

**THIRD YEAR**

**COMPUTER ENGINEERING**

**Pattern - B**

**FIRST SEMESTER**

**(THEORY )**

# CS0513 : Operating Systems

## **Prerequisites (if any):**

Data Structures and Algorithms.  
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 & OS services, Batch, multiprogramming, multitasking, time sharing, parallel, distributed & real-time OS.  
Examples of OS: Linux and variants, MS-Windows 2000, Symbian OS.

## **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: S/W approaches, H/W Support, Semaphores, Message Passing, Monitors, Classical Problems of Synchronization: Readers-Writers problem, Producer Consumer Problem, Dining Philosopher problem, Sleeping Barber problem, Cigarette Smokers problem.

## **Unit 3 : Deadlock and Shell programming**

**( 7 Hrs )**

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

#### **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, Working Set Model.

#### **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, Security Issues, Secondary Storage Management.

#### **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.

**Note:** Every aspect of OS should be taught in comparison W.R.T. WINDOWS 2000 & UNIX.

**Text Books:**

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

**Reference Books:**

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

# CS0523 : Database Management System

**Prerequisites :** Data structures

**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 )**

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**

**( 10 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.

Relational Data Model: 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, relational calculus

**Unit 3 : RDBMS Design**

**( 11 Hrs )**

Normalization: need of normalization, Functional dependency, Inference Rules for Functional Dependencies ,Closure of functional dependencies, algorithm to find candidate key, Algorithm to find closure of given attribute set, algorithm to find minimal cover of functional dependencies, Normal forms 1NF, 2NF, 3NF, 4NF, 5NF and DKNF. Checking of lossless join decomposition,

Automation of RDBMS Design and properties of automation tool. A real world example of normalization.

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

DDL, DML, DCL, simple and Nested queries, sub queries, PL/SQL procedures, functions, triggers, security and Authorization in SQL, embedded SQL

**Unit 5 : Transaction management ( 6 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 ( 4 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

**Text 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.

**Reference Books**

1. Thomas Connolly and Carolyn Begg, 'Database Systems', Pearson Education Low Price Edition, Third Edition, 2003.
2. Ramakrishnan and Gehrke, 'Database Management Systems', McGraw-Hill International Edition, Third Edition, 2003.
3. Rob, Coronel, "Database system, design, implementation and management", Thomson learning, fourth edition, 2001.

# CS0533 : THEORY OF COMPUTATION

## Prerequisites :

Data Structures and Algorithms

## Aim :

This course introduces students to fundamental area of computer science which enables students to focus on the study of abstract models of computation. The course exposes students to the computability theory, as well as to the complexity theory. By the end of this course, students will develop the rigor and skills required to precisely present and prove various properties of computations.

## Objectives:

1. Study abstract computing models (FA,PDA, PM, TM), their languages, grammar, applications, limitations and relevance to modern day computing.
2. Learn about the theory of computability and complexity.

## Unit 1 : Automata Theory

(7 Hrs )

Introduction to Finite Automata, Structural Representations, Automata and Complexity, Central Concepts to Automata Theory: Alphabets, Strings, Languages and Problems, Finite Automata: An Informal Picture of FA, Deterministic Finite Automaton (DFA): How a DFA processes Strings, Simpler Notations for DFA, Extending the transition function to strings, the language of DFA, Non-deterministic Finite Automaton (NFA): NFA, Extended transition function, the language of an NFA, Equivalence of NFA and DFA, FA with  $\epsilon$ -transitions: Use of  $\epsilon$ -transitions, NFA with  $\epsilon$ ,  $\epsilon$ -closures, Extended transitions and languages for  $\epsilon$ -NFA, Eliminating  $\epsilon$ -transitions-Conversion of NFA with  $\epsilon$  to NFA without  $\epsilon$ , Conversion of NFA without  $\epsilon$  to DFA, Conversion of NFA with  $\epsilon$  to DFA (direct method), FA with output: Moore and Mealy machines -Definition, models, inter-conversion.

## Unit 2 : Regular Expressions (RE) and Languages

( 7 Hrs )

Regular Expressions - Operators of RE, Building RE, Precedence of operators, Algebraic laws for RE, Arden's Theorem, FA and RE: DFA to RE, RE to DFA (RE to  $\epsilon$ -NFA &  $\epsilon$ -NFA to DFA and RE to DFA-direct method), FA limitations, Properties of Regular Languages: pumping lemma for regular languages, closure and decision properties of regular languages, Equivalence and minimization of automata, Application of RE: Regular expressions in Unix, GREP utilities of Unix, Lexical analysis and finding patterns in text.

## Unit 3: Context Free Grammars (CFG) and Languages

(7 Hrs )

Context Free Grammar- Definition, derivations, languages of a grammar, sentential form, Parse Tree- inference, derivation and parse tree, from inference to tree, Ambiguity in grammars and languages: removal of ambiguity, inherent ambiguity, Properties of CFL- Normal forms- Chomsky Normal Form and Greibach Normal Form, Eliminating unit productions, useless production, useless symbols, and  $\epsilon$ -productions, Regular Grammar - definition, left linear and right linear Regular Grammar, Regular Grammar and Finite Automata, FA to RG and RG to FA, Inter-conversion between left linear and right linear regular grammar.

#### **Unit 4: Push Down Automata (PDA)**

**(7 Hrs )**

Definition, The Language of PDA, Equivalence of PDA's and CFG- CFG to PDA, PDA to CFG, Deterministic Push Down Automata (DPDA)- Regular language and DPDA, DPDA and CFL, DPDA and ambiguous grammar, Non-deterministic Push Down Automata (NPDA), The pumping lemma for CFL, Closure properties of CFL, Decision properties of CFL, Chomsky Hierarchy, Application of CFG: Parser, Markup languages, XML and Document Type Definitions.

