (b) Branch & Bound approach.
10. Simulation/ Implementation of any Parallel Algorithms.

**Outcomes:**

On successful completion of this course, the student will be able to

1. analyze the average- and worst-case performance of algorithms,
2. use dynamic programming and the greedy paradigm effectively,
3. apply the concept of NP-completeness and be familiar with approximation algorithms
4. implement the families of algorithms in the appropriate high-level language.

**Text Books**


1. Horowitz/Sahani, "Fundamentals of computer Algorithms", Galgotia.
2. Bressard, "Fundamental of Algorithm." ,PHI


**Reference Books**


1. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm" ,PHI
2. A. V. Aho and J.D. Ullman, "Design and Analysis of Algorithms", Addison Wesley

## CS6044 : PROJECT WORK

**Teaching Scheme** **Examination Scheme**

**Practical : 02 Hrs**

**Aim**

This course addresses the issues associated with the successful management of a software development project. The course emphasizes five project processes: initiating, planning, executing, controlling and closing. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and their career.

**Objectives**

1. To provide an understanding of the project planning process and problem scoping
2. To define roles and work with cross functional teams
3. To establish clear project objectives and milestones
4. To create effective and deliverable project plans
5. To perform rigorous analysis and design activities for solution planning
6. To select appropriate technology for implementation and testing

**Overview of the Course:**

1. The Student Project Group is expected to make a survey of situation for identifying the Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting Requirement Definition Document, Literature Survey and Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
3. The Project Work will be assessed jointly by a panel of examiners having more than Five Years experience. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.
4. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work, understanding of the Project, analysis and design performed for the project.
5. The Student Project Groups are expected to work on the recommendations given by the panel of examiners. In no case any variation in Project Theme will be permitted.

**Outcomes:**

Upon completion of the course, the students will be to:

1. Identify major issues in complex situations; and know how to prepare alternative solutions and make decisions.
2. Becoming aware of the need to think and act in an entrepreneurial manner by developing the ability to: critically and creatively understand innovations and development, work independently and collaboratively.

**Note:**

The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members.

The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The Project Synopsis should not be more than THREE to FOUR pages. The Project Group needs to follow the Development Cycle as communicated to them.

## Seminar (Engineering Based)

**Teaching Scheme** **Examination Scheme**

**Aim:**
This course aims to prepare students in communication practices used in professional environments. The course focuses on developing technical writing and speaking which will enable the students to bear adequate skills required in global workplace.

**Objectives:**
1. To introduce technical communication as a practice
2. To strengthen your understanding of, and ability to apply, communications strategies.
3. To learn importance of various types of media and its effective usage
4. To learn effective document preparation activities with Completeness, Conciseness, Concreteness, Clarity and Correctness characteristics
5. To design and use tables, graphs, and technical illustrations.

**Overview of the Course:**
The student will select and explore a topic of Computer Engineering or Information Technology with an interest in learning about the document design process. The student will prepare a Seminar Report based on information gathered and present the work to panel of examiners. The panel of examiners will evaluate the student's performance on the basis of technical contents, organization of the work, communication skills and ability to address audience queries. The Seminar Report need to contain following information:
1. Seminar Title Page
2. Seminar Abstract
3. Table of Contents, List of Figures, List of Tables
4. Introduction to the topic consisting historical perspective and theme
5. Literature Survey and Findings
6. Technical Contents of the topic: Methodologies, Algorithms, Analytical / Mathematical Models, Empirical Experimental Results
7. Conclusion
8. References

**Outcomes:**
Upon completion of this course, the student should be able to
1. Trace the communication situation based on needs and audiences.
2. Develop document information logically and efficiently
3. Select the appropriate media and format for presenting information.
4. Evaluate their documents to be sure that the documents fulfill their purpose and to ensure that they can be revised if necessary.
5. Write specific kinds of documents in technical and scientific communities.

# FOURTH YEAR

# COMPUTER ENGINEERING

# SEMESTER 1

# ELECTIVE 2

# THEORY

# CS1314 : ANALYSIS OF ALGORITHMS.

**Prerequisites:**
Computer Fundamentals and knowledge of 'C' language

**Aim:**
This course introduces basic algorithmic techniques, time requirements of an algorithm and mathematical techniques used in analysis of algorithms. The emphasis will be learning analysis of algorithms for a wide variety of foundational problems occurring in computer science applications with discussions on complexity and NP-completeness.

**Objectives:**
1. Fundamental understanding of the mathematics used to analyze, evaluate, and design algorithms
2. Develop the ability to assess the advantages and disadvantages of different types of algorithms.
3. Understand methods for designing time and space efficient algorithms.
4. Increased ability to design and implement efficient solutions to problems.

**Unit 1 :Overview of Linear, Non-linear Data Structures, Static and Dynamic memory allocation. ( 07 Hrs )**

Linear data structures: Single dimensional and multidimensional Arrays, stacks, queues. Non linear data structures : Trees, Graphs. Static and Dynamic memory allocation : Linked lists , singly linked list, doubly linked list, circular linked list, Insertion, Deletion and traversal on above data structures.

**Unit 2 : Algorithms Analysis (6 Hrs)**
Introduction to Algorithms: Definition, Characteristics of Algorithm, General guidelines for creating good programs, Algorithm analysis: Frequency count, Time Complexity, Space Complexity: Worst Case, Best Case and Average Case. Big-Oh notation. Theta and Omega notations.
Problems solved using Brute Force : Bubble sort, Selection Sort

**Unit 3: Divide And Conquer Strategy: (5 Hrs)**
Divide and Conquer: General Strategy, Exponentiation. Binary Search, Quick Sort and Merge Sort.

**Unit 4 : Greedy Method** **(7 Hrs )**

General Strategy, Knapsack problem, Job sequencing with Deadlines, Optimal merge patterns, Minimal Spanning Trees and Dijkstra's algorithm

**Unit 5 : Dynamic Programming** **(7 Hrs )**

General Strategy, Multistage graphs, Change coins problem, 0/1 Knapsack, Traveling Salesperson Problem, Flow Shop Scheduling

**Unit 6 : Backtracking and P-NP Theory** **(10 Hrs )**

General Strategy, 8 Queen's problem, Graph Coloring, Hamiltonian Cycles, 0/1 Knapsack.

P and NP. Examples of NP-Hard And NP-Complete Problems,
**Algorithms, Complexity-intractability, Non-Deterministic Polynomial time (NP)**

**Outcomes:**

On successful completion of this course, the student will be able to

1. analyze the average- and worst-case performance of algorithms,
2. use the various strategies effectively,
3. apply the concept of NP-completeness and be familiar with approximation algorithms
4. Implement the families of algorithms in the appropriate high-level language.

**Text Books**


1. Horowitz, Sahani, "Fundamentals of computer Algorithms", Galgotia. 2nd Edition, 1998.
2. Bressard, Bratley "Fundamentals of Algorithmics." ,PHI, 2nd Edition,1996.


**Reference Books**


1. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm" ,PHI 2nd edition, 2001.
2. V. Aho and J.D. Ullman, "Design and Analysis of Algorithms", Addison Wesley. 2nd edition.

# CS1324:Software Engineering

**Prerequisites:**
1.Data Structures and Algorithms
2.Object Oriented Programming

**Aim:**
1.Software engineering is aimed at creating practical, cost-effective solutions to computing and information processing problems, preferentially by applying scientific knowledge, developing software systems in the service of mankind.
2.This course covers the fundamentals of software engineering, including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools.
3.The course will combine a strong technical focus with a capstone project providing the opportunity to practice engineering knowledge, skills, and practices in a realistic development setting.

**Objectives:**
1.To learn the complete Software life cycle and understand its major activities such as software requirement analysis, design, testing, and implementation.
2.An understanding of different software processes and how to choose between them.
3.Understanding and Experience in Writing Requirements and Specifications.
4.Introducing the various design approaches, models and metrics.
5.Understanding of Software Management including Planning/scheduling.
6.Presenting the various techniques of software cost estimation and risk assessment.

**Unit 1 : Software Process Models ( 7 Hrs )**

Overview of Software Engineering, Software Process Framework, Process Patterns, Personal and Team Process Models, Process Models: Waterfall Model, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Agile process, Extreme Programming, Process Assessment, CMMI, Impact of Processes and Outcomes, Process Selection and applicability, Software Engineering Principles and Practices

**Unit 2 : Requirements Engineering ( 7 Hrs )**

Requirements Engineering Tasks, Requirement Elicitation Techniques, Software Requirements: Functional, Non-Functional, Domain, Requirements Characteristics and Characterization, Requirement qualities, Requirement Specification, Requirement Traceability, System Analysis Model Generation, Requirement Prioritization, Context Models, Behavioral Models, Data Models, Object Models, Structured Methods

**Unit 3 : Design Engineering ( 7 Hrs )**

Design quality, Design Concepts, The Design Model, Introduction to Pattern-Based Software Design, Architecture styles, Reference Architectures
Architectural Design: Software Architecture, Data Design and Architectural Design, User Interface Design: Rules, User Interface Analysis and Steps in Interface Design, Design Evaluation

**Unit 4 : Principles of Testing ( 7 Hrs )**

Testing Concepts: Purpose of Software Testing, Testing aspects: Requirements, Test Scenarios, Test cases, Test scripts/procedures, Strategies for Software Testing, Testing Activities, Mistakes, Faults & Failures, Software Inspections
White-Box Testing: Test Adequacy Criteria, Static Testing, Structural Testing, Code Complexity Testing, Mutation Testing
Black-Box Testing**:** Test Case Design Criteria, Requirement Based Testing, Positive and Negative Testing, Boundary Value Analysis, Equivalence Partitioning State Based Testing, Compatibility Testing, User Documentation Testing, Domain Testing

**Unit 5 : Project Planning and Estimation ( 7 Hrs )**

Project Management Activities, Structures and Frameworks, Project Planning, Project Scheduling, Risk Analysis, Critical Path, Introduction to Function Points, Empirical Estimation, COCOMO II model, Software Measurement Framework, Ishikawa's Seven tools, Process Assessment and patterns

**Unit 6 : Configuration Management ( 7 Hrs )**

Configuration Management Planning, Change Management, Version and Release Management, System Building, Process and Product Quality, Quality Assurance and standards, Quality Planning, Quality Control

**Outcomes:**
Upon completion of this course, the student should be able to

1. Use the appropriate methods and tools for estimating software cost.
2. Identify the difference between different software design models and techniques and how to apply them.
3. Understand the principles and techniques underlying the process of inspecting and testing software and making it free of errors and tolerable.
4. Recognize the importance of software standards and quality assurance.
5. Apply the appropriate software evolution methods and development.

**Text Book**


1. Ian Sommerville, 'Software Engineering', 7th Edition, Addison-Wesley, 2004,ISBN 81-7758-530-4
2. Roger S Pressman, 'Software Engineering: A Practitioner's Approach', 6/e, McGraw Hill, 2005, ISBN: 0072853182


**Reference Books:**


1. Grady Booch, James Rambaugh, Ivar Jacobson, 'Unified Modeling Language Users Guide', 2nd Edition, Addison- Wesley, ISBN – 0321267974.
2. Jim Arlow, Ila Neustadt, 'UML 2 and Unified Process: Practical Object Oriented Analysis and Design.', 2nd Edition, Addison- Wesley, ISBN – 0321321278.
3. Tom Pender, 'UML Bible', John Wiley & sons, ISBN – 0764526049.
4. Desikan, Ramesh, 'Software Testing: principles and Practices', Pearson Education, ISBN 81-7758-121-X.
5. Burnstein, 'Practical Software Testing', Springer International Edition, ISBN 81-8128-089-X
6. William E. Perry, 'Effective Methods for Software Testing', John Wiley and Sons, ISBN 9971-51-345-5
7. Stephen H. Kan, 'Metrics and Models in Software Quality Engineering', Pearson Education, ISBN 81-297-0175-8

# CS1334: Database Management System

**Aim :**
This course focuses on fundamentals of database architecture, database management systems, database systems, data modeling to systems design to logical models and computational complexity. The course is principally aimed at database design and use of database management systems in implementing database applications.

