Second Year

Computer Engineering

Pattern - B

First Semester

(Theory)

CS0112 : Data Structure And Algorithms - I

Prerequisites :
Computer fundamentals and Programming.

Aim :
The objective of the course is to introduce the concepts of data structures, algorithms and complexity. The data structures include linear data structures with application of divide and conquer, recursion and randomization algorithms. The course also focuses on sorting and searching algorithms.

Objectives:

1. To understand use of efficient programming and documentation.
2. To introduce algorithmic analysis, fundamental data structures, problem solving paradigms.
3. To study the representation, implementation and application of basic data structures.
4. To introduce algorithmic strategies and time complexity analysis of problems.

Unit 1 : Introduction to Program Development (6 Hrs)

Program Development: Analysis, Design, Coding, Testing and Verification. Introduction to Algorithms: Definition, Characteristics of Algorithm, General guidelines for creating good programs, Algorithm analysis: Frequency count, Space Complexity, Time Complexity: Worst Case, Best Case and Average Case. Basics of 'C': Data Types, Operators, Conditional Statements, Loops, Pseudo code.

Unit 2 : Functions and Pointers (7 Hrs)

Functions: Defining function calls and returns, Actual and formal Parameters, Local and Global variables, scope rules, Recursive functions: Definition, Writing recursive functions, How Recursion works? Applications of recursion, User defined functions, Library functions.

Pointers: Declaration, Pointers and functions parameters, pointers and strings, Parameter passing: call by value and reference, pointer to functions, pointer to pointer, Pointer to array, array of pointers

Unit 3 : Linear data structures (9 Hrs)

Linear data structures using sequential organization: Single dimensional Arrays, Initialization, Memory representation, Use of single dimensional arrays, Strings: array of Characters, String functions using arrays and pointers, Multidimensional arrays, Storage representations (row major, column major and their address calculation), Magic Squares. Concept of ordered list and polynomial representation using arrays, operations on polynomials like add, multiply, evaluate. Matrix operations like add, multiply, transpose, etc. Representation of sparse matrix, algorithm for sparse matrix addition, simple and fast transpose. Time and space complexity analysis for simple and fast transpose.

Unit 4 : Searching and sorting **(7 Hrs)**

Introduction to Searching techniques: Linear, Index sequential and binary Search
Sorting Techniques: Sort Order, efficiency (Time and Space complexity analysis) and passes, Bubble sort, insertion sort, selection sort, shell sort on numbers and character strings, bucket sort, Concept of Divide and Conquer strategy for merge sort, quick sort.

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

Fundamentals of stack, Representation and Implementation of stack using arrays, Applications of stack: Decimal to Binary Conversion, reversing a string, parsing: Well-form parenthesis, Different expression conversions and evaluation, representation of multiple stacks, Simulating recursion using stack.

Unit 6 : Queues **(6 Hrs)**

Fundamentals of queue, Representation and Implementation of queue using arrays, Circular queue: representation and implementation, Applications of queue: Josephus Problem, Job Scheduling, Queue Simulation, Categorizing Data, Double Ended Queue, representation of multiple queues, Priority queue.

Outcomes:

Upon Completion of the course, the students will be ale to:

1. Understand of how several fundamental algorithms work, particularly those concerned with sorting and searching
2. Understand of the fundamental data structures used in computer science
3. Analyze the space and time efficiency of most algorithms
4. Design new algorithms or modify existing ones for new applications and reason about the efficiency of the result

Text Books:

1. Y. Langsam, M.J.Augenstein, A.M.Tenenbaum, 'Data structures using C and C++', Pearson Education, Second Edition, 2002.

2. E.Balaguruswamy,' C and Data Structures' TMH Publication, 2003

Reference Books :

1. E. Horwitz , S. Sahani, D. Mehta, 'fundamentals of Data Structures in C++', Galgotia book source, New Delhi, 1995.
2. M. Weiss, 'Data structures and Algorithm Analysis in C++', Pearson Education, 2nd Edition, 2002
3. J. Tremblay, P. soresan, 'An Introduction to data Structures with applications', TMH Publication, 2nd Edition, 1984.

4. A. Drozdek, 'Data Structures in C++', 2nd Edition, Thomson Brookes / COLE Books, 2002.

CS0122: Numerical Methods and Simulation

Prerequisites: Elementary knowledge of 1) Numbers 2) Differentiation and integration 4) Differential equations 5) Algebra 6) Probability

Objectives:
The course explains the importance and symbiosis between Mathematics and Engineering. This course module has helped to develop skills of numerical techniques, probability, queuing models. The objective of the course is to achieve fluency with Numerical tools which is an essential weapon in modern Graduate Engineer's Armory and the balance between the development of understanding and mastering of numerical techniques.