#### **Unit 5: Turing Machine and Post Machine**

**(7 Hrs )**

Problems that computers cannot solve, The Turing Machine(TM)- Notation, the language of TM, TM and Halting, Programming techniques to TM, Extensions to basic TM, TM and Computers. Introduction to Post Machines, Comparison between FA, PDA, Post Machine and TM

#### **Unit 6: Introduction to Computational Complexity**

**(7 Hrs)**

Un-decidability: A Language that is not recursively enumerable, An un-decidable problem that is RE, Post Correspondence Problem, Intractable Problems. Application and Comparison of FA, RE, PDA, PM, TM and their limitations.

#### **Outcomes:**

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

1. Understand and use language theory and computability. Language theory includes: regular expressions, regular languages, finite automata (deterministic and non-deterministic), context-free languages, pushdown automata, and language grammars. Computability includes: Turing machines and their computing power, unsolvable problems, and intractable problems (NP-Completeness)
2. Student should have a thorough understanding of Formal Grammar, Languages and the machines that accept them.
3. Should understand the limitations and applications of all the machines described in the framework of this subject.

#### **Text Books**

1. Cohen D., 'Introduction to Computer Theory', Wiley Publications, 2<sup>nd</sup> Edition

2. Mishra K., Chandrasekaran N., 'Theory of Computer Science (Automata, Languages and Computation)', Prentice Hall of India, Second Edition

**Reference Books**

1. Martin J., 'Introduction to Language and Theory of Computation', Tata McGraw-Hill  
Third edition
2. Hopcroft J., Motwani R., Ullman J., 'Introduction to Automata Theory, Languages and Computations', Pearson Education Asia, Second edition
3. Lewis H., Papadimitriou C., 'Elements of Theory of Computation', Pearson  
Education Asia ,Second edition

# EC4873: Digital Signal Processing

## **Prerequisites:**

Knowledge of basic Engineering Mathematics

## **Aim:**

To understand methodology to analyze signals and systems.

To get acquainted with the DSP Processors.

## **Objectives:**

To understand how to analyze and manipulate digital signals in various domains and design digital filters.

## **Unit 1 :Introduction to signals and systems**

**( 7 Hrs )**

Classification of Signals: Analog, Discrete-time and Digital, Basic sequences and sequence operations. Discrete-time (D.T) systems, Properties of D. T. Systems and Classification Linear Time Invariant Systems, impulse response, linear convolution and its properties, properties of LTI systems, parallel and cascade connection, Correlation of DTS, DTS described by difference equations, Recursive and Non recursive, Relaxed and Non relaxed systems, Periodic Sampling, Sampling Theorem, Frequency Domain representation of sampling, reconstruction of a band limited Signal, A to D conversion Process: Sampling, quantization and encoding.

## **Unit 2: Fourier and Z transforms**

**( 7 Hrs )**

Representation of Sequences by Fourier Transform, Symmetry properties of F. T., F. T. theorems: Linearity, time shifting, frequency shifting, time reversal, differentiation, convolution theorem, and windowing theorem. Z-transform, ROC and its properties, Inverse z transform by inspection, partial fraction and complex inversion, Z transform properties: multiplication by exponential sequence, Linearity, time shifting, frequency shifting, time reversal, convolution theorem initial value theorem, final value theorem Unilateral Z-transform: solution of difference equation with and without initial condition.

## **Unit 3: Frequency analysis of Signals and Systems**

**( 7 Hrs )**

Frequency Response of LTI Systems: Ideal frequency selective filters, magnitude and phase response, group delay, System Functions for LTI Systems: Stability and causality, inverse systems, significance of poles/zeros, Frequency Response for Rational System Functions: Frequency Response of a single zero or pole, Frequency response from pole-zero plot using

simple geometric construction, systems with Linear phase, Generalized Linear phase systems, Four Types of GLPS

#### **Unit 4 : DFT and its Properties**

**( 7 Hrs )**

Sampling the F.T., Fourier representation of finite-duration sequences, The Discrete Fourier Transform, Properties of DFT: Linearity, circular shift, duality, symmetry, Circular Convolution, Linear Convolution using DFT, Effective computation of DFT and FFT, DIT FFT, DIF FFT, Overlap and save algorithm, Goertzel Algorithm Inverse DFT using FFT, Practical considerations in FFT implementation

#### **Unit 5 : Design of Digital Filters**

**( 7 Hrs )**

Concept of filtering, Ideal filters and approximations, specifications, IIR filter design from continuous time filters: Characteristics of Butterworth, Chebyshev approximations, impulse invariant and bilinear transformation techniques, Design examples, FIR filter design using windows: properties of commonly used windows, incorporation of Generalized Linear Phase, Design Examples, Design using Kaiser window, Comparison of IIR and FIR Filters

#### **Unit 6 : Realization of Filters**

**( 7 Hrs )**

Block diagrams and Signal flow graph representation of LCCDE, Basic structures for IIR Systems: direct form, cascade form, parallel form, feedback in IIR systems, Basic Structures for FIR Systems: direct form, cascade form, structures for linear phase FIR Systems DSP Processors case study.

#### **Outcomes: At the conclusion of the course the students would be able to :**

1. Study classification of signals and system, verify sampling theorem, convolution and study its application.
2. Understand the frequency and Z domain properties and analysis of signals.
3. Study computation of DFT and FFT.
4. Design and implement digital filters .