**Objectives:**
1. To interpret an entity relationship diagram (ERD) to express requirements and demonstrate skills to model data requirements and create data models into normalized designs
2. To use SQL to create database objects, populate tables, and retrieve data
3. To describe the causes of performance problems and how to improve database application performance
4. To understand approaches and trade-offs in the design and development of database systems.
5. To develop a working understanding of database systems theory in order to apply that knowledge to any particular database implementation.

**Unit 1 : Introduction ( 5 Hrs )**

Introduction to data structures in brief, Basic Concepts: File processing system, Need of DBMS, data, database, database systems, database management system, data abstraction, data independence, overall system architecture of DBMS

**Unit 2 : Data modeling ( 8 Hrs )**

ER modeling: Entity , entity set, attributes, relationship type, relationship set, relationship instance, role, recursive relationship, cardinality ratio, participation constraint, attributes of relationship types, weak entity type, the identifying entity type, the identifying relationship Entity relationship Diagram, Extended E-R features. Design of an E-R schema for a realistic problem. Schema Diagram

**Unit 3 : Relational Data Model ( 10 Hrs )**

Relational structure - tables (relations), rows (tuples), domains, attributes, keys, super key, candidate keys, primary key, entity integrity constraints, referential integrity constraints; ER- to - relational mapping Relational algebra, Normalization !NF, 2NF and 3NF

**Unit 4 : Relational database implementation using SQL ( 6 Hrs )**

DDL, DML, DCL, simple and Nested queries, sub queries, Authorization in SQL

**Unit 5 : Transaction management ( 4 Hrs )**

Transaction concept, ACID properties, schedule and recoverability, serializability, cascadeless schedule, concurrency control and protocols: lock based, timestamp based, tree protocol, recovery systems

**Unit 6 : Current trends ( 6 Hrs )**

Introduction to data warehouse, properties and use of data warehouse, introduction to data mining, text mining, multimedia databases, Spatial and temporal databases

**Outcomes:**

Upon completion of this course, the students will be able to:

1. understand user requirements/views and analyze existing and future data processing needs with data model development
2. develop and refine the conceptual data model, including all entities, relationships, attributes with integration and merging database views into conceptual model
3. apply normalization techniques with identification of data integrity and security requirements

**Reference Books**


1. Silberschatz, Korth and S.Sudarshan,"Database system Concepts", McGraw-Hill international edition, Fifth edition, 2006.
2. Elmasri and Navathe, "Fundamentals of Database systems", Addison Wesley, second edition, 1994.
3. Ramakrishnan and Gehrke, "database management systems", McGraw-Hill, international edition, third edition, 2003.

# CS1344:Operating Systems

**Aim:**
To understand the basic theoretical concepts involved in the design of Operating Systems

**Objectives:**
1. To study the operations performed by Operating Systems as a resource manager.
2. To learn the evolution of Operating Systems.

**Unit 1 : Introduction to OS ( 7 Hrs )**

Architecture, Goals & Structures of O.S., Hardware Abstraction layer, Basic functions, Interaction of OS and hardware architecture, System Calls & OS services, Batch, multiprogramming, multitasking, time sharing, parallel, distributed & real-time OS.

Examples of OS: Linux and variants, MS-Windows 2000

**Unit 2 : Process Management ( 7 Hrs )**

Process description & control: Process Concept, Process states, Process description, Process control, Threads Concurrency: Principles of Concurrency, Mutual Exclusion: Semaphores, Message Passing, Monitors, Classical Problems of Synchronization: Readers-Writers problem, Producer Consumer Problem, Dining Philosopher problem

**Unit 3 : Deadlock ( 7 Hrs )**

Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery

**Unit 4 : Scheduling ( 7 Hrs )**

Uniprocessor Scheduling: Types of scheduling: Preemptive, Non-preemptive, Long-term, Medium-term, Short-term. Scheduling Algorithms: FCFS, SJF, RR, Priority

Multiprocessor Scheduling: Granularity, Design Issues, Process Scheduling, Thread Scheduling, Real Time Scheduling: Characteristics, Real Time Scheduling

**Unit 5 : Memory Management** **( 7 Hrs )**

Memory management requirements, Memory partitioning: Fixed and Variable Partitioning, Memory Allocation: Allocation Strategies (First Fit, Best Fit, Worst Fit), Fragmentation, Swapping.

Virtual Memory: Concepts, Segmentation, Paging, Address Translation, Demand paging, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies), Thrashing

**Unit 6** : **I/O Devices & Files** **( 7 Hrs )**

I/O management & Disk scheduling: I/O Devices, Organization of I/O functions, Operating System design issues, I/O Buffering, Disk Scheduling (FCFS, SCAN, C-SCAN, SSTF), RAID, Disk Caches.
File Management: Concepts, File Organization, File Directories, File Sharing, Record Blocking, Free Space management, Secondary Storage Management

**Outcomes:**

Student from a non Computer Engineering background will be able to

apply the basic concepts in industry. Since IT industry has a need for engineering students from other domain areas to also have basic knowledge of foundation courses in computer engineering, this course will equip the student in gaining this knowledge.

**Text Books:**


3. Stalling William, "Operating Systems", Pearson Education,4th Edition ,2001.
4. Silberschatz A., Galvin P., Gagne G., "Operating System Concepts", John Wiley and Sons, 7th Edition, 2003.


**Reference Books:**


4. Tanenbaum Andrew S., "Modern Operating Systems" PHI, 2nd Edition, 2001.
5. Das Sumitabha, "Unix Concepts and Applications", Tata McGraw Hill, 3rd Edition, 2003.

# CS1354: Computer Networks

**Prerequisites:**
Principles of Communications Engineering

**Aim :**
This course introduces fundamental concepts and principles of computer communication networks, from the primary perspective of the TCP/IP Internet framework.This course provides a detailed examination of the conceptual framework for modeling communications between processes residing on independent hosts, and the rules and procedures that mediate the exchange of information between two communication processes.

**Objectives:**
1. To understand some of the common data link layer protocols used in the Internet.
2. To learn how IP datagrams are handled by routers.
3. To understand the two basic transport protocols, UDP and TCP -- what they do, how they work, when each is appropriate for use by applications, and issues with their use.
4. To use the OSI Reference model to identify the services required for communications to take place between processes on autonomous hosts.

**Unit 1 : Introduction ( 8 Hrs )**

**Modulation techniques**: Principle of amplitude modulation, modulation index and percentage of modulation, Single sideband communication, ISB modulation, frequency modulation principle, phase modulation, AM vs FM, pulse code modulation, delta modulation
**Multiplexing**: Introduction, FDM, TDM, WDM
Transmission media: guided, unguided
Reference Models: OSI and TCP/IP,

**Unit 2 : Data Link Layer (5 Hrs )**
Design Issues, Error Detection and correction, Sliding Window protocols, HDLC
**Point-to-Point-Access** (PPP): Frame format, Transition states, PPP Stack: LCP, NCP

**Unit 3 : Medium Access Control ( 5 Hrs )**
Channel allocation: Static and Dynamic allocation, Multiple Access Protocols: ALOHA, CSMA, Collision-free and limited-contention protocols, WDMA.
Ethernet: Cabling, encoding, MAC sub-layer protocol, Switched, fast and Gigabit Ethernet, Logical link control

**Unit 4 : Network Layer ( 8 Hrs )**

Design Issues, Packet switching, Connectionless and Connection-oriented Services, Virtual Circuit and Datagram Subnets, Routing Algorithms.

**Congestion Control and QOS:** General Principles, Congestion prevention policies, Load shading, Jitter Control, Quality of Service, Internetworking.

**Network layer Protocols:** ARP,RARP, IP protocol, IP Addresses, IPV4, IPV6, ICMP, Unicast Routing Algorithms: RIP, OSPF, BGP, Multicast Routing: IGMP, Mobile IP.

**Unit 5 : Transport Layer ( 8 Hrs )**

**Transport Layer:** Services and service primitives, Elements of Transport protocol: Addressing, Connection establishment and release, flow control and buffering, Multiplexing, Crash recovery, Simple Transport Protocol, UDP: Introduction, TCP: Introduction, protocol header, connection establishment and release, connection management, Transmission policy, congestion control, timer management, Sockets Introduction to wireless TCP and UDP

**Unit 6 : : Application Layer ( 8 Hrs )**

Domain Name System (DNS) and DNS servers, Electronic Mail: Architecture and services, MIME, SMTP, Mail Gateways, Remote login, File Transfer Protocol,

**World Wide Web:** Introduction, Architectural overview, static and dynamic web pages, HTTP, LDAP, Browser Architecture, Wireless Web

**Outcomes:**

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

1. Name, and list the major functions of, each of the layers of the ISO and the Internet protocol stacks.
2. Understand the movement of bits through a medium and determine the transmission time and propagation time, given the transmission speed, packet size, medium length, and propagation speed.
3. Describe, basically, the operation of hubs, switches, and routers, and how and why each is used in a network.
4. Describe, basically, the operation of, and the services provided by, the two Internet transport protocols, TCP and UDP.
5. Describe, basically, the operation of the Network Layer routing protocols

**Text Books**


1. Tanenbaum A. S., "Computer Networks", 4th Edition, Pearson Education, ISBN 81 – 7808 – 785 – 5
2. Forouzan B. A, "Data Communications and Networking", 4th edition, Tata McGraw-Hill Publications, 2006, ISBN 0 – 07 – 063414 – 9

**Reference Books**


1. James F. Kurose, "Computer Networking- a top-down approach featuring the internet" , 2nd Edition, Person Education, ISBN 81- 7808-787-1
2. Leon-Garcia-Wadjaja, "Communication Networks- Fundamental Concepts and Key Architectures", Tata McGraw-Hill Publications, ISBN 0-07-040235-3
3. Comer D., "Computer Networks and Internet", 2ND Edition, Pearson Education, ISBN 81 – 7808 – 086 – 9

# CS1364: MULTIMEDIA SYSTEMS

**Aim :**
This course provides the design and development of computer-based systems that combine text, still images, sound, animation, and full motion video. This course covers the state-of-the-art technology for multimedia systems with essential aspects of the different media types images, video, audio, graphics etc and how they are used to create multimedia content, compress and distribute them via networked system to variety of end clients.

**Objectives:**

1. To introduce students to the different media used and the design issues in multimedia systems with understanding of multimedia programming.
2. To train students in the application of suitable evaluation techniques for multimedia systems.
3. Provide an opportunity for students to apply design, implementation and evaluation concepts and techniques to the development of a realistic multimedia system.