Unit-1. Introduction to Numerical Computation (7 Hrs)
Forward Difference, Backward Difference, Central Difference, Relation between operators, Interpolation (Lagrange's / Newton's/ Sterling), Least square approximation, Curve-Fitting method**.**

Unit-2. Ordinary Differential Equations: Initial value Problems (7 Hrs)
Introduction, Difference Equations, Numerical Methods, Singles step methods, Stability Analysis, Multi Step Methods, Predictor-Corrector Methods, Stability Analysis of Multi Step methods.

Unit-3. Ordinary Differential Equations: Boundary value Problems. (7 Hrs)
Introduction, Initial Value Problem Method (Shooting Method), Finite Difference Methods, Finite Element Methods.

Unit-4. Discrete and Continuous Probability distribution and density Function (7 Hrs)
Random Variables, Random Processes, Bernoulli Trials , Pascal, Binomial ,Poisson, Normal, Beta, Cauchy, Chi-Square, Erlang, Exponential, Gamma, Delta, Laplace, Log-Normal, Maxwell, Normal- Bi-variate, Rayleigh, Uniform, Weibull

Unit-5. Sampling and Information Theory. (7 Hrs)
Sampling Theory, Sample mean, Sample variance, Sampling Distribution, Confidence Interval, Hypothesis Testing.
Information Theory: Measuring Information, Entropy, Minimum and Maximum Entropy, Interpretation of Entropy, Joint Entropy, Conditional Entropy, Huffman Codes.

Unit-6. Queuing Theory. (7 Hrs)
Queuing Models- Little's Theorem**,** Probabilistic Form of Little's Theorem**,** Application of Little's Theorem**,** Poisson Arrival Model , M/M/1 Queuing System**,** Arrival Statistics, Service Statistics**,** Markov Chain Formulation**,** Deviation of the Stationary Distribution**,** Occupancy Distribution upon Arrival**,** Occupancy Distribution upon Departure**,** The M/M/m, M/M/∞, M/M/m/m, AND Other Markov Systems**,** The M/G/1 System, M/G/1 Queues with Vacations**,** Reservations and Polling**,** Priority Queuing.

OUTCOMES:
By the end of this module students are expected to demonstrate the knowledge of

1. Difference operators, Different Interpolation Methods and the curve fitting.
2. Euler's Method, Euler's Modify Method, Runge Kutta's Fourth Order Method and Adom Bashforth Method for First Order Differential Equations with their Stability Analysis
3. Shooting Method for Initial Value Problem.
4. Finite Difference Method for Boundary Value Problem and Rayleigh Ritz's method, Galarkin Method, Finite Element Ritz's Method (Second Order Differential Equations only)
5. Random Variables and Random Processes and various Probability Distributions.
6. Sample mean, Sample variance, Sampling Distribution, Student's t Test, Chi Square test, Measuring Information, Entropy, Interpretation of Entropy, Joint Entropy, Conditional Entropy, Huffman Codes.
7. Little's Theorem, Poisson Arrival Models, Markov Chain Formulation, Different Markov Systems**,** The M/G/1 System, M/G/1 Queues with Vacations**,** Reservations and Polling**,** Priority Queuing.

TEXT BOOKS:

1. M. K. Jain, S.R.K Iyengar & R .K. Jain, 'Numerical Methods for scientific & engineering computations', Wiley Eastern Ltd., New Delhi, 2nd edition, 1991.
2. Gross D. and Harris C., 'Fundamentals of Queuing Theory', John Wiley & Sons , New York, 3rd Edition, 1985

REFERENCE BOOKS:

1. S. M. Ross, 'Probability models for computer science', Academic Press, 1st Indian Edition, 2006.
2. Albert Leon-Gracia, 'Probability and Random Processes for electrical engineering', University of Toronto, Pearson Education, 2nd Edition, 2004.
3. S. M. Ross, 'Simulation by S. M. Ross', John Wiley & Sons, 2nd Edition, 2002.
4. Allen, A.O., 'Probability, Statistics and Queuing theory', Academic Press,…..1981
5. R. B. Cooper, McMillan, 'Introduction to Queuing theory', New York, …….1972.
6. Erwin Kreyszig, 'Advanced Engineering Mathematics', John Wiley and sons, inc., 8th Edition, 2003
7. Robert B. Ash, 'Information Theory', Dover Publications, Inc. New York, 1st South Asian Edition, 2007

CS0132 : Computer Organization

Aim:
To learn structure, function and characteristics of computer systems and understand the design of its various functional units.