#### **Text Books**

1. Proakis J., Manolakis D., 'Digital signal processing', Prentice Hall, 3<sup>rd</sup> Edition
2. Oppenheim A., Schafer R., Buck J., 'Discrete time signal processing', Prentice Hall, 2<sup>nd</sup> Edition, 2003

#### **Reference Books**

1. Babu R., 'Digital Signal Processing', Scitech Publications, 2<sup>nd</sup> Edition
2. Mitra S., 'Digital Signal Processing: A Computer Based Approach', Tata McGraw-Hill, 1998
3. Vallavraj A., 'Digital Signal Processing',

4. White S., 'Digital Signal Processing', Thomson Learning
5. E. C. Ifleachor and B. W. Jervis, "Digital Signal Processing- A Practical Approach", 2nd Edition, Pearson education.

## **EC4813 : Microprocessors and Microcontrollers**

### **Prerequisites :**

Understanding of Microprocessors, Peripheral Chips, Analogue Sensors, Conversion, Interfacing Techniques.

### **Aim :**

This course covers the design of hardware and software codesign using a modern microcontroller. It emphasizes on assembly language programming of the microcontroller including device drivers, exception and interrupt handling, and interfacing with higher-level languages.

### **Objectives:**

1. To exhibit knowledge of the architecture of microcontrollers and apply program control structures to microcontrollers;
2. To develop the ability to use assembly language to program a microcontroller and demonstrate the capability to program the microcontroller to communicate with external circuitry using parallel ports;
3. To demonstrate the capability to program the microcontroller to communicate with external circuitry using serial ports and timer ports.

### **Unit 1 : Introduction to Pentium microprocessor**

**( 7 Hrs )**

Pentium Microprocessor: History ,Feature & Architecture, Pin Description , Functional Description Real Mode, Risc Super Scalar, Pipe lining , Instruction Pairing, Branch Prediction, Inst Data Cache. FPU

### **Unit 2 : Bus Cycles and Memory Organization:**

**( 7 Hrs )**

Bus Cycles & Memory Organisation : Init & Configuration, Bus Operations-RST, Bus Operations-RST, Mem/Io Organisation, Data Transfer Mechanism , 8/16/32 bit Data Bus I, Programmers Model, Register Set, Instru Set , Data Types, Instructions

### **Unit 3 : Protected Mode:**

**( 6 Hrs )**

Protected Mode :Intro Segmentation, Supp Registers ,Rel Int Desc, Mem Man thru Segmentation , Logical to linear translation, protection by segmentation, Privilege Level protection, related instructions, inter - privilege level transfer of control, paging-support registers, descriptors ,linear-physical add trans, TLB, page level protection ,virtual memory.

#### **Unit 4 : Multitasking, Interrupts, Exceptions and I/O**

**( 6 Hrs )**

Multitasking, Interrupts, Exception I/O :Multi Tasking Support Reg , Rel Des, Task Switch I/O per BitMap, Virtual Mode, Add Gen, Priv Level, Inst &Reg ,enter/Leaving V86 M, Interrupt Structure Real/Prot V86 Mode, I/O Handling, comparison of 3 modes.

#### **Unit 5 : AVR Micro controller**

**( 7 Hrs )**

Atmega Family Architecture , ,Data / Programme Memory , Reg set Reg Bank SFR, Ext Data / Mem Programme Mem, Interrupt Structure , Timer Prog ,Serial Port Prog , Misc Features, Min System

#### **Unit 6 : PIC Micro-Controller**

**( 7 Hrs )**

PIC Micro-Controller :OverView ,Features, Pin Out, Capture /Compare /Pulse width modulation Mode , Block Dia Prog Model, Rest /Clocking, Mem Org, Prog/Data, Flash Eprom, Add Mode/Inst Set Prog , I/o, Interrupt , Timer, ADC

#### **Outcomes:**

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

1. Describe and use the functional blocks utilized in a basic microcontroller based system.
2. Describe the programmer's model of the CPU's instruction set and various addressing modes.
3. Proficiently use the various instruction set and functional groups, when programming.
4. Integrate structured programming techniques and sub-routines into microcontroller based hardware topologies.
5. Develop I/O port, ADC hardware, and software interfacing techniques.
6. Describe the use of sensors, interfacing, and signal conditioning when utilizing the microcontroller in control and monitor applications.

#### **Text Books:**

1. Antonakos J., "The Pentium Microprocessor", Pearson Education, 2004, 2<sup>nd</sup> Edition.
2. Deshmukh A., "Microcontrollers - Theory and Applications", Tata McGraw-Hill, 2004,

#### **Reference Books:**

1. Mazidi M., Gillispie J., " The 8051 Microcontroller and embedded systems", Pearson education, 2002, ISBN - 81-7808-574-7
- 2 Intel Pentium Data Sheets
3. Ayala K., "The 8051 Microcontroller", Penram International, 1996, ISBN 81 -900828-4-1
4. Intel 8 bit Microcontroller manual
5. Microchip manual for PIC 16CXX and 16FXX

**Third Year**  
**Computer Engineering**

**First Semester**  
**(Practicals )**

# CS5513: Operating Systems

## Prerequisites (if any):

Data Structures and Algorithms  
Computer Organization

## Objectives:

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

## List of Practicals

1. Execution of basic Unix commands
2. Execution of advanced Unix commands
3. Implement a shell program to find out if a given string is a palindrome or not.
4. Generate a student report using Awk programming
5. Solve the Readers-Writers problem using threads and semaphores.
6. Solve the Readers-Writers problem using threads and mutex.  
(Use 2 Readers and 2 Writers)
7. Solve the Producers-Consumers problem using threads and mutex
8. Solve the Producers-Consumers problem using threads and semaphores  
(Use 2 producers and 2 consumers)
9. Implement the Dining Philosopher's problem using Multithreading
10. Simulate the following CPU scheduling algorithms:
  - a. Shortest Job First (Non-preemptive)
  - b. Round Robin(Draw the Gantt charts and display the finish time, turnaround time, waiting time for each process)
11. Simulate the LRU page replacement algorithm
12. Write a program to simulate the following disk scheduling algorithms:

- a. SSTF
- b. SCAN

**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,4<sup>th</sup> Edition ,2001 .
- 2. Silberschatz A., Galvin P., Gagne G., “Operating System Concepts”, John Wiley and Sons, 7<sup>th</sup> Edition,2003.