**Unit 1 Introduction to Multimedia, Multimedia authoring tools and Multimedia devices:**

**( 7 Hrs)**

Introduction to multimedia, Multimedia architecture, Windows multimedia support, hardware support, distributed multimedia applications, streaming technologies, and multimedia database systems

Multimedia authoring and multimedia devices : Overview of multimedia, multimedia building blocks, multimedia authoring, different authoring tools ,Overview of devices such as magnetic devices, optical devices, scanners, CRT display their controllers, video display adapters, graphics device drivers and display buffers

**Unit 2 Graphics: ( 7 Hrs)**

Introduction to computer graphics, lines, line segments, vectors, pixels and frame buffers, vector generation, Generation of line using DDA and Bresenham's line drawing algorithm, Generation of circle using Bresenham's algorithm, Polygon scan line seed filling algorithm, 2D Line clipping using Cohen Sutherland algorithm and Lang Barky algorithms

**Unit 3. 2D & 3D Transformation: ( 7 Hrs)**

2D transformations: Translation, rotation, scaling, reflection and shear

3D transformations translation, scaling, rotation about any axis not parallel x, y or z axis,

Introduction to rendering, illumination models,Gourad and Phong shading, aliasing and antialiasing techniques in graphics

**Unit 4. Audio and Audio Compression: ( 7 Hrs)**

Basic concepts of audio, digitization of sound, sound processing, Elements of audio systems: Microphone, amplifiers, speakers, synthesizer, CD formats,

Audio file formats WAV, VOC, AIFF and MIDI, audio compression techniques such as DM, ADPCM and MPEG

**Unit 5. Animation and Video: ( 7 Hrs)**

Principles of animation, Uses of animation, types of animation, techniques of animation: Onion Skinning, Motion Cycling, masking, Flip Book animation, Rotoscoping & blue-screening, color cycling, morphing, animation on the web, 3D animation, Creating animation.

Fundamentals of still images color models of images Fundamentals of video, color models in video, Introduction to analog and digital video

**Unit 6. Compression Techniques and Graphics File Format: ( 7 Hrs)**

Introduction and need for data Compression, broad categories of compression techniques. Types of Lossless compression techniques such as Huffman and LZW and Run length encoding, lossy compression techniques and hybrid compression technique: case study of JPEG

Study of Graphics file format: BMP, JPEG, GIF, TIFF

**Outcomes:**

Upon completion of the course, the student will:

1. Understand the characteristics of multimedia systems and how to address issues
2. Be aware of the differences among multimedia authoring systems.
3. Be familiar with the software development process as practiced in a multimedia development environment
4. Identify the media used in multimedia systems and to assess their relative advantages and disadvantages relative to both user and system points of view.
5. Explain the interaction problems introduced by multimedia (e.g., compression and synchronisation) and to be able to enumerate and critique the techniques available for ameliorating or removing these problems.
6. Design, implement and evaluate a small multimedia system.

**Text Books:**


1. Li Ze-Nian, Drew Mark S., "Fundamentals of Multimedia", Pearson Education, 2004, ISBN 81-297-0438-2
2. Rogers David F., "Procedural Elements for Computer Graphics", Tata McGraw Hill, 2001, ISBN 0-07-047371-4


**Reference Books:**


1. Harrington Steven, "Computer graphics: A Programming approach", Tata McGraw Hill, ISBN 0-07-026753-7
2. M Paulin Baker, Donald Hearn, "Computer Graphics", PHI, 2001, ISBN 81-203-09440-8
3. Steinmetz Ralf, Nahrstedtk, "Multimedia Computing, Communication and Applications", Pearson Education, 2004, ISBN 8178083191
4. Born Gunter, "Handbook of file formats ", Imprints Of International Thomson Learning, 1997, 1-85032-117-5

# FOURTH YEAR

# COMPUTER ENGINEERING

# SEMISTER II

# THEORY

# CS1044: Software Design and Architecture

**Prerequisites :**
1. Software Engineering
2. Object Oriented Programming
3. Object Oriented Modeling and Design

**Aim :**
1. This course is based around the notion of a design pattern: an abstraction of a proven solution to a recurring problem in a particular context in system structure and behavior.
2. The students are expected to use the language of patterns to find and to record solutions to recurring problems of system architecture with practical experience of a number of the best and most useful patterns

**Objectives:**
1. To describe what patterns are, how to use them, and why they are important.
2. To understand contribution of patterns towards the design process efficiency
3. To trace the relationship between patterns and traditional methods
4. To learn the relationship among patterns, frameworks, object-oriented languages, and software architecture
5. To identify fundamental and advanced concepts of design and architectural patterns
6. To Structure systems by applying architectural patterns
7. To focus on Archetypes and Architecture Documentation

**Unit 1 : Introduction to Software Architecture ( 7 Hrs )**

Software Architecture, Relationships to Other Disciplines, Multi-Disciplinary Overview, Foundations of Software Architecture, Software architecture in the context of the overall software life cycle, Architectural Styles, CASE study of Architectures

**Unit 2 : Software Architecture Design ( 7 Hrs )**

Designing, Describing, and Using Software Architecture, IS2000: The Advanced Imaging Solution, Global Analysis, Conceptual Architecture View, Module Architecture View, Styles of the Module Viewtype, Execution Architecture View, Code Architecture View. Component-and-Connector Viewtype, Styles of Component-and-Connector Viewtype, Allocation Viewtype and Styles, Documenting Software Interfaces, Documenting Behavior, Choosing the Views, Building the Documentation Package

**Unit 3 : Archetype Patterns** **( 7 Hrs )**

Archetypes and Archetype Patterns, Model Driven Architecture with Archetype Patterns. Literate Modeling, Archetype Pattern. , Customer Relationship Management (CRM) Archetype Pattern, Product Archetype Pattern, Quantity Archetype Pattern, Rule Archetype Pattern

**Unit 4 : Design Patterns and Patterns Types** **( 7 Hrs )**

Design Patterns: Creational Patterns, Patterns for Organization of Work, Access Control Patterns, Service Variation Patterns, Service Extension Patterns
Pattern Types: Object Management Patterns Adaptation Patterns, Communication Patterns, Architectural Patterns, Structural Patterns, Patterns for Distribution, Patterns for Interactive Systems , Adaptable Systems, Frameworks and Patterns, Analysis Patterns

**Unit 5 : Enterprise Architecture Integration** **( 7 Hrs )**

Defining EAI, Data-Level EAI, Application Interface-Level EAI, Method-Level EAI. , User Interface-Level EAI, The EAI Process, An Introduction to EAI and Middleware, Transactional Middleware and EAI, RPCs, Messaging, and EAI, Distributed Objects and EAI, Database-Oriented Middleware and EAI, Java Middleware and EAI, Implementing and Integrating Packaged Applications, XML and EAI, Message Brokers, Process Automation and EAI.

**Unit 6 : Enterprise Architecture Patterns** **( 7 Hrs )**

Layering, Organizing Domain Logic, Mapping to Relational Databases, Web Presentation, Domain Logic Patterns, Data Source Architectural Patterns, Object-Relational Behavioral Patterns, Object-Relational Structural Patterns, Object-Relational Metadata Mapping Patterns, Web Presentation Patterns, Distribution Patterns, Offline Concurrency Patterns

**Outcomes:**

Upon completion this course, students will be able to:
1. Recognize and understand the various aspects of architectures
2. Recognize, understand and use the most common architectural patterns
3. Effectively document architectures
4. Identify the various customers of architecture

**Text Books**


1. Christine Hofmeister, Robert Nord, Deli Soni, “Applied Software Architecture”, Addison-Wesley Professional; 1st edition (November 4, 1999) ,ISBN-10: 0201325713 , ISBN-13: 978-0201325713
2. Ian Gorton Springer,” Essential Software Architecture”, 1 edition (2006), ISBN-10: 3540287132, ISBN-13: 978-3540287131

**Reference Books**


1. Documenting Software Architectures: Views and BeyondPaul Clements, *Software Engineering Institute,* Felix Bachmann Len Bass, *Software Engineering Institute* David Garlan James Ivers Reed Little Robert Nord Judith Stafford Publisher: Addison-Wesley Professional 2003, ISBN-10: 0201703726, ISBN-13: 9780201703726
2. Frank Buschmann, Hans Rohnert, Kevin Henney, Douglas C. Schmidt ,"Pattern-Oriented Software Architecture Volume 1, 2, 3, 4, 5", Publisher: Wiley, 1 edition (August 8, 1996-2004), ISBN-10: 0471958697, ISBN-13: 978-0471958697
3. Erich Gamma, Richard Helm, Ralph Johnson, ,"Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
4. Martin Fowler ,"Patterns of Enterprise Application Architecture", Addison-Wesley Professional, 2003, ISBN-10: 0321127420 ISBN-13: 9780321127426
5. Fred A. Cummins ,"Enterprise Integration: An Architecture for Enterprise Application and Systems Integration", Wiley; 2002 ISBN-10: 0471400106 ISBN-13: 978-0471400103

# CS1054 : Computer Architecture

**Prerequisites:**
Understanding of Microprocessors, Computer Organization

**Aim**:
To help the student develop an understanding of the nature and characteristics of the organization and design of modern computer systems.

**Objectives:**
1. To understand the key concepts that are likely to be included in the design of any modern computer system
2. To understand and evaluate the impact that languages, their compilers and underlying operating system(s) have on the design of computer systems

**UNIT 1:** **(7 Hrs)**

**Overview of Parallel Processing and Pipelining Processing**

Necessity of high performance, Constraints of conventional architecture, Parallelism in uniprocessor system, Evolution of parallel processors, future trends, Architectural Classification, Applications of parallel processing, Instruction level Parallelism and Thread Level Parallelism, Explicitly Parallel Instruction Computing (EPIC) Architecture, Case study of Intel Itanium Processor

**Principles of scalable performance:** Performance Metrics and Measures, Speedup Performance Laws.

**UNIT 2** **(7 Hrs)**

**Pipeline Architecture**

Principles and implementation of Pipelining, Classification of pipelining processors, General pipelining reservation table, Design aspect of Arithmetic and Instruction pipelining, Pipelining hazards and resolving techniques, Data buffering techniques, Job sequencing and Collision, Advanced pipelining techniques, loop unrolling techniques, out of order execution, software scheduling, trace scheduling, Predicated execution, Speculative loading, Register Stack Engine, Software pipelining, VLIW (Very Long Instruction Word) processor, Case study: Super scalar Architecture- Pentium, Ultra SPARC

**UNIT 3:** **(7 Hrs)**

**Vector and Array Processor**

Basic vector architecture, Issues in Vector Processing, Vector performance modeling, vectorizers and optimizers, Case study: Cray Arch.

**SIMD Computer Organization** Masking and Data network mechanism, Inter PE Communication, Interconnection networks of SIMD, Static Vs Dynamic network, cube hyper cube and Mesh Interconnection network.

**Parallel Algorithms For Array Processors:** Matrix Multiplication. Sorting, FFT

**UNIT 4: (7 Hrs)**

**Multiprocessor Architecture**

Loosely and Tightly coupled multiprocessors, Processor characteristics of multiprocessors, Inter Processor communication network, Time shared bus, Crossbar switch, Multiport Memory Model, Memory contention and arbitration techniques, Cache coherency and bus snooping, Massively Parallel Processors (MPP), Cow's and NOW's Cluster and Network of Work Stations), Chip Multiprocessing (CMP), Case Study of IBM Power4 Processor Inter Processor Communication and Synchronization

**UNIT 5: (7 Hrs)**

**Multithreaded Architecture** Multithreaded processors, Latency hiding techniques, Principles of multithreading, Issues and solutions.

**Parallel Programming Techniques** "; Message passing program development, Synchronous and asynchronous message passing , Message passing parallel programming, Shared Memory Programming, Data Parallel Programming

**UNIT 6: Parallel Software Issues (7 Hrs)**

a) Parallel algorithms for multiprocessors, classification of parallel algorithms, performance of parallel algorithms

b) Operating systems for multiprocessors systems, Message passing libraries for parallel programming interface, PVM (in distributed memory system), Message Passing Interfaces (MPI), Threads (in shared memory system)

c) Parallel Programming Languages : Fortan 90, Occam, C-Linda, CCC etc.

d) Issues towards cluster computing. Introduction to Neuro Computing and Grid Computing .