Objectives:
1. To understand the structure, function and characteristics of computer systems
2. To understand the design principle of the various functional units of digital computers
3. To understand Parallel Processing concepts.

Unit 1: Structure of a Computer System (6 Hrs)

Brief History of computers, Von Neumann Architecture, Functional Units
Data Types and Computer Arithmetic: Fixed and Floating point numbers, Signed numbers, Integer Arithmetic, 2's Complement multiplication, Booths Algorithm, Hardware Implementation, Division, Restoring and Non Restoring algorithms, IEEE standards for Floating point representations,

Unit 2: Control Unit Design (6 Hrs)

Machines instructions and addressing modes, Single Bus CPU, Control Unit Operation: Instruction Sequencing, Micro-operations. Hardwired Control: Design methods, Design examples: Multiplier CU. Micro-programmed Control: Basic concepts, Microinstruction-sequencing and execution, Micro-program control, Applications of microprogramming.

Unit 3: Processor Design (7 Hrs)

CPU Architecture, Register Organization, Instruction Set - Instruction types, instruction formats (Intel, Motorola processors), Instruction cycles, Instruction pipelining, Types of operands, Addressing Modes (Intel, Motorola processors), ALU design-ALU organization. RISC and SUPERSCALAR Processors: RISC- Features, Register File, RISC v/s CISC, Superscalar Processors- Overview, Organization.

Unit 4 : Memory Organization (8 Hrs)

Characteristics of memory systems, Internal and External Memory
Main Memory- ROM, PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM, error correction.
Virtual Memory: Main Memory allocation, Segmentation, Paging. High-Speed Memories: Cache Memory, organization and Mapping, Replacement Algorithms, Cache Coherence, MESI protocol. Interleaved and Associative Memories, Performance Characteristics, Secondary Storage: Magnetic Disk, RAID, Optical memory, CDROM, DVD

Unit 5 : I/O Organization (6 Hrs)

Input/Output Systems, Programmed I/O, Interrupt Driven I/O, I/O channels, Buses and standard Interfaces: Synchronous, Asynchronous, Parallel, Serial, PCI, SCSI, USB Ports Peripherals: Keyboard, Mouse, Scanners, Video Displays, Dot Matrix, Desk-jet and Laser Printers

Unit 6 : Multiprocessor Configurations (9 Hrs)

Multiprocessor Configurations:
Closely coupled and loosely coupled multiprocessor architectures, Problems of bus contentions, Inter-process Communications, Coprocessor and I/O Processor, Bus structures and Interconnection networks, Performance Bus controller, Bus Arbitration, System Bus.
Introduction to parallel processing:
Trend towards parallel processing, Parallelism in Uniprocessor systems. Parallel computer structures, Architectural classification schemes, Parallel processing applications.

Outcomes:

At the end of the course students will:

1. Obtain knowledge of integer and floating point arithmetic, algorithms for multiplication and division.
2. Obtain Knowledge about the control unit design.
3. Acquire the information about Intel, Motorola Processor's addressing modes, Instruction pipelining, overview of RISC,CISC and Superscalar processor
4. Understand memory types, secondary memory devices, virtual & cache memory concepts.
5. Study the details of different input and output devices with their functions.
6. Understand multiprocessor configuration, Coprocessor and I/O Processor, and parallel processing concept.

Text Books

1. C. Hamacher, V. Zvonko, S. Zaky, "Computer Organization", McGraw Hill, 2002
2. Hwang and Bringgs – Computer Architecture and Parallel Processing-TMH.

Reference Books

1. W. Stallings William, "Computer Organization and Architecture: principles of structure and function", Maxwell Macmillan Editions, 2nd Edition, 1990, ISBN 0 – 02 – 946297 – 5
2. A. Tanenbaum, "Structured Computer Organization", Prentice Hall of India, 4th Edition, 1991, ISBN 81 – 203 – 1553 – 7 .
3. G. George, "Computer Organization: Hardware and Software", Prentice Hall of India, 2nd Edition ,1986 .
4. D. Paterson, J. Hennesy, "Computer Organization and Design: The Hardware Software Interface", Morgan Kauffman, 2nd Edition, 2000 ISBN 981 – 4033 – 588.
5. W. Stallings, "Computer Organization and Architecture: Designing for performance", Prentice Hall of India, 2003, 6th Edition, ISBN 81 – 203 – 2103 – 0,

6. J. Hays, “Computer Architecture and Organization”, McGraw-Hill, 2nd Edition 1988 ISBN 0 – 07 – 100479 – 3.

EC4992 : Digital Electronics And Logic Design

Prerequisites:
Basic Electronics Engineering

Aim:

To Learn basics of digital electronics and its practical application.