**Reference Books:**

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

## CS5523: Database Management System

### Prerequisites :

Data structures

### Objectives:

1. Deep understanding of database design.
2. Implementation of database systems
3. Conversant with oracle database: SQLPLUS, PL/SQL, Embedded SQL
4. Introduction of current trends.

### List of Practicals (Any 12-14 of the following)

1. Describe the database application you propose to work with throughout the course. Your description should be brief and relatively informal.
2. Draw an entity-relationship diagram for your proposed database.
3. Translate an E/R diagram to set of relations.
4. Normalize these relations to BCNF. Check normalized relations for lossless join decomposition.
5. Write an SQL database schema for your application, using the CREATE TABLE command. Tables must be created by using on delete cascade, on update cascade, primary key, foreign key and any other integrity constraints like not null. Add CHECK constraints to relations of your database schema. Show the revised schema. Add a new column and constraint in any existing table which already has few tuples in that. Insert few tuples in newly created tables.
6. Write DML statements for your tables.
7. Write few queries on your database, using the select-from-where construct of SQL
8. Make use of various operators as between...and, in, not null, like etc. Also make use of order by clause.
9. Write queries involving single row functions.
10. Write queries using group functions. Make use of group by and having clause.
11. Write queries involving multiple tables using equijoin, non equijoin, and self join.

- Write queries involving subqueries.
12. Write queries involving set operators. Create two views on top of your database schema involving two or more tables.
  13. Write stored functions and/or procedures in PL/SQL.
  14. Write SQL Triggers.
  15. Write a C program that takes input as set of Functional Dependency F and computes F closure.
  16. Write a C program that takes input as set of Functional Dependency F and attributes of a relation and find out key of the given relation.
  17. Write an embedded SQL program to add, modify and delete records from your table.
  18. Write an embedded SQL program to execute given queries.
  19. Write a C program that takes input as set of Functional Dependency F and a given attribute set and computes closure of given attribute set
  20. Write a C program that takes input as set of Functional Dependency F and computes minimal cover of F.
  21. Controlling user access to database (Oracle/MySql/PostGraySQL/Ms SQL Server).

### **Outcomes:**

Students will be able to design and implement database system for a small scale real world application.

### **Text 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.

### **Reference Books**

1. Thomas Connolly and Carolyn Begg,'Database Systems', Pearson Education Low Price Edition, Third Edition, 2003.
2. Ramakrishnan and Gehrke, 'Database Management Systems", McGraw-Hill

International Edition, Third Edition, 2003.

3. Rob, Coronel, "Database system, design, implementation and management", Thomson learning, fourth edition, 2001.

## **EC9883: Digital Signal Processing**

### **Prerequisites :**

Knowledge of basic Engineering Mathematics, Computer graphics, 'C programming'

### **Objectives:**

To understand how to analyze and manipulate digital signals, implement digital filters and have the fundamental programming knowledge to do so

### **List of Practicals**

1. Verification of sampling Theorem:
2. Linear Convolution
3. GDE
4. Correlation.
5. Pole Zero Implementation
6. Magnitude and Phase response of a system
7. DFT
8. Circular convolution
9. FFT Algorithms (DIT/DIF)
10. Design IIR filter using analog filter approximations
11. Designing FIR filters using windowing techniques

### **Outcomes: The students would be able to :**

1. Study and verify sampling theorem.
2. Implement Convolution and study its application.
3. Plot a Magnitude and phase response of a system from the location of poles and zeros
4. Study circular convolution and its implementation to compute DFT.
5. Implementing FFT to compute DFT
6. Design, analyze, and implement digital filters in Matlab.
7. Implement filters on a digital signal processor

### **Note:**

All assignments to be implemented using either C or Matlab any 2 of the above using DSP Processor

### **Text Books**

1. Oppenheim A., Schafer R., Buck J., 'Discrete time signal processing', Prentice Hall, 2<sup>nd</sup> Edition, 2003
2. Proakis J., Manolakis D., 'Digital signal processing', Prentice Hall, 3<sup>rd</sup> Edition

### **Reference Books**

1. Babu R., 'Digital Signal Processing', Scitech Publications, 2<sup>nd</sup> Edition
2. Mitra S., 'Digital Signal Processing: A Computer Based Approach', Tata McGraw-Hill, 1998
3. Vallavraj A., 'Digital Signal Processing',
4. Manual - ADSP 21 XX family DSP
5. White S., 'Digital Signal Processing', Thomson Learning
6. Fundamentals of signals and systems using Matlab by Edward Kamen and Bonnie Heck, Prentice Hall.
7. Getting started with Matlab, Rudrapratap

## **EC9893 : Microprocessors and Microcontrollers**

### **Prerequisites :**

Understanding of Microprocessors, Peripheral Chips, Analogue Sensors , Conversion , Interfacing Techniques.

### **Aim :**

This course covers the design of hardware and software codesign using a modern microcontroller. It emphasizes on assembly language programming of the microcontroller including device drivers, exception and interrupt handling, and interfacing with higher-level languages.

### **Objectives:**

1. To exhibit knowledge of the architecture of microcontrollers and apply program control structures to microcontrollers;
2. To develop the ability to use assembly language to program a microcontroller and demonstrate the capability to program the microcontroller to communicate with external circuitry using parallel ports;
3. To demonstrate the capability to program the microcontroller to communicate with external circuitry using serial ports and timer ports.