**Outcome:**

On completion of this course, a student should be able:

1. To understand processor features including RISC/CISC, caching, branch prediction, latency hiding, dataflow and SIMD/MIMD multiprocessing as expressed in the iA32 and iA64 architectures

**Text Books**


1. Kai Hwang, Faye A. Briggs, "Computer Architecture and Parallel Processing" McGraw-Hill international Edition
2. Kai Hwang, "Advanced Computer Architecture", Tata McGraw-Hill


**Reference Books:**


1. V. Rajaraman, L Sivaram Murthy, "Parallel Computers", PHI.
2. William Stallings, "Computer Organization and Architecture, Designing for performance" Prentice Hall, Sixth edition.
3. Kai Hwang, Scalable Parallel Computing.
4. Harrold Stone, High performance computer Architecture .
5. Richard Y. Kain, Advanced Computer Architecture
6. http://www.intel.com/products/processor (for Intel Itanium Processor) For IBM Power 4 Processor
7. http://www.ibm.com/servers/eserver/pseries/hardware/whitepapers/power/ppc_arch.html

# CS1034: Compiler Design

**Prerequisites:** Automata Theory: DFAs, NFAs, Regular Expressions

**Aim:**
This course is aimed at elaborating the implementation issues and strategies behind converting a program written in a high-level language into a form that will execute correctly on some target machine architecture. This course will emphasize the implementation of a compiler through the development of its various phases such as the scanner, parser, and code generator.

**Objectives:**
1. To understand the theory and practice of compiler implementation;
2. To learn the application of grammars and formal languages in compiler writing; the process involved in the design of a high-level programming language;
3. To learn lexical analysis, a variety of parsing techniques and semantic analysis of a programming language, along with error detection and recovery;
4. To learn various storage allocation, code generation and code optimization techniques.

**Unit 1: Introduction to Compilers & Lexical Analysis ( 5 Hrs )**

Introduction to Compilers, Assemblers, Pre-processors, Linkers and Loaders, Role of a Lexical Analyzer, Specification and Recognition of Tokens, Implementing Scanners, LEX

**Unit 2: Syntax and Semantic Analysis ( 9 Hrs )**

Expressing Syntax, Top-Down Parsing, Bottom-Up Parsing, LR Parsers, YACC, Type Checking, Type Conversion.

**Unit 3: Syntax-Directed Translation & Intermediate Code Generation ( 7 Hrs )**

Syntax-Directed Definitions, Bottom-Up Evaluation, Top-Down Translation, Intermediate Representations, Intermediate Code Generation

**Unit 4: Runtime Storage Organization ( 7 Hrs )**

Storage Organization, Storage Allocation Strategies, Symbol-Table Structure, Run-Time Support

**Unit 5: Code Generation ( 7 Hrs )**

Issues in Code Generation, Basic Blocks and Flow Graphs, Instruction Selection, Instruction Scheduling, Register Allocation.

**Unit 6: Code Optimization ( 7 Hrs )**

Early Optimizations, Redundancy Elimination, Loop Optimizations, Procedure Optimizations, Control and Data Flow Analysis

**Outcomes:**

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

1. Describe the design of a compiler including its phases and components
2. Develop the various phases of a compiler such as the scanner, parser, code generator, and optimizer
3. Identify the similarities and differences among various parsing techniques

**Text Books**


1. K. Cooper, L. Torczon, 'Engineering a Compiler', Morgan Kaufmann, 1st Edition, 2003
2. A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, 'Compilers: Principles, Techniques, and Tools', Addison Wesley, 2nd Edition, 2007


**Reference Books**


1. K. Louden, 'Compiler Construction: Principles and Practice', Course Technology, 1st Edition, 1997
2. S. S. Muchnik, 'Advanced Compiler Design and Implementation', Morgan Kaufmann, 1st Edition, 1997
3. J. R. Levine, T. Mason, D. Brown, 'Lex & Yacc', O'Reilly, 2nd Edition, 1992

# FOURTH YEAR

# COMPUTER ENGINEERING

# SEMESTER II

# LABORATORY

# CS1044: Software Design and Architecture

**Prerequisites :**

1. Software Engineering
2. Object Oriented Programming
3. Object Oriented Modeling and Design

**Objectives:**

1. To describe what patterns are, how to use them, and why they are important.
2. To understand contribution of patterns towards the design process efficiency
3. To trace the relationship between patterns and traditional methods
4. To learn the relationship among patterns, frameworks, object-oriented languages, and software architecture
5. To identify fundamental and advanced concepts of design and architectural patterns
6. To Structure systems by applying architectural patterns
7. To focus on Archetypes and Architecture Documentation

## List of Practicals

1. Select a moderately complex system and narrate concise specification for the same.
2. Implement the system features using Abstract Factory and Composite design patterns. State the complete pattern specification and note the difference between the patterns.
3. Select a moderately complex system and narrate concise specification for the same.
4. Implement the system features using Facade and Proxy design patterns. State the complete pattern specification and note the difference between the patterns.
5. Select a moderately complex system and narrate concise specification for the same.
6. Implement the system features using Flyweight and Iterator design patterns. State the complete pattern specification and note the difference between the patterns.
7. Select a moderately complex system and narrate concise specification for the same.
8. Implement the system features using Template and Command design patterns.
9. State the complete pattern specification and note the difference between the patterns.
10. Select a moderately complex system and narrate concise specification for the same.
11. Implement the system features using Mediator and Observer design patterns. State the
12. complete pattern specification and note the difference between the patterns.
13. Select a complex system and narrate concise specification for the same. Develop

14. architecture specification and use archetypes to recognize the architectural elements.

**Outcomes:**

Upon completion this course, students will be able to:

1. Recognize and understand the various aspects of architectures
2. Recognize, understand and use the most common architectural patterns
3. Effectively document architectures
4. Identify the various customers of architecture

**Text Books**


1. Documenting Software Architectures: Views and BeyondPaul Clements, *Software Engineering Institute,* Felix Bachmann Len Bass, *Software Engineering Institute* David Garlan James Ivers Reed Little Robert Nord Judith Stafford Publisher: Addison-Wesley Professional 2003, ISBN-10: 0201703726, ISBN-13: 9780201703726
2. Frank Buschmann, Hans Rohnert, Kevin Henney, Douglas C. Schmidt ,"Pattern-Oriented Software Architecture Volume 1, 2, 3, 4, 5", Publisher: Wiley, 1 edition (August 8, 1996-2004), ISBN-10: 0471958697, ISBN-13: 978-0471958697


**Reference Books**


1. Christine Hofmeister, Robert Nord, Deli Soni, "Applied Software Architecture", Addison-Wesley Professional; 1st edition (November 4, 1999) ,ISBN-10: 0201325713 , ISBN-13: 978-0201325713
2. Ian Gorton Springer," Essential Software Architecture", 1 edition (2006), ISBN-10: 3540287132, ISBN-13: 978-3540287131
3. Erich Gamma, Richard Helm, Ralph Johnson, ,"Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) ,John Vlissides, Publisher: Addison-Wesley Professional, 1st edition (January 15, 1995) , ISBN-10: 0201633612 ISBN-13: 978-0201633610
4. Martin Fowler ,"Patterns of Enterprise Application Architecture", Addison-Wesley Professional, 2003, ISBN-10: 0321127420 ISBN-13: 9780321127426
5. Fred A. Cummins ,"Enterprise Integration: An Architecture for Enterprise Application and Systems Integration", Wiley; 2002 ISBN-10: 0471400106 ISBN-13: 978-0471400103

# CS1034 : Compiler Design Laboratory

**Prerequisites:** Knowledge of C Language

**Objectives:**
To design and implement the various phases of a compiler for a concise programming language.

## List of Practicals

Implement the following for a subset of C

1. Lexical Analyzer using LEX
2. Parser using LEX and YACC
3. Generate and populate appropriate Symbol Table (Optional)
4. Semantic Analysis Operations (like type checking, verification of function parameters, variable declarations and coercions), possibly using an Attributed Translation Grammar. (Optional)
5. Generation of appropriate Intermediate Code, possibly using an Attributed Translation Grammar.

Implement the following assuming suitable processor details.

6. Generate appropriate Target Code from the intermediate code given
7. A Register Allocation algorithm that translates the given code into one with a fixed number of registers. (Optional)
8. An Instruction Scheduling Algorithm (Optional)

Implement the following using appropriate Intermediate Code

9. Local and Global Code Optimizations, such as Common Sub-expression Elimination, Copy Propagation, Dead-Code Elimination, Loop and Basic-Block Optimizations (Optional)

**Outcomes:**

The students will be able to develop a complete compiler for any concise programming language.

**Text Books**


1. K. Cooper, L. Torczon, 'Engineering a Compiler', Morgan Kaufmann, 1st Edition, 2003
2. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, 'Compilers: Principles, Techniques, and Tools', Addison Wesley, 2nd Edition, 2007


**Reference Books**


1. J. R. Levine, T. Mason, D. Brown, 'Lex & Yacc', O'Reilly, 2nd Edition, 1992
2. K. Louden, 'Compiler Construction: Principles and Practice', Course Technology, 1st Edition, 1997
3. S. S. Muchnik, 'Advanced Compiler Design and Implementation', Morgan Kaufmann, 1st Edition, 1997

# FOURTH YEAR

# COMPUTER ENGINEERING

# SEMESTER II

# ELECTIVE 1 & 3

# THEORY

# CS1114 : MULTIMEDIA SYSTEMS

**Aim :**
This course provides the design and development of computer-based systems that combine text, still images, sound, animation, and full motion video. This course covers the state-of-the-art technology for multimedia systems with essential aspects of the different media types images, video, audio, graphics etc and how they are used to create multimedia content, compress and distribute them via networked system to variety of end clients.

**Objectives:**

1. To introduce students to the different media used and the design issues in multimedia systems with understanding of multimedia programming.
2. To train students in the application of suitable evaluation techniques for multimedia systems.
3. Provide an opportunity for students to apply design, implementation and evaluation concepts and techniques to the development of a realistic multimedia system.

**Unit 1 : Introduction**: **( 6 Hrs)**
What is multimedia, Goals and objectives, characteristics of multimedia presentation, multimedia applications, Multimedia building blocks, multimedia and internet,
Multimedia architecture, Windows multimedia support, hardware support, distributed multimedia applications, streaming technologies, multimedia database systems, Multimedia authoring tools, overview of multimedia software tools, multimedia Document Architecture,
(MHEG, SGML, ODA, OMF etc.) Text:
Types of text, Text compression: Huffman coding, LZ & LZW, text file formats: TXT, DOC; RTF, PDF, PS.

**Unit 2 : Digital Image Processing** **( 8 Hrs)**
Basic Image fundamentals, Image data types, image File formats,- (BMP, TIFF, JPEG, PCX etc), Image acquisition, storage processing, Communication, and display, Image enhancement: Enhancement by point processing, Spatial filtering.
Image compression: Types of compression: lossy & lossless, symmetrical & asymmetrical, intraframe & interframe Hybrid, Lossless: RLE, Shannon- Fano algorithm, Arithmetic coding. Lossy: Vector quantization, fractal compression technique, transform coding, psycho-analysis, and interframe correlation. Hybrid: JPEG-DCT

**Unit 3 : Multimedia Audio** **( 6 Hrs)**
Nature of sound waves, characteristics of sound waves, psycho-acoustic, and elements of audio systems: Microphone, amplifiers, speakers, synthesizer, MIDI, digital audio, CD formats. Audio file formats:
WAV, AIFF, VOC, AVI, MPEG Audio File formats, RMF, WMA etc.

**Unit 4 : Video** **( 8 Hrs)**
Video signal formats, Video transmission standards: EDTV, CCIR, CIF, SIF, HDTV, digitization of video, video recording systems: VHS, Video Compact Cassette, DVCAN, Camcorder, Lesser disc, VCD, DVD-video, micro-MV, Video file formats: MOV, RealVideo, H-261, H-263, cinepack. Nerodigtal, Video editing, DVD formats

**Unit 5 : Virtual Reality and Multimedia (6 Hrs)**

Concept, Forms of VR, VR applications, VR devices: Hand Gloves, Head mounted tracking system, VR chair, CCD, VCR, 3D Sound system, Head mounted display. Virtual Objects
Basics of VRML.