Objectives:
To understand combinational and sequential circuits and their applications and get introduced to programmable devices and their implementation in simple circuits.

Unit 1: Number system and Codes (6 Hrs)

Introduction, Binary number System, Sign-Magnitude representation, One's and Two's complement representation, Binary arithmetic, 2's complement arithmetic, Octal number System, Hexadecimal number System, BCD code, Excess-3 code, Gray code. Code conversion, Boolean algebra: Basic theorems and properties

Unit 2: Combinational logic circuits: (8 Hrs)

K-Map: Representation of truth-table, SOP form, POS form, Simplification of logical functions, Minimization of SOP and POS forms using K- Map (K-maps up to 4 variables only). Don't care conditions Design examples- multiplexers : code converter, demultiplexers, encoder: Priority encoders, decoders: 74138, BCD-to-7-segment Decoder, Half adder and subtractor, full adder and subtractor, adder with look ahead carry, BCD adder and subtractor, ALU 74181, parity generator and checker, Digital comparator: 7485

Unit 3 : Sequential logic circuits (9 Hrs)

Introduction of flip-flop (F.F), 1 bit memory cell, clocked S-R F.F., J-K F.F race around condition, M/S J-K F.F, flip-flop truthtable, excitation table, flip-flop conversion, flip-flop characteristics. T and D F.F, Design of 4 – bit UP-Down ripple counter using J-K flip-flop , Design of Synchronous 3 bit up/down counter, mod-n counters (IC 7490,74190,7493), Moore/Mealy M/c's: representation techniques, state diagrams, state tables, state reduction, state assignment, implementation using flip-flops. Applications like sequence generator and detector. Shift register (modes of operation), 4bit bi-directional universal shift register (using D or J-K FF), application of shift registers (Ring counter, Sequence generator, Johnson's counter).

Unit 4: Logic Families (7 Hrs)

Characteristics of Digital ICs: Speed, Power dissipation, fan-out, current and voltage parameters, noise margin, operating temperature etc., TTL: Operation of TTL NAND gate, Standard TTL, TTL Characteristics, Active pull-up, Wired-AND, totem pole, open collector, Unconnected Inputs. CMOS Logic: CMOS Inverter, CMOS NAND and NOR, CMOS characteristics. Wired-logic, Unconnected Inputs, Open-Drain Outputs, Comparison of TTL

and CMOS, interfacing TTL to CMOS and vice versa, tri-state logic: tri-state buffers, inverters, Study of Data sheets of 7400 Series ICs: (Basic and Universal logic gates)

Unit 5 : Programmable Logic Devices (6 Hrs)

Programmable Logic array: Input, Output Buffers, AND, OR, Invert/Non-Invert Matrix, Programming the PLA, Applications of PLAs to implement combinational and sequential logic circuits Introduction to :FPGA, CPLD

.

Unit 6 : ASM, VHDL (6 Hrs)

ASM charts, notations, design of simple controller, multiplexer controller method, RTL notations and implementation

Introduction to VHDL: Modeling Digital systems, modeling languages, modeling concepts

Outcomes:

At the end of the course students will:

1. Get adequate knowledge of Number systems, Boolean algebra and concept of code conversion
2. Understand K-map Simplification and designing of different combinational circuit.
3. Get knowledge about 3-bit,4 bit counters, shift register, sequence generator, sequence detector
4. Be Familiar with TTL and CMOS logical families with their characteristics.
5. Understand Programmable Logic array, FPGA, CPLD, ASM
6. Be able to design simple combinational circuit: half adder and subtractor using VHDL language.

Text Books

1.R.P. Jain, "Modern Digital Electronics", Tata McGraw-Hill, 3rd Edition ,2003, ISBN 0 - 07 - 049492 - 4

2. J. Bhaskar, “A VHDL Primer”. Englewood Cliffs, NJ: Prentice Hall, 1995.

Reference Books

1. M. Mano, "Digital Design", Pearson Education, 3rd Edition ,2002, ISBN - 81 - 7808 - 555 - 0

2. J. Yarbrough, "Digital Logic: Applications and Design", Thomson Brooks/Cole (Vikas Publishing House, New Delhi), 2002, ISBN 981 - 240 - 062 - 1

3. J. Bignell, R. Donovan, "Digital Electronics", DELMAR Thomson Learning, 4th Edition, 2001, ISBN 981 - 240 - 352 - 3

4. A. Malvino, D. Leach, "Digital Principles and Applications", Tata McGraw Hill, 5th Edition, 2003, ISBN 0 - 07 - 047258 – 05