### **List of Practicals**

1. Write an ALP using 8086 instructions to simulate following DOS commands.
  - Type command.
  - Copy command.
  - Delete and Rename file.Make use of the PSP to handle command line argument.
2. Write an ALP for floating point calculations using NDP (Numeric Data Processor) 80x87. Calculate the difference between the areas of the two circles. User should be able to enter the radius in Real format.
3. Write a 'C' program with inline assembly language to read the content of BOOT SECTOR of the Floppy Disk and display the following information:
  - Bytes per sector.
  - Number of sector per cluster
  - Number of sectors per FAT
  - Number of FAT

- Number of root directory entries.
4. Write a 'C' program with inline assembly language to read root directory of the A drive and display the contents of the root directory as follows:
 

Filename	Extension	File Size	Starting Cluster
----------	-----------	-----------	------------------
  5. Write a 'C' program with inline assembly language to display a file content using File Allocation Table (FAT). The file to be displayed should present on the floppy disk. Also display the FAT chain.
  6. Write a program in assembly language for cache memory simulation using following information:
 

Consider 16 KB of main memory, 128 bytes of 2-way set associative cache. Assume 4 Bytes/Line. The user gives a series of physical address. Find cache Hit or cache MISS in a line. If cache MISS there in a line that block should be replaced in cache by LRU technique.
  7. Write a program in assembly language for installable device driver for printer. Check your program by modifying CONFIG.SYS file from drive A.
  8. Write a 'C' program for PC to PC communication using NULL modem connection. Also observe the signals on PC explorer Kit
    - Online data transfer.
    - File transfer.
  9. Write an ALP to glow the pixels on the screen according to the following conditions. All pixels on which mouse arrows scrolls due to dragging of left button.
    - a. One pixel on which left button clicks.
    - b. Pixels on straight line, joining last two points on left click if right button clicks. (Use Bresenham's algorithm).

Also show X and Y co-ordinates at the bottom left corner of the screen continuously as the mouse drags.
  - 10 Write a program to switch the processor from real mode to protected mode & display message on screen. Demonstrate the proof of being in protected mode and perform task switching.
  11. Write an ALP to perform detection of DPMI (DOS protected mode interface) and the following:
    - Make CPU identification.
    - Capture MSW.
    - Display contents of GDTR, IDTR, LDTR and Task register.

12. Interface microcontroller 8051 with successive approximation byte ADC 0809 and Write a program to read output for input varying from 0 to 5 Volt.
  
13. Interface 8051 with DAC 0808 to obtain following waveform
  - Square.
  - Triangular.
  - Stair case.
  - Temple
  
14. Write a program using microcontroller 8051 to interface with stepper motor and using user inputs do the following.
  - Rotate it in clockwise or anticlockwise direction.
  - Vary the speed.
  -

**Outcomes:**

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

1. Describe and use the functional blocks utilized in a basic microcontroller based system.
2. Describe the programmer's model of the CPU's instruction set and various addressing modes.
3. Proficiently use the various instruction set and functional groups, when programming.
4. Integrate structured programming techniques and sub-routines into microcontroller based hardware topologies.
5. Develop I/O port, ADC hardware, and software interfacing techniques.
6. Describe the use of sensors, interfacing, and signal conditioning when utilizing the microcontroller in control and monitor applications.

**Text Books:**

1. Antonakos J., "The Pentium Microprocessor", Pearson Education, 2004, ISBN - 81-7808-545-3
2. Deshmukh A., "Microcontrollers - Theory and Applications", Tata McGraw-Hill, 2004, ISBN 0-07-058595-4

**Reference Books:**

1. Mazidi M., Gillispie J., " The 8051 Microcontroller and embedded systems", Pearson education, 2002, ISBN - 81-7808-574-7
- 2 Intel Pentium Data Sheets
3. Ayala K., "The 8051 Microcontroller", Penram International, 1996, ISBN 81 -900828-4-1
4. Intel 8 bit Microcontroller manual
5. Microchip manual for PIC 16CXX and 16FXX

**Third Year**  
**Computer Engineering**

**Second Semester**  
**(Theory )**

# CS0543 : 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.

## 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 and Experience in Designing and Rapid Prototyping.
6. Presenting the various techniques of software cost estimation and risk assessment.
7. To learn how to work in teams.

## 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, Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods

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.

**Text Books**

1. Ian Sommerville, 'Software Engineering', Addison-Wesley, 7<sup>th</sup> Edition ,2004.
2. Roger S Pressman,'Software Engineering: A Practitioner's Approach', McGraw Hill, 6/e,2005,

**Reference Books**

1. Desikan, Ramesh, ' Software Testing: principles and Practices', Pearson Education
2. Burnstein, 'Practical Software Testing', Springer International Edition,
3. William E. Perry, ' Effective Methods for Software Testing', John Wiley and Sons

4. Stephen H. Kan, 'Metrics and Models in Software Quality Engineering', Pearson Education

## **CS0553 : 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**

**(06 Hrs )**

Review of Applications of Computer Networks, Network hardware and software, Reference Models: OSI and TCP/IP, Example Networks: Internet, X.25, Frame Relay, ATM, Ethernet, Switching, Buffering and Multicasting

**Unit 2 : Data Link Layer**

**( 06 Hrs )**

Design Issues, Error Detection and correction, Examples on Checksum, Stop-and-Wait protocol, Sliding Window protocols, HDLC  
Point-to-Point-Access (PPP): Frame format, Transition states, PPP Stack: LCP, NCP.