**Unit 6 : Animation ( 8 Hrs)**

Uses of animation, types of animation, principles of animation, Techniques of animation: Onion Skinning, Motion Cycling, masking, Flip Book animation, Rotoscoping & blue-screening, color cycling, morphing, animation on the web, 3D animation, Creating animation using Flash,3D-Max

## Outcomes:

Upon completion of the course, the student will:

1. Understand the characteristics of multimedia systems and how to address issues
2. Be aware of the differences among multimedia authoring systems.
3. Be familiar with the software development process as practiced in a multimedia development environment
4. Identify the media used in multimedia systems and to assess their relative advantages and disadvantages relative to both user and system points of view.
5. Explain the interaction problems introduced by multimedia (e.g., compression and synchronisation) and to be able to enumerate and critique the techniques available for ameliorating or removing these problems.
6. Design, implement and evaluate a small multimedia system.

## Text Books


1. Ranjan Parekh, "Principles of Multimedia",TMH, ISBN 0-07-058833-3
2. Ralf Steinmetz and Klara Nahrstedt "Multimedia Computing, Communication and Applications", Pearson Education.


## Reference Books


1. Ze-Nian Li, Marks S. Drew, "Fundamentals of Multimedia", Pearson Education.
2. Nigel Chapman and Jenny Chapman. Wiley "Digital Multimedia"
3. A. K. Jain "Fundamentals of Digital Image Processing", PHI
4. Gonzalez, Woods, "Digital Image Processing" Addison Wesley
5. Mark Nelson "Data Compression Book ", BPB.
6. Judith Jeffcoate "Multimedia in Practice":, Pill.
7. Robert Reinhardt, Snow Dowd, “Flash 8 Bible”
8. Keith Peters, “Foundation AS Animation: Making Things Move!”
9. Sanford Kennedy, “3ds max Animation and Visual Effects Techniques”

# CS1124 :Mobile Computing

**Aim :**
This course provides an in depth understanding of wireless access and core networks and mobility in cellular and wireless networks using the important standards of the industry like GSM, CDMA, GPRS and IEEE 802.11 WiFi technologies. The course will focus on understanding the quantitative techniques to evaluate the different, protocols, network architecture options and the application performances using the different types of wireless access technologies and mobility protocols.

**Objectives:**

1. To understand network and transport protocols for wireless networks, including mobile IP and variants of TCP.
2. distributed systems platforms for mobile computing, including proxy based architectures and service discovery and interaction platforms.
3. To understand characteristics of local and wide area technologies such as Bluetooth, 802.11 and GSM.
4. To learn the file systems support for mobile computing.

**Unit 1 : Introduction to wireless communication ( 6 Hrs )**

Need and Applications of wireless communication, Wireless Data Technologies, Market for mobile communication, Mobile and wireless devices.

**Unit 2 : Wireless transmission ( 6 Hrs )**

Frequencies for radio transmission, signals, antennas, signal propagation, Multiplexing, Modulation, Spread spectrum and Cellular systems.

**Unit 3 : Medium Access Control ( 7 Hrs )**

Specialized MAC, SDMA, FDMA, TDMA and CDMA

**Unit 4 : Telecommunication Systems ( 6 Hrs )**

GSM, DECT systems – Architecture and protocols, Tetra frame structure, UMTS basic architecture and UTRA modes

**Unit 5 : Wireless LAN ( 7 Hrs )**

Introduction, Infrared v/s Radio transmission, Infrastructure and ad-hoc networks, IEEE 802.11, HIPERLAN, Blue Tooth.

**Unit 6 : Mobile Network Layer and Mobile Transport Layer ( 7 Hrs )**

Mobile Transport Layer : Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast and selective retransmission and recovery, Transaction oriented TCP.

**Outcomes:**

1. Use principles of Mobile Computing and its enabling technologies, and explore exciting ideas, solutions, and paradigm shifts.
2. Understand wireless access and core networks and mobility in cellular and wireless networks using the important standards like GSM, CDMA, GPRS and IEEE 802.11.
3. Evaluate the different, protocols, network architecture options and the application performances.

**Text Books**


1. Jachen Schiller, 'Mobile Communications',Addison-Wesley
2. Peter T. Davis, Craig R. Mc Guffin, 'Wireless LAN' MGH International Edn


**Reference Books:**


1. Sandeep Singhal, Jari Alvinen and group, 'The Wireless Application Protocol',Addison-WesleyCharles Arehart and group, 'Professional WAP',SPD

# CS1134: Geographical Information System

**Aim :**
This course emphasizes on the principles of GIS for characterizing environmental systems and computer-based techniques for processing and analyzing spatial data with map understanding, database design and development, data transformations, spatial analysis, map accuracy assessment, and data visualization. Students will learn to: identify and describe hardware components of a GIS; state differences between databases models; describe and evaluate methods of data capture and sources of data; understand typical GIS operations; identify types of GIS products; identify various applications of GIS.

**Objectives:**

1. To increase student awareness of GIS science and technology and provide opportunities to process, analyze, and visualize spatial data and information using commercially-available GIS software
2. To understand the complexities of data manipulation, analysis, and mapping at different scales of space, time, and complexity
3. To study applications of (GIS) & Remote Sensing
4. To study Data Modeling and Analysis required by these systems

**Unit 1 : Remote Sensing Fundamentals ( 7 Hrs )**

Remote Sensing - Basic Principles: Electromagnetic remote sensing, energy sources, energy interactions with Earth's surface materials.
Microwave Remote Sensing: The radar principle, factors affecting microwave measurements, radar wavebands, SLAR systems, sar, Interpreting sar images, geometrical characteristics.

Remote Sensing Platform and Sensors: Satellite system parameters, sensor parameters, imaging sensor systems, Earth resources satellites, meteorological satellites, OCEANSAT-1, (IRS-4), IKONOS satellite series

**Unit 2 : Image Processing ( 7 Hrs )**

Digital Image Processing: Basic character of digital images, preprocessing, registration, enhancement, spatial filtering, transformations, classification, image classification and GIS.

Visual Image Interpretation: Types of pictorial data products, Image interpretation strategy, image interpretation process, overview of image interpretation equipments

**Unit 3 : GIS and Maps** **( 7 Hrs )**

Fundamentals of GIS : Roots of GIS, overview of information system, the four Ms, GIS definition and terminology, GIS queries, GIS architecture, models of GIS, framework for GIS, GIS categories, levels/scales of measurement.
Map Language: Map as a model, classification of maps, spatial referencing system, map projections, commonly used map projections, grid systems, computer in map production, digital database in a GIS, linkage of GIS to remote sensing

**Unit 4 : Spatial Data Modeling and Management** **( 7 Hrs )**

Spatial Data Modeling: Stages of GIS data modeling, graphic representation of spatial data, raster GIS models, vector GIS models.
GIS Data Management : Data base management systems, GIS data file management, database models, storage of GIS data object based data models, temporal topology, organizational strategy of DBMS in GIS.

**Unit 5 : Data Input, Quality and Analysis** **( 7 Hrs )**

Data Input and Editing: The data stream, data input models, data input methods, GPS for GIS data capture, data editing.
Data Quality Issues: Components of data quality, accuracy, precision and resolution,consistency, completeness, sources of error in GIS, GIS output, modeling errors and error evaluation.

Data Analysis and Modeling: Format conversion, data medium conversion, spatial measurement methods, buffering techniques, overlay analysis, modeling surfaces, modeling networks

**Unit 6 : GIS Applications** **( 7 Hrs )**

Integration of remote sensing and GIS
Urban and municipal applications

**Outcomes:**

1. Upon completion of this course the student will be able to
2. Gain a basic understanding of Remote Sensing concepts, techniques and real world applications.
3. Understand basic GIS data and analysis concepts.
4. Understand the practical applications of GIS.

**Text Books**


1. M. Anji Reddy, "Remote Sensing and Geographical Information Systems", B S Publications, Second Edition, 2006.
2. George B Korte, "The GIS Book", Onword press (Thomson Learning), 5th Edition, 2001.

**Reference Books**


1. Peter A Burrough and McDonell, “Principles of Geographical Information Systems”, Oxford University Press, 1998.
2. Heywood & Raju, “Introduction to Geographical Information Systems” Pearson Education, 2005

# CS1144: Distributed Systems

**Prerequisites:** Operating Systems

**Aim :** To understand the basic theoretical concepts involved in Distributed Systems

**Objectives:**

1. To study the different design issues of Distributed Operating Systems
2. Create an awareness of the major technical challenges in distributed systems design and implementation;
3. Expose students to current technology and distributed infrastructure software;
4. Expose students to past and current research issues in the field of distributed systems;
5. Provide experience in the implementation of typical algorithms used in distributed systems.

**Unit 1: Introduction ( 7 Hrs )**

Introduction to distributed Systems, examples of distributed systems, goals, hardware and software concepts, design issues.
System models: Introduction, Architectural Model, Fundamental Model, Client Server Model.

**Unit 2: Communication ( 7 Hrs )**

Interprocess Communication: Message oriented Communication, Stream Oriented Communication
Layered Protocols: Lower Level, Transport Level and Higher-Level Protocols
Distributed Objects: RPC & LRPC, Remote Method Invocation, Events and Notifications

**Unit 3: Distributed File Systems ( 7 Hrs )**

Distributed File Systems: SUN NFS, CODA, Other DS, Comparisons.
Name Services: Name Entities, Locating Mobile Entities, Removing unreferenced entities, Case studies: DNS Directory, Global Name Service, X 500 DS

**Unit 4: Synchronization ( 7 Hrs )**

Time and Global States: Clock Synchronization, Logical clocks, global state, Event ordering
Distributed Deadlock Algorithms for Avoidance, Prevention, Detection

Co-ordination: Election Algorithms, mutual exclusion, Distributed Transaction

**Unit 5 : Fault Tolerance ( 7 Hrs )**

Process Resilience, Reliable client server communication, Reliable group communication, Distributed Commit and Recovery

**Unit 6: Distributed Shared Memory and Case Studies ( 7 Hrs )**

Architecture of DSM Systems, Design and Implementation Issues of DSM, Structure of Shared memory space, Consistency Models, Replacement strategies, Thrashing, Heterogeneous DSM.
Case Studies: CORBA, Mach, Chorus, Amoeba, Grid, Clusters.

**Outcomes:**

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

1. List the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness.
2. Recognize how the principles are applied in distributed systems and its effect on the software design decisions.
3. Design a distributed system that fulfills requirements with regards to desired properties, be able to recognize when this is not possible, and explain why;

**Text Books:**


1. Andrew S. Tanenbaum & Maarten van Steen, "Distributed Systems Principles and Paradigms", PHI, 1st Edition, 2002.
2. George Coulouris, Jean Dollimore & Tim Kindberg, "Distributed Systems – Concepts and Design", Pearson (LPE), 2nd Edition, 2000.


**Reference Books:**


1. Pradeep K. Sinha, "Distributed Operating Systems Concepts and Design", PHI, 1st Edition, 1997.
2. Andrew S. Tanenbaum, "Distributed Operating Systems", Prentice Hall, 1st Edition, 1994.

# CS1154: Embedded Systems

**Prerequisites :**
Understanding of Microprocessors, Peripheral Chips, Analogue Sensors , Conversion , Interfacing Techniques, Microcontrollers

**Aim :**
This course covers the principles of embedded and real-time systems inherent in many hardware platforms and applications being developed for engineering and science. This course introduces the concepts shared by most embedded systems and their software. It also introduces the techniques used in the development of embedded multitasking application software.

**Objectives:**

1. To understand engineering issues in the development of embedded software, such as the importance of addressing the user's concerns, working with limited resources, maintainability, dependability.
2. To design and implement module structure to solve a problem, and evaluate alternatives and check its correctness
3. To understand fundamentals of Embedded Systems Power Management I/O Interfacing. RTOS and two Real Time OS (RTLinux, VxWorks, Watcom)

**Unit 1: Introduction Embedded Systems** (7Hrs)
Embedded Systems**:** Components, Classification , Characteristics of ES. Review of Microprocessors and Microcontrollers. Embedded Processor Technology, Application Specific System Processor.CISC , RISC, Exemplary Inst Set . IC Technology. ARM Processor.

**Unit 2: System Hardware** (7Hrs)
Power Management of peripherals, Reset Ckt / Watch Dog Timers, Structural Units of Microprocessor , Processor and ,Memory Selection ,Interfacing of Processor Memory, I/O Bus management techniques.

**Unit 3: I/O Interfacing** (7 Hrs)
ADC / DAC, Optical Devices , LCD, High Power Devices( Relays , Stepper Motors), Timing / Counting Devices. Communication using RS232C , I2c , CAN , LIN , USB. Inter Process Communication , Process Synchronisation. Multiple Tasks , Threads, Processes. OS & Embedded System Model.

**Unit 4: Programming Concepts, Embedded Systems Programming C&C+** (7 Hrs)
Software Development Cycle, Assembly Language, High Level Language, C program Elements Micros and Functions, Data Types Data Structures, Modifiers, Ststements, Loops And Pointers, Queues, Stacks, Lists, Order Lists, Program Compilers and Cross compilers, InCircuit Emulators.

**Unit 5: RTOS** (7 Hrs)

Task Scheduling , Semaphores, Signals, Events, Queues, Mail Boxes. Creation of Threads and Inter Thread Communication. Interrupts and ISR , Latency, Deadline, Response Time, Tasks for Resource Management. Device Drivers.

**Unit 6: RTOS Case Studies** (7 Hrs)

Case Studies using any two RTOS (eg. VxWorks , RTLinux) , Cruise Control , Washing Machine , vending Machine , Sending Bytes Stream on TCPIP Network .

**Outcomes:**

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

1. To use high-level system function units for designing embedded computer systems and be able to verify their performance and use it to modify the design
2. To design an embedded system with functional requirements for hardware and software components including processor, networking components, and sensors, along with applications, subsystem interfaces, networking, and middleware
3. To measure embedded system operating characteristics (for example, latency and reliability) and to determine system performance relative to functional requirements.

**Text Books:**


1. Raj Kamal 'Embedded Systems ', Tata McGraw-Hill
2. Dr. K. V. K. K. Prasad "Embedded / real time System : Concepts, Design, &
3. Programming -Black Book" Dreamtech Press Publication


**Reference Books:**


1. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system " Wiley -Dreamtech India Pvt. Ltd.
2. Sriram lyer , Pankaj Gupta,"Embedded Real time Systems Programming", Tata Me Graw Hill. . .
3. Tammy Nergaard "Embedded Systems Architecture - A Comprehensive Guide For Engineering & Programming", Elesevier Publication

# CS1164:Artificial Neural Networks

**Aim:**
This course provides introductions to various aspects of artificial neural networks, with emphasis on elements of design of trainable systems. The course introduces students to the fundamental theory, mathematics and modeling tools necessary to analyze and simulate natural and engineered systems. Theories and real-world application examples will be covered order to explore potential research topics of neural network techniques.

**Objectives:**
1. To focus on the foundations of neural network theory and the application of neural network models in engineering, cognitive science, and artificial intelligence.
2. To introduce the neural networks as means for computational learning;
3. To present the basic network architectures for classification and regression;
4. To give design methodologies for artificial neural networks;
5. To demonstrate neural network applications on real-world tasks

**Unit 1 : Introduction ( 8 Hrs )**

Biological neuron, Models of artificial neural networks, neural processing, neural Network learning rules, Learning & adaptation, Classification Neural learning, Rules-Hebbian, perceptrons, Delta, Widrow Hoff, Winner take all outstar learning Rule, Competitive Learning.

**Unit 2 : Classification ( 8 Hrs )**

Perceptrons, Discrete perceptron as a classifier, Decision and discriminant Functions, Linearly non separable patterns. Perceptrons training for two class and multiclass dichotomizer. Multilayer networks: Delta learning rule for Multiperceptron layer, Generalized Delta learning rule, Feed forward recall and Error back-propagation, Training algorithm.

**Unit 3 : Performance ( 7 Hrs )**

Madeline, Network pruning, Marchands, Neural tree and filing algorithm, Prediction network. Unsupervised learning: Winner take all networks, Hamming networks, Max net, competitive learning K means Clustering and LVQ algorithms, Adaptive resonance theory, ART

**Unit 4 : Associative memories ( 7 Hrs )**

Noneterative procedures for association hop field networks, Discrete Hopfield Networks storage capacity of Hop field networks. Continuous Hopfield networks, Brain state in a box (BOB) networks Boltzmann machines Hetero associations.

**Unit 5 : Optimization techniques ( 7 Hrs )**

Optimization using Hop field networks. Traveling salesperson problem, Iterated gradient descent techniques. Simulated annealing technique, Random search technique genetic algorithm for optimization problems

**Unit 6 : Application of ANN ( 5 Hrs )**

Character recognition, Speech recognition, Signature verification application, Human face recognition.

**Outcomes:**

Upon completion of the course, the students will:

1. Have an understanding of the concepts and techniques of neural networks through the study of the most important neural network models.
2. Have knowledge of sufficient theoretical background to be able to reason about the behaviour of neural networks with application of neural network design principles;
3. be able to evaluate whether neural networks are appropriate to a particular application and undertake analysis of neural network performance;

**Text Books**

1. Zurada, 'Introduction to Artificial Neural Systems', JAICO
2. Mehrotra, Hohan, Ranka , 'Elements of Artificial Neural Networks',PENRAM

**Reference Books**

1. B. Yegnanarayana, 'Introduction to Artificial Neural Networks', PHI

# CS1064: Artificial Intelligence

**Prerequisites:** Data Structure, Design and Analysis of Algorithm

**Aim:** To understand a broader spectrum of Artificial intelligence.

**Objectives:**
1. To understand the concepts of Artificial intelligence
2. To learn and understand the knowledge representation techniques for knowledge base
3 To learn and understand the fundamentals of Neural Network

**Unit 1: Introduction (7 hrs)**
Definition, what is A.I? Foundation of A.I., History, intelligent Agents, Agent Architecture, A.I. A.I. Representation, Properties of internal representation, Futures of A.I
AI Techniques – Importance of AI – Representation of Knowledge, Knowledge Base Systems, State Space Search – Production Systems – Problem Characteristics,

**Unit 2: Heuristics Search Techniques (7 hrs)**
Generate & test – Hill Climbing, Depth First Search, Breadth First Search, Best First Search, A* and AO* Algorithm, Problem reduction – Constraint satisfaction – Means-Ends Analysis.
Game playing – Minimax & Alpha-Beta Cutoffs, waiting for Quiescence, Secondary search.

**Unit 3:Predicate & Logic (7 hrs)**
Using predicate logic: Predicate Calculus, Predicate and arguments, ISA Hierarchy, Frame notation, Resolution, Natural Deduction
Representing simple facts in Logic - Logic Programming ,Computable functions in predicates, resolution – unification, Forward and Backward reasoning , Forward and
Backward chaining rules.

**Unit 4: Structured Knowledge Representation (7 hrs)**
TMS (Truth maintenance system),Statistical and probabilistic reasoning ,Associative Networks, Semantic Nets, Frames Structures, Conceptual Dependencies & Scripts Learning – Concept of Learning – Learning Automata, Genetic Algorithm, Learning by induction
Planning: Block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, least commitment strategy.

**Unit 5: Natural Language Processing (7 hrs)**
Overview of Linguistics, Grammars and Languages, basic Parsing techniques, Semantic analysis and representation structures. Natural Language generation and Natural Language Systems. Syntactic Processing, ATN, RTN..

**Unit 6: Expert Systems** **(7 hrs)**

Architecture – Need and Justification of Expert Systems – knowledge representation, Knowledge acquisition and validation. Utilization and functionality, Perception and Action, real time search, perception, action, vision, robot architecture, Basics of PROLOG,

**Outcome:** The students will play own role as professionals efficiently and effectively in Natural language processing, Perception, Robotics, medicines

**Text Books**


1. Eiaine Rich and Kerin Knight: "Artificial Intelligence." Tata McGraw Hill
2. Stuart Russell & Peter Nerving : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.


**Reference Books**


1. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley, 1990.
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
3. Patterson: “Introduction to AI and Expert Systems”, PHI
4. Nilsson : “Principles of Artificial Intelligence”, Morgan Kaufmann.
5. N.P. Padhy, “ Artificial Intelligence and Intelligent Systems”, OXFORD, 2005

# CS1104:Network Security

**Prerequisites:** Computer Networks

**Aim :** To understand network and information security threats and countermeasures

**Objectives:**

1. Understanding various vulnerabilities posing security threats to computer networks and information systems
2. Understanding solutions proposed to ensure network and information security
3. Understand the pros and cons of cryptographic techniques, their feasibility and security, social implications of network security

**Unit 1 : Introduction ( 8 Hrs )**

Types of attacks: DoS, IP spoofing, replay, DNS poisoning, Worms, viruses, Trojans,Phishing
Need of security, attributes of security, authentication, confidentiality, integrity and cryptography, Vulnerabilities in OSI model, layers
Bioinformatics security
Mathematical foundation for security : Group, Rings , Number theory (Modular arithmetics, Big Integer calculation)

**Unit 2 : Secret Key Cryptography ( 6 Hrs )**

DES and security analysis , security analysis, AES and security analysis, key distribution

**Unit 3 : Public Key Cryptography ( 6 Hrs )**

Principles, RSA, ECC, DSA, key management, Kerberos, diffie-helleman key exchange, attacks and security analysis of each technique, message authentication and hash functions, Hash algorithms, digital signatures, X.509,Certification authorities in public key cryptography
Man-in-the-middle attack

**Unit 4 : Applying security (12 Hrs )**

Network layer security : IPSec for Ipv4 and Ipv6
Transport layer security: SSL and TLS
Application layer security :
Security services, web security considerations, and S/MIME, PGP ,PEM,Https
IPS ,Honey pots, VPN

**Unit 5 : Security policies ( 4 Hrs )**

Types of security policies,role of trust, access control , policy languages
Integerity policies : Biba Integrity Model , Clark-Wilson Integrity model

Hybrid policies : Chinese Wall Model

**Unit 6 : Cyber law** **(4 Hrs )**

Online Business, Jurisdiction, Risk Management, Contracts, e-Contracts, Privacy, Online Privacy. IT Act 2000, Related IT Acts in Other Countries.

**Outcomes:** The student will gain insight into the fundamentals of the broad area of computer network security.

**Text Books**


1. William Stallings,'Cryptography and network security, principles and practices'


**Reference Books**


1. Charlie Kaufman, Radia Perlman and mike speciner,'Network security, private communication in a public world'
2. Christopher M. King, Curtis patton and RSA press, 'Security architecture, design deployment and operations'
3. Stephen northcatt, leny zeltser, et al, 'INSIDE NETWORK Perimeter Security" Pearson Education Asia.
4. Matt Bishop, 'Computer Security :Art and Science', Pearson Education

# CS1074:Web Technology

**Prerequisites :**

1. HTML – Language for designing web Pages
2. Basic Java

**Aim :**

The course delivers an overview of Java-based APIs for developing Web-based applications. This course focuses on techniques for building server-side programs for dynamically generated Web sites, electronic commerce, Web-enabled enterprise computing, and other applications that require WWW access to server-based resources.