**Unit 3 : Medium Access Control**

**(06 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, Wireless LAN, Broad band wireless, Bluetooth.

**Unit 4 : Network Layer**

**(08 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, IP protocol, IP Addresses, IPV6, ICMP, Unicast Routing Algorithms: RIP, OSPF, BGP, Multicast Routing: IGMP, Mobile IP.

### **Unit 5 : Transport Layer**

**( 08 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, Model, protocol, header, connection establishment and release, connection management, Transmission policy, congestion control, timer management, Sockets and Socket programming, RPC, Introduction to wireless TCP and UDP, Performance issues

### **Unit 6 : Application Layer**

**( 08 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 Network Management: SNMP

### **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', Pearson Education ,4<sup>th</sup> Edition,
2. Forouzan B. A, 'Data Communications and Networking', Tata McGraw-Hill Publications, 4th edition, 2006.

## **Reference Books**

1. James F. Kurose, 'Computer Networking- a top-down approach featuring the internet', Person Education, 2nd Edition,
2. Leon-Garcia-Wadjaja, 'Communication Networks- Fundamental Concepts and Key Architectures', Tata McGraw-Hill Publications,
3. Comer D., 'Computer Networks and Internet', Pearson Education, 2<sup>nd</sup> Edition,

# CS0563 : Principles of Programming Languages

**Prerequisites:** C, C++

**Aim:**

This course introduces the various fundamental concepts of programming languages and gives an overview of tools needed to use, evaluate, design and choose programming languages. It explores the motivation behind various programming paradigms such as imperative, procedural, object-oriented and functional. The finer aspects of various programming languages are also introduced through case studies.

**Objectives:**

1. To learn the fundamental concepts of programming languages
2. To understand the need and objectives behind various programming paradigms
3. To learn the important features of different types of languages and their relevance
4. To learn past the superficial differences in languages and recognize the commonalities in meaning.

**Unit 1: Imperative and Procedural Programming (Language: C) (11 Hrs )**

Imperative: Assignments, Mutables, Constants, Operators, Control Structures, Loops, Conditionals

Procedural: Blocks, Local variables, Scope, Activation Records, Procedure Calls, Function Calls, Recursion, Functionality Abstraction and Component Reuse using Procedures and Functions (Structured Programming), Exception Handling, Packages, Namespaces

**Unit 2: Data Structures (Language: C) ( 3 Hrs )**

User-Defined Data Types, User-Defined Data Values (Enum, Property-Value pairs, Data Ranges), Passing and Returning Objects of User-Defined Data Types.

**Unit 3: Object Oriented Programming (Language: C++) ( 4 Hrs )**

Classes and Objects, Methods, Inheritance, Polymorphism, Interfaces, Operators.

**Unit 4: Consistency in Object Oriented Programming (Language: C++) ( 6 Hrs )**

Necessity of maintaining internal consistency of Objects (for large projects), Protection Mechanisms: Public-Private-Protected, Access Control Lists and Friend Classes, References,

Constructors, Anywhere declaration of Local Variables, Destructors, Dynamic Memory Allocation, New, Delete, Garbage Collection.

**Unit 5: Functional Programming (Language: Haskell) ( 6 Hrs )**

Programming as Composition of Operations, Absence of Mutables, Lambda Calculus, Absence of Control Structures, Optimized Implementation of Recursion.

**Unit 6: Case Studies of Programming Languages ( 12 Hrs )**

For each of the following languages, explain their motivation(s), target user base, choice and paradigms of features, special features relevant to target users.

Java, Matlab, SQL, HTML-CSS-Javascript, Actionscript, PHP-HTML, LISP, LEX & YACC, Prolog, Make, VHDL-Verilog-RTL

**Outcomes:**

Upon completion of the course, the students will:

1. Be able to read, understand and use new languages quickly from their descriptions.
2. Have a clear understanding of the different programming language paradigms, the distinction between them and their benefits and drawbacks.
3. Understand how various language features are implemented.
4. Understand why certain design choices or trade-offs are taken while designing a language.
5. Become more effective programmers irrespective of the programming language chosen.

**Text Books:**

1. T. W. Pratt, M.V. Zelkowitz, 'Programming Languages Design and Implementation', Prentice Hall, 4<sup>th</sup> Edition, 2000
2. K. C. Loudon, 'Programming Languages: Principles and Practice', Course Technology, 2<sup>nd</sup> Edition, 2002

**Reference Books :**

1. B. W. Kernighan, D. H. Ritchie, 'The C Programming Language', Prentice-Hall, 2<sup>nd</sup> Edition, 1988
2. B. Stroustrup, 'The C++ Programming Language', Addison-Wesley, 3<sup>rd</sup> Edition, 1997
3. S. Thompson, 'Haskell: The Craft of Functional Programming', Addison Wesley, 2<sup>nd</sup> Edition, 1999
4. A. B. Tucker, R. Noonan, 'Programming Languages: Principles and Paradigms', McGraw-Hill, 2001
5. Individual Language Reference Manuals.

## **IC1903 : Process Control Systems**

### **Prerequisites :**

Understanding of Microprocessors, Peripheral Chips, Analogue Sensors , Conversion , Interfacing Techniques, Digital Logic.

### **Aim :**

To teach the student fundamentals and techniques of Process Control.

### **Objectives:**

To logically translate Process Control Ideas into Control Instrumentation.

### **Unit 1 :**

**( 7 Hrs )**

Sensors: Transducers: Actuators: Principles, Terminology, Classification, Characteristics, Criteria for Selection and Application, Components Devices Assemblies: Resistors Capacitors, Diodes.

### **Unit 2 :**

**( 7 Hrs )**

Signal Conditioning: Bridges ,basic principles .Op Amps , Operations .ADC , DAC , Signal Transmission. EMI Effects and EMC measures.