**Objectives:**

1. To understand different web technologies like XML, XLS, DOM, Servlet etc.
2. Understanding the Servlet Model
3. Designing and Developing Thread-safe Servlets
4. Designing and Developing Servlets Using Session Management
5. Designing and Developing Servlets to Handle Server-side Exceptions
6. Understanding the Java Server Pages (JSP) Technology Model
7. Designing and Developing JSP pages Using JavaBean Components

**Unit 1 : Java servlets ( 7 Hrs )**

Servlet Architecture, Servlet Interface, Servlet HTTP Interface, Request Processing, Response Generation, Session Management, Servlet Deployment, Servlet Configuration, Servlet Service Management

**Unit 2 : Java server pages ( 7 Hrs )**

JSP Overview, JSP Language Basics, JSP Translation and Compilation Directives, Java Scripting from JSP, Java Abstraction of JSP, Standard Java Objects from JSP, Standard Java Action from JSP, JSP Configuration and Deployment, Custom Java Actions and Tags from JSP.

**Unit 3 : Active Server Platform ( 7 Hrs )**

Fundamentals of ASP, Working with databases & active X data object, Introduction to distributed objects

**Unit 4 : An XML Prime ( 7 Hrs )**

History of XML, Benefits of XML, Components of XML; Parsing XML : Parsing Methodologies, SAX API, The Java API for XML parsing (JAXP), Alternative Parsers.

**Unit 5 : The Document Object Model (DOM)** **( 7 Hrs )**

Defining the Document Object Model , DOM Core Level I, Creating Document Objects, Node Interface, NodeList and NamedNodeMap, Document Interface, Element Interface, Attr Interface, Additional Interfaces, Creating DOM Elements, DOM Level II, The DOM Core Defined, Implementation Anomalies.

**Unit 6 : Extensible Style sheet (XLS)** **( 7 Hrs )**

Overview, XSLT, Referring XLS style sheet

**Outcomes:**

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

1. Identify user and content goals of the proposed web site and create functional and formal design specifications for a site.
2. Understanding the use of web technology to access databases, gather and disseminate information.
3. Exploration of the use of XML, Document Object Model, servlet
4. Student should be able to develop web site using different technologies like ASP, JSP, ActiveX data Object, etc.

**Text Books**


1. Perrone & Chaganti, 'Building Java Enterprise System with J2EE', SAMS Publication
2. Matt J. Crorch, 'Web Programming with ASP & COM', Addison Wesley


**Reference Books**


1. Wendy Lehnert, Web 101, ' Making the Network for you', Addison Wesley
2. Kalakota & Whinston, 'Frontiers & electronic commerce', Addison Wesley
3. Matthews, Jones, 'Unix web server book', Galgotia Publications
4. Daconta & saganich, 'XML Development with Java 2', Sams Techmedia Publications
5. Goldfourb, Prescod, AWL LPE , 'The XML Handook'
6. Larne Pekowsky, 'JavaServer Pages',Addison Wesley
7. Karl Moss, 'Java Servlets', TMGH
8. DOM Level 1 Specifications - www.w3.org/TR/REC-DOM-Level-1/
9. XML DOM Tutorials - http://www.w3schools.com/dom/default.asp

# FOURTH YEAR

# COMPUTER ENGINEERING

# SEMESTER II

# ELECTIVE 1 & 3

# LABORATORY

# CS6114: MULTIMEDIA SYSTEMS

**Objectives:**

1. To implement basic concepts of computer graphics.
2. To implement algorithms to draw various graphics primitives.
3. To implement 2-D and 3-D transformations.
4. To implement applications of multimedia.

## List of Practicals

1. Write a menu Driven program
   a) Create, Edit .Voc file and convert into .Wav file format
   b) Create, Edit .Wav file and convert it to .Voc file format.
2. Create an animation using 3-D Max
3. Create an animation using Flash
4. Write a tool to create presentation slide with audio & video effects
5. Using VRML generate anyone virtual scene
   1. Coffee House
   2. Building Model
   3. Garden Model
6. Implement Edge Detection Algorithm on any Image
7. Implement arithmetic coding technique on any image.
8. Implement JPEG-.DCT compression Technique

**Outcomes:**

Upon completion of the course, students should be able to successfully perform :

1. Implementation of text /image compression algorithm.
2. Design, implementation and evaluation concepts and techniques to the development of a small but realistic multimedia system.

**Text Books**


1. Ranjan Parekh, "Principles of Multimedia",TMH, ISBN 0-07-058833-3
2. Ralf Steinmetz and Klara Nahrstedt "Multimedia Computing, Communication and Applications", Pearson Education.

**Reference Books**


1. Ze-Nian Li, Marks S. Drew, "Fundamentals of Multimedia", Pearson Education.
2. Nigel Chapman and Jenny Chapman. Wiley "Digital Multimedia"
3. A. K. Jain "Fundamentals of Digital Image Processing", PHI
4. Gonzalez, Woods, "Digital Image Processing" Addison Wesley
5. Mark Nelson "Data Compression Book ", BPB.
6. Judith Jeffcoate "Multimedia in Practice":, Pill.
7. Robert Reinhardt, Snow Dowd, “Flash 8 Bible”
8. Keith Peters, “Foundation AS Animation: Making Things Move!”
9. Sanford Kennedy, “3ds max Animation and Visual Effects Techniques”


1.

# CS6124 :Mobile Computing

**Aim :**
This course provides an in depth understanding of wireless access and core networks and mobility in cellular and wireless networks using the important standards of the industry like GSM, CDMA, GPRS and IEEE 802.11 WiFi technologies. The course will focus on understanding the quantitative techniques to evaluate the different, protocols, network architecture options and the application performances using the different types of wireless access technologies and mobility protocols.

**Objectives:**

1. To understand network and transport protocols for wireless networks, including mobile IP and variants of TCP.
2. distributed systems platforms for mobile computing, including proxy based architectures and service discovery and interaction platforms.
3. To understand characteristics of local and wide area technologies such as Bluetooth, 802.11 and GSM.
4. To learn the file systems support for mobile computing.

**1.** Write a simple program for displaying "Hello, World" on your mobile screen using J2ME Wireless Toolkit. Test output in different Emulators provided by J2ME Wireless Toolkit and Transfer it to the mobile.

**2.** Write a program for sending SMS to your friend by using your mobile phone. Use J2ME Wireless Toolkit to develop your application. Test it in the Emulator provided by Toolkit and Transfer it to your mobile.

**3.** Develop a simple calculator for your mobile by using J2ME Wireless Toolkit.

**4.** Write a simple program to take a snapshot by using the Camera in your mobile. Save the snapshot in the image or video format. Use Camera Media API provided J2ME. Test it in Emulator and Transfer it to your mobile.

**Outcomes:**

1. Use principles of Mobile Computing and its enabling technologies, and explore exciting ideas, solutions, and paradigm shifts.
2. Understand wireless access and core networks and mobility in cellular and wireless networks using the important standards like GSM, CDMA, GPRS and IEEE 802.11.
3. Evaluate the different, protocols, network architecture options and the application performances.

**Text Books**


1. Jachen Schiller, 'Mobile Communications' ,Addison-Wesley
2. Peter T. Davis, Craig R. Mc Guffin, 'Wireless LAN' MGH International Edition


**Reference Books:**


1. Sandeep Singhal, Jari Alvinen and group, 'The Wireless Application Protocol', Addison-WesleyCharles Arehart and group, 'Professional WAP',SPD

# CS6134:Geographical Information System

**Aim**
Study various concepts and technology used in a Geographic Information System (GIS) & Remote Sensing

**Objectives**
1. To study applications of (GIS) & Remote Sensing
2. To study Data Modeling and Analysis required by these systems.

Students should complete following assignments in a group of four students (Max).

Each assignment should be completed in 4 hrs. [Contribution of each student of a group should be 4 hrs for completion of each assignment.]

1. Consider any geographical area of your choice and make a survey. Prepare a map for the same (Cartography).
2. Design a database for the entities and geographical area specified in above map.
3. How are databases linked with geographical maps?
4. Discuss case study of any GIS application of your choice.
5. Prepare a presentation on any technology topic on GIS.
6. Study any GIS software and evaluate its main features.

**Outcomes**
Upon completion of this course the student will be able to

1. Gain a basic understanding of Remote Sensing concepts, techniques and real world applications.
2. Understand basic GIS data and analysis concepts.
3. Understand the practical applications of GIS.

**Text Books :**

1. M. Anji Reddy, "Remote Sensing and Geographical Information Systems", B S Publications, Second Edition.
2. Heywood & Raju, "Introduction to Geographical Information Systems" Pearson Education


**Reference Books :**

1. George B Korte, "The GIS Book", Onword press (Thomson Learning), 5th Edition.
2. Peter A Burrough and McDonell, "Principles of Geographical Information Systems", Oxford University Press, 1998.

# CS6144: Distributed Systems

**Prerequisites:** Operating Systems

**Objectives:**

1. To study the different design issues of Distributed Operating Systems
2. To study various methods of communication in Distributed Operating Systems.
3. To analyze Distributed File Systems

## List of Practicals

1. Implement Remote Method Invocation
2. Write a program to implement Remote Procedure Call
3. Simulate Cristian's algorithm for clock synchronization
4. Simulate Berkeley's algorithm for clock synchronization
5. Simulate Lamport's algorithm for clock synchronization
6. Simulate the Ring election algorithm
7. Simulate the Bully election algorithm
8. Simulate the Causal Consistency model
9. Simulate the centralized algorithm for mutual exclusion
10. Simulate the distributed algorithm for mutual exclusion
11. Simulate the token ring algorithm for mutual exclusion
12. Implement the Byzantine algorithm

**Outcomes:**

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

1. List the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness.
2. Recognize how the principles are applied in distributed systems and its effect on the software design decisions.
3. Design a distributed system that fulfills requirements with regards to desired properties, be able to recognize when this is not possible, and explain why;

**Text Books:**


1. Andrew S. Tanenbaum & Maarten van Steen, "Distributed Systems Principles and Paradigms", PHI, 1st Edition, 2002.
2. George Coulouris, Jean Dollimore & Tim Kindberg, "Distributed Systems – Concepts and Design", Pearson (LPE), 2nd Edition, 2000.


**Reference Books:**


1. Pradeep K. Sinha, "Distributed Operating Systems Concepts and Design", PHI, 1st Edition, 1997.
2. Andrew S. Tanenbaum, "Distributed Operating Systems", Prentice Hall, 1st Edition, 1994.

# CS6154: EMBEDDED SYSTEMS

**Objectives:**

1. To understand engineering issues in the development of embedded software, such as the importance of addressing the user's concerns, working with limited resources, maintainability, dependability.
2. To design and implement module structure to solve a problem, and evaluate alternatives and check its correctness
3. To understand fundamentals of Embedded Systems Power Management I/o Interfacing. RTOS and two Real Time OS (RTLinux, VxWorks, Watcom)

## List of Practicals

Note:1) Assignments 1 -10 to be carried out on PIC Microcontroller Kit
2) Assignment 11onwards to be performed on Linux/RTlinux /VXWorks platform
.