### **Unit 3 :**

**( 6 Hrs )**

Microprocessor based Instrumentation : Need , Concept of Embedded System ,Processor Study , Memories, H/w interfacing, Test and Measuring Instruments.

### **Unit 4 :**

**( 7 Hrs )**

Principles Analog a, Digital Millimeters, Recorders, Oscilloscope, Performance characteristics.

### **Outcomes:**

1. Determine how to simultaneously meet system and physical design constraints and identify effective trade-offs between concerns to optimize system performance.
2. Determine the hardware and software resources required to interface various kinds of sensors and actuators

**Text Books**

1. C. D. Johnson,' Process Control Instrumentation Technology' 2<sup>nd</sup> Edition
2. C.S. Ranjan ,G .R. Sarma , V.S.V. Mani , 'Instrumentation Devices and System'  
Tata McGraw-Hill ,2<sup>nd</sup> Edition

**Reference Books**

1. Bela Liptak,' Instrumentation and Process Control Handbook'
2. Ramakant Gaikwad, OP-AMP and Integrated Circuits PHI

## **IP0933: Product Design**

### **Aim :**

This course focuses on processes that guide the design and development of high-quality products. The course provides product design specific knowledge such as critical aspects of Product Design, its evaluation and quality issues. The purpose of this course is to teach the basic usability engineering methods for the design and evaluation of software systems and their user interfaces.

### **Objectives:**

1. To understand development of product and its feasibility.
2. To study effect of product development on business plans, team work and growth.

### **Unit 1: Introduction and User Studies**

**(7Hrs)**

Products, Understanding of problem areas and the limitations, Introduction to basic elements and concepts of visual design, Spatial relationship and compositions, Developing questionnaires, interviewing users and observing user and photographic studies of products in use, role of creativity in understanding of latent needs, Comparative analytical studies in other creative fields, Gathering user data, creating work models, Using data in the design process, focus groups, Rapid Assessment Procedure, task analysis, problems faced by users from rural areas, users with special needs, literacy issues etc.

### **Unit 2 : Design and Usability Evaluation**

**(7Hrs)**

Design methodology for complex products, services and events: Design of integrated systems, products for future use, products to be used in groups, devices used in public places, design of multi-modal interfaces, expressive interfaces, products that enrich user experience, Human information processing, human memory, Fitt's law, Hick's law, Human errors, Heuristic evaluation, cognitive walkthroughs, User testing using think aloud protocol and its variations, Field trials and user logs, GOMS, Theoretical models for evaluating products.

### **Unit 3 : Design Management, Human Errors and Professional Practice**

**(7Hrs)**

Designer attributes, Setting up a design office, Finding clients, Business correspondence, Brief and briefing, Letter of contract, Professionalism and Ethics. Costing design and fee estimation, Management of Design Process, human factors, Role of psychology, physiology in interaction design, Designing for Special People, Sensation and perception Cognitive psychology, Human information processing and execution, Human factor in managing team work., Role of staff designer in R&D activities, Design Evaluation, Patent and Design Registration laws and procedure.

#### **Unit 4 : Product Planning and Marketing**

**(7Hrs)**

Corporate strategy for product planning, Management thinking on new products, seeing product as part of the image of the company, Defining companies business, SWOT analysis, Study of product life cycle, Monitoring of sale and competition, Assessing market potentials for new products, market research, Consumer research and its demographic aspects, Establishing market segment and their dimensions, Assessing competitors share and locating direct and indirect sources, Assessing competitors marketing approach and strategies, Developing a strategy to introduce new products, cost considerations and profitability of new products, Developing a product plan and price policy, product positioning, planning for future position.

#### **Outcomes:**

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

1. Describe the human-centered design process, usability engineering process and role in system design and development
2. Discuss usability design guidelines, their foundations, assumptions, advantages and weaknesses
3. Apply key design and evaluation techniques, including observational research, prototyping and heuristic evaluation in the design of a user interface based on analysis of user needs
4. Assess user interfaces using different usability engineering techniques and understand the strengths and weaknesses of different evaluation methods.
5. Learn how to present prototypes and ideas in compelling and convincing way as well as giving and receiving constructive criticism.

#### **Text Books:**

1. Product Design and Development by Karl T. Ulrich and Steven D. Eppinger, McGraw-Hill, 2004.
2. Design Secrets: Products 2: 50 Real-Life Projects Uncovered - Industrial Designers Society of America, Edited by Lynn Haller, Cheryl Dangel Cullen, Publisher: Rockport Publishers Oct 2004.

#### **Reference Books:**

1. What is a Designer: Things, Places, Messages, by Norman Potter, Publisher: Princeton Architectural Pr, 2002.
2. The Design of Everyday Things, by Donald A. Norman, Publisher: Basic Books; 1st Basic edition, September, 2002.
3. History of Modern Design by David Raizman, Publisher: Prentice Hall, 2004.

**Third Year**

**Computer Engineering**

**Second Semester**

**(Practical )**

# CS5553: Computer Networks

**Prerequisites :** Principles of Communications Engineering.

**Objectives:**

- 1.To learn and understand fundamentals of computer network
- 2.To learn and understand network architectures, protocols and applications

## List of Practicals

1. Study any protocol analyzer software (eg. LanExplorer) to learn and use its important features, Study of network monitoring software like ETHREAL software. Assignment to examine TCP/IP and non-TCP/IP protocols (IPX/SPX) and capture them using protocol analyzer Software
2. Study of existing LAN and understand the design and various components. Set up a small network of 3 to 4 computers and Hub/Switch as directed by the instructor. Use Lan Card, UTP Cables and Connectors. Install LAN Cards and Crimp the connectors. Assign unique IP addresses and share C drive on each machine. Test the network by using PING command. Use protocol analyzer Software.
3. Installation and configuration of IIS /PWS/Apache server. Study and use various facilities/commands and features
4. PC-to-PC communication through RS-232 port (COM) using null modem configuration and optical link
5. Study and simulate OSI reference model.
6. Program to implement sliding window protocol (Go back to N)
7. Program to implement sliding window protocol (Selective Repeat)
8. Install two LAN Cards in one of the machine. Install and study router.
9. Simulate Routing protocol (BGP,RIP, Link state routing,flooding).
10. Implement socket TCP : Client in Windows, Server in Linux
11. Implement socket UDP : Client in Linux, Server in Windows
12. Implement any one application layer protocol like HTTP, FTP, SMTP

**Outcomes:**

A student well equipped with the required knowledge and fundamentals to understand and gain insight into advanced computer network courses.

**Note :**

Staff in-charge will suitably frame the above assignments and flexibility may be incorporated. Students will submit term work in the form of a journal, which will include at least 8 assignments. Assignments must contain problem definition, implementation and result.

**Text Books**

1. Tanenbaum A. S., 'Computer Networks', Pearson Education ,4<sup>th</sup> Edition,
2. Forouzan B. A, 'Data Communications and Networking', Tata McGraw-Hill Publications, 4th edition, 2006.

**Reference Books**

1. James F. Kurose, 'Computer Networking- a top-down approach featuring the internet', Person Education, 2nd Edition,
2. Leon-Garcia-Wadjaja, 'Communication Networks- Fundamental Concepts and Key Architectures', Tata McGraw-Hill Publications,
3. Comer D., 'Computer Networks and Internet', Pearson Education, 2<sup>nd</sup> Edition,

## CS5543 : Advanced Development Tools Laboratory

### Objective:

Students should get an idea about the complete end to end Web Development process. This includes understanding the typical two tier Web Setup including the role of the Web servers. Students will understand the choice of technologies available on client and server side.

### UNIT-I: Introduction to JAVA

( 3 Hrs)

Introduction to Programming language like JAVA a desktop or a language for the web environment  
Classes, Constructor, packages, interfaces etc.

### UNIT II: Multithreading and Exception handling

( 3 Hrs)

Multithreading, Exception handling, AWT, Swing, Java I/O.

### Unit-III: Client Side Technologies

( 3 Hrs)

HTML, HTML Forms, DHTML, Client side scripting.(VBScript / Java script)  
Components on Client side:Active X Controls OR Java Applets

### Unit-IV: Server side Programming

( 3 Hrs)

JSP and Servlet, JDBC, Server side technologies Java beans.

### Unit V: Introduction to VB.NET Technology

( 3 Hrs)

Introduction and need of .NET framework, Basics of VB.NET.

### ASSIGNMENT LIST:

1. Write a simple Hello World application using Java swings or AWT with objectives of exposing students to concepts of GUI, and event handler
2. Write a Program to demonstrate the use Exception Handle. Cover at least 3 standard exceptions and 1 of your own exception.
3. Write a Program Producer Consumer to demonstrate the use Multithreading.
4. Write a simple AWT/Swings application to understand basic AWT/SWINGS concepts
5. Develop MVC Architecture, menus, dialog boxes, Basic Graphics Device Interface commands, toolbars, status bars, printing & print previewing.
6. Write a program using servlet (use genericServlet and HttpServlet).
7. Write a program using JSP(Use declaration, expression and standard actions) .
8. Implement Database handling using JDBC using servlet and JSP.
9. Mini project

**Note :** A group for mini project can be formed with maximum 3 students .

### **Text Books**

1. Herbert Schildt, "Java 2: The Complete Reference", Osborne publishing, 5<sup>th</sup> Edition.
2. Steven Holzner, Dreamtech Press , “Visual Basic .Net Black Book” ,2006

### **Reference books:**

1. Jell Prorise /"Programming Windows with MFC" ,2<sup>nd</sup> Edition, Microsoft Press.
2. David Schneider, "An Introduction to Programming Using Visual Basic 6.0", Pren Hall, 1999.
3. Hans Bergsten, "JavaServer Pages", OReilly 3rd Edition,2003.
4. " Java Servlet", OReilly 3rd Edition,2003.

## **IC6803 : Process Control Systems**

### **Prerequisites :**

Microprocessors and Peripherals  
Microprocessors Interfacing Techniques

### **Objectives:**

Introduction of Elements of Control System from the practical perspective and understanding of how these elements affect overall system operation

### **List of Practicals**

1. Interfacing of Temperature Sensor, Calibration and Graphical Display of Temperature with respect to time.
2. Interfacing of Optical Sensor for speed measurement including calibration and
3. output display.
4. Measurement of voltage / current – Signal Conditioning and Calibration.
5. Design of simple ON OFF control system for Temperature monitoring System
6. Experiment on Ladder Programming using PLC for simple tank filling System.
7. Data Transfer using 0-20 mA current Loop.
8. Designing Proportional Controller for simple Temperature Monitoring System.
9. Case Study of PID Controller.

### **Outcomes:**

The student will have understanding of Process Control Elements, and will be in a position to understand and deploy Process Control Systems.

### **Comments:**

1. Teacher has to design 10 -12 Experiments based on above assignments.
2. Instrumentation Setup can be used while performing the assignments.

3. Student can use appropriate programming language like VC++ ,C, Assembly etc..

**Text Books:**

1. Process Control Instrumentation Technology, C. D. Johnson. Pearson Education
- 2 Ramakant Gaikwad, OP-AMP and Integrated Circuits PHI.

**Reference Books :**

- 1 Instrumentation and Process Control Handbook, Bela Liptak
- 2 D. Hall, "Microprocessors and Interfacing", 2nd edition, 1992, McGraw-Hill, ISBN – 0 - 07 - 100462 - 9.