1. Study of PIC Microcontroller Development Kit
2. Generate a frequency == <YOUR ROLL NO.: * 1000> Hz. Using all 3 Timers on PIC Kit.
3. Establish Serial Communication between PIC KIT and PC.
4. Read Analog Data , Convert to BCD / ASCII and display it on PC.
5. Read PIC Kit to Read input from a switch count number key presses and display on LED Display.
6. Stepper Motor Controller using PIC KIT. Use switch press to change Direction and Increase / Decrease Speed.
7. Function timing using the timer.
8. Demonstrate Non Pre-emptive scheduling.
9. Demonstrate Pre-emptive scheduling.
10. Create a Function delay using the Timer.
11. Write a linux Shell Script to display no of readable Writable Files in a Specific Directory.
12. Write a C program use 2 Threads One Thread reads from Keyboard . Second Thread Displays the Data. Use Message Queue.
13. Write a C program use 2 Threads One Thread reads from Keyboard . Second Thread Displays the Data. Use Shared Memory..
14. Write a Producer Consumer Program using Semaphore.
15. Testing the Serial Port using the Watcom RTOS.
16. Semaphores.
17. Message Queues.
18. Priority Inversion.
19. Signals.

**Outcomes:**

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

1. To use high-level system function units for designing embedded computer systems and be able to verify their performance and use it to modify the design
2. To design an embedded system with functional requirements for hardware and software components including processor, networking components, and sensors, along with applications, subsystem interfaces, networking, and middleware
3. To measure embedded system operating characteristics (for example, latency and reliability) and to determine system performance relative to functional requirements.

**Text Books:**


1. Raj Kamal ‘Embedded Systems ‘, Tata McGraw-Hill
2. Dr. K. V. K. K. Prasad "Embedded / real time System : Concepts, Design, & Programming -Black Book" Dreamtech Press Publication


**Reference Books:**


1. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system " Wiley -Dreamtech India Pvt. Ltd.
2. Sriram lyer , Pankaj Gupta,"Embedded Real time Systems Programming", Tata Me Graw Hill. . .
3. Tammy Nergaard "Embedded Systems Architecture - A Comprehensive Guide For Engineering & Programming", Elesevier Publication

# CS6164:Artificial Neural Networks

**Objectives:**

1. To focus on the foundations of neural network theory and the application of neural network models in engineering, cognitive science, and artificial intelligence.
2. To introduce the neural networks as means for computational learning;
3. To present the basic network architectures for classification and regression;
4. To give design methodologies for artificial neural networks;
5. To demonstrate neural network applications on real-world tasks

## List of Practicals

1. Supervised Leaning rules
2. Unsupervised Learning rules.
3. Simple Perceptron classifier
4. Feed forward neural network.
5. Delta Rule
6. Back propagation algorithm
7. MAXNET
8. Hamming Distance Classifier
9. Hopfield network
10. SOM
11. ART
12. BAM
13. Applications of ANN

**Outcomes:**

Upon completion of the course, the students will:

1. Have an understanding of the concepts and techniques of neural networks through the study of the most important neural network models.
2. Be able to reason about the behaviour of neural networks with application of neural network design principles;
3. Be able to evaluate whether neural networks are appropriate to a particular application and undertake analysis of neural network performance;

**Text Books**

1. Zurada, 'Introduction to Artificial Neural Systems', JAICO
2. Mehrotra, Hohan, Ranka , 'Elements of Artificial Neural Networks',PENRAM

**Reference Books**

1. B. Yegnanarayana, 'Introduction to Artificial Neural Networks', PHI

# CS6064: Artificial Intelligence

**Objectives:**

1 To understand the concepts of Artificial intelligence
2 To learn and understand the knowledge representation techniques for knowledge base
3 To learn and understand the fundamentals of Neural Network

**Assignments based on:**

1. Implement 8 puzzle problem using A* algorithm.
2. Implement AO* algorithm for tower of Hanoi.
3. Implementation of Unification Algorithm.
4. Implementation of Truth maintenance system using prolog
5. Implementation of Min/MAX search procedure for game Playing
6. Parsing Method Implementation using Prolog.
7. Development of mini expert system using Prolog.

**Outcome:** The students will play own role as professionals efficiently and effectively in Natural language processing, Perception, Robotics, medicines

**Text Books**


3. Eiaine Rich and Kerin Knight: "Artificial Intelligence." Tata McGraw Hill
4. Stuart Russell & Peter Nerving : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.


**Reference Books**


6. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley, 1990.
7. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
8. Patterson: “Introduction to AI and Expert Systems”, PHI
9. Nilsson : “Principles of Artificial Intelligence”, Morgan Kaufmann.
10. N.P. Padhy, “ Artificial Intelligence and Intelligent Systems”, OXFORD, 2005

# CS6104:Network Security

**Objectives:**

1. Understanding various vulnerabilities posing security threats to computer networks and information systems
2. Understanding solutions proposed to ensure network and information security
3. Understand the pros and cons of cryptographic techniques, their feasibility and security, social implications of network security

## List of Practicals

1. Study of DoS attack and IP spoofing
2. Implementation of an antivirus tool
3. Implementation of DES
4. Implementation of Rijndael algorithm using 64-bit key (AES)
5. Implementation of RSA
6. ECC implementation
7. Implementation of Diffie-hellman key exchange algorithm.
8. Implement MD5/SHA algorithm
9. Implementation of email security using PGP
10. https
11. Implement packet filter
12. Kerberos simulation

**Outcomes:** The student will gain insight into the fundamentals of the broad area of computer network security.

**Text Books**

1. William Stallings,'Cryptography and network security, principles and practices' PHI

**Reference Books**

1. Charlie Kaufman, Radia Perlman and mike speciner,'Network security, private communication in a public world'
2. Christopher M. King, Curtis patton and RSA press, 'Security architecture, design deployment and operations'
3. Stephen northcatt, leny zeltser, et al, 'INSIDE NETWORK Perimeter Security" Pearson Education Asia.
4. Matt Bishop, 'Computer Security :Art and Science', Pearson Education

# CS6074:Web Technology Lab

**Objectives:**

1. To understand different web technologies like XML, XLS, DOM, Servlet etc.
2. Understanding the Servlet Model
3. Designing and Developing Thread-safe Servlets
4. Designing and Developing Servlets Using Session Management
5. Designing and Developing Servlets to Handle Server-side Exceptions
6. Understanding the Java Server Pages (JSP) Technology Model
7. Designing and Developing JSP pages Using JavaBean Components

## List of Practicals

1. Servlet program to add information in the database from an HTML form (fields :- Name,Acc.No. , Amount, City, Password)
2. Servlet program to update database on some transactions.
3. Servlet program for user validation Login , Acc. No., Password, etc.
4. Servlet to send back a gif image i.e. image is picked from server hard disk or a file and written as a binary stream to the client. Give the relevant values
5. Servlet which will upload a client side text & save it on server hard-disk.
6. ASP to assign a new session id for a fresh user. The same page on being requested by a user (who already has a session) should abandon the session and display relevant message.
7. ASP to display a form. The target (action) of the form should be the same ASP which should display the form values relevant in the form in the tabular format.
8. ASP to accept the registration details from a form and add it to the database.
9. Design of a web site using above technology.
10. Java Program for parsing XML file (convert XML objects to Java objects)
11. XML/XSL script to show particular HTML file.
12. Implement DOM interface to add, update, and delete 'Nodes'
13. Convert Database to XML file

**Outcomes:**

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

1. Identify user and content goals of the proposed web site and create functional and formal design specifications for a site.
2. Understanding the use of web technology to access databases, gather and disseminate information.
3. Exploration of the use of XML, Document Object Model, servlet
4. Student should be able to develop web site using different technologies like ASP, JSP, ActiveX data Object, etc.

**Text Books**


1. Perrone & Chaganti, 'Building Java Enterprise System with J2EE', SAMS Publication
2. Matt J. Crorch, 'Web Programming with ASP & COM', Addison Wesley


**Reference Books**


1. Wendy Lehnert, Web 101, ' Making the Network for you', Addison Wesley
2. Kalakota & Whinston, 'Frontiers & electronic commerce', Addison Wesley
3. Matthews, Jones, 'Unix web server book', Galgotia Publications
4. Daconta & saganich, 'XML Development with Java 2', Sams Techmedia Publications
5. Goldfourb, Prescod, AWL LPE , 'The XML Handook'
6. Larne Pekowsky, 'JavaServer Pages',Addison Wesley
7. Karl Moss, 'Java Servlets', TMGH

## CS6044: Project Work

**Teaching Scheme** **Examination Scheme**
**Practical : 02 Hrs**

**Aim**
This course addresses the issues associated with the successful management of a software development project. The course emphasizes five project processes: initiating, planning, executing, controlling and closing. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and their career.

**Objectives**
1. To provide an understanding of the project planning process and problem scoping
2. To define roles and work with cross functional teams
3. To establish clear project objectives and milestones
4. To create effective and deliverable project plans
5. To perform rigorous analysis and design activities for solution planning
6. To select appropriate technology for implementation and testing

**Overview of the Course:**
1. The Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report should consist of following documents:

| Sr. | Project Item |
|---|---|
| 1 | Project Cover Front Page |
| 2 | Project Completion Certificate [Institute] |
| 3 | Project Completion Letter [In case of Sponsored Projects] |
| 4 | Acknowledgments |
| 5 | Table of Contents |
| 6 | List of Figures |
| 7 | List of Tables |
| 8 | Project Synopsis [Problem Background, Existing System Details, Proposed Solution] |
| 9 | Feasibility Study Report |
| 10 | Project Plan |
| 11 | System Requirement Specification |
| 12 | System Analysis Document: UML Use Case Diagrams |
| 13 | System Analysis Document: UML Sequence Diagrams |
| 14 | System Analysis Document: UML State Diagrams |
| 15 | System Design Document with Module Specifications |
| 16 | System Implementation |
| 17 | System Testing and Experimental Findings |
| 18 | Conclusion |
| 19 | References |

2. The Project Work will be assessed jointly by a panel of examiners having more than Five Years experience. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.
3. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

**Note:**

The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

**Outcomes:**

Upon completion of the course, the students will be to:

1. Identify major issues in complex situations; and know how to prepare alternative solutions and make decisions.
2. Becoming aware of the need to think and act in an entrepreneurial manner by developing the ability to: critically and creatively understand innovations and development, work independently and collaboratively.

## Seminar (Project Based)

**Teaching Scheme** **Examination Scheme**

**Aim:**
This course aims to prepare students in communication practices used in professional environments. The course focuses on developing technical writing and speaking which will enable the students to bear adequate skills required in global workplace.

**Objectives:**
1. To introduce technical communication as a practice
2. To strengthen your understanding of, and ability to apply, communications strategies.
3. To learn importance of various types of media and its effective usage
4. To learn effective document preparation activities with Completeness, Conciseness, Concreteness, Clarity and Correctness characteristics
5. To design and use tables, graphs, and technical illustrations.

**Overview of the Course:**
The student will explore the Project Work in Computer Engineering or Information Technology with an interest in learning about the document design process. The student will prepare a Project Report based on Project Work and present the work to panel of examiners. The panel of examiners will evaluate the student's performance on the basis of technical contents, organization of the work, communication skills and ability to address audience queries. The presentation need to contain following information:

1. The Problem Statement
2. The Proposed Solution
3. Need of the System constructed
4. System Analysis (Use case/Sequence/State chart/Activity Diagrams)
5. System Design (Class Diagram/ System Subsystem Diagram)
6. Algorithms used (If Any)
7. Implementation (Technology/ Programming Languages/ Environment)
8. Testing (Unit/ Integration/ System)
9. Merits/Demerits of System Constructed
10. Future Enhancements
11. Conclusion
12. References

**Outcomes:**
Upon completion of this course, the student should be able to

1. Trace the communication situation based on needs and audiences.
2. Develop document information logically and efficiently
3. Select the appropriate media and format for presenting information.
4. Evaluate their documents to be sure that the documents fulfill their purpose and to ensure that they can be revised if necessary.
5. Write specific kinds of documents in technical and scientific communities.