5..J. Wakerly, "Digital Design - Principles and Practices", Pearson Education,

EC4972 : ELECTRONIC DEVICES AND CIRCUITS

Prerequisite: **Basic Electronics Engineering.**

Aim: To study various electronic components and its applications**.**

Objectives

1. To study principles of electronic devices and circuits
2. To study applications of electronic devices and circuits
3. To study and build electronic circuits

Unit 1:
Transistor, Biasing and Small Signal Amplifiers (8 Hrs)

Transistor Biasing Circuits:
Overview of Biasing, Fixed Bias, Collector Base Bias, Self Bias, Definition and derivation for Stability Factors S, numerical for Self Bias Circuit only, Comparison of the bias circuits, Need of and techniques for bias compensation, Thermal run away, Derivation for Thermal Stability. Analysis of C.E. Amplifier using hybrid parameters.
FET: Symbol, Operation, Construction, Drain and Transfer Characteristics and Parameters

Unit 2: Multi Stage Amplifiers (6 Hrs)

Need of Cascading, Block Diagram of Cascaded Amplifier, Effect on Gain and Bandwidth, Analysis of Two Stage R-C Coupled Amplifier in CE-CE, configurations for determination of Voltage gain (A_v), Current gain (A_i), Input Impedance (R_i) and Output Impedance (R_o) Concept of Large Signal Amplification, Large Signal Amplifiers – Classification on the Basis of Q point location, Comparison of various Amplifier Classes

Unit 3: Operational Amplifiers (7Hrs)

Block Diagram of an OP-Amp, Direct Coupled Differential Amplifier Circuit, Level Shifter and the Output Stage. Op-Amp Parameters: Bias, Offset current; Bias, offset voltage; CMRR, slew rate, PSRR Definitions, Significance and Typical Values

Op-Amp Applications –Adder, Subtractor, Precision Rectifier, Zero Crossing Detector, Schmitt Trigger, Waveform Generator (triangular), Integrator, Differentiator, Instrumentation Amplifier, Voltage to Current, Current to Voltage Circuits using Op-Amp

Unit 4: Power Devices and Power Supplies (7 Hrs)

Power Devices: - Characteristics and principle of Operation of SCR Applications: Half-wave and Full-wave Controlled rectifiers. Introduction to DIAC,TRIAC.
Power supplies: - Introduction to SMPS, types of SMPS: Buck, Boost, Buck-Boost SMPS, online UPS, off line UPS, Line interactive UPS

Outcome:
At the conclusion of the course student will be able to

1. Differentiate between biasing circuits and significance of self-bias circuit Basic concepts of h- parameter.
2. Understand the effect of gain and bandwidth on cascade amplifier, comparison of various power amplifiers.
3. Study Op-Amp and its applications
4. Acquire knowledge about Power devices and their application; get introduced to different types of SMPS.

Textbooks:

1. J. Millman, C. Halkias, "Integrated Electronics Analog and Digital Circuits and Systems", McGraw Hill, 1988, ISBN 0 – 07 – Y85493 – 9
2. R. Gaikwad, "Op-Amp and Linear Integrated Circuits", 4th Ed, Prentice Hall of India, 2002, ISBN 81 – 203 – 2058 – 1

Reference books:

1. T. Floyd, "Electronic Devices", 5th Ed, Pearson Education, 2001, ISBN 81–7808–355-8
2. Liao, "Microwave Devices and Circuits", 3rd Edition, Prentice Hall of India, ISBN 81 – 203 – 0699 – 6
3. N. Mohan, T. Undeland, W. Robbins, "Power Electronics, converters, applications and design", 2nd Edition, John Wiley, 1995, ISBN 9971 – 51 – 177 – 0
4. R. Boylstad, L. Nashelsky, "Electronic Devices and Circuit Theory", 8th Ed, Pearson Education, ISBN 81 – 7808 – 590 – 9, 2002
5. G. Deboo, C. Burrous, "Integrated Circuits and Semiconductor Devices, Theory

Second Year

Computer Engineering

First Semester

Practicals

CS5112 : Data Structure And Algorithms - I

Prerequisites:
Computer fundamentals and Programming.

Aim :
The objective of the course is to introduce the concepts of data types, data structures, algorithms and complexity. The data structures include linear data structures with application of divide and conquer, recursion and randomization algorithms. The course also focuses on sorting and searching algorithms.

Objectives:

5. To understand use of efficient programming and documentation.
6. To introduce algorithmic analysis, fundamental data structures, problem solving paradigms
7. To study the representation, implementation and application of basic data structures.
8. To introduce algorithmic strategies and time complexity analysis of problems.

List of Practicals

1 Write a C Program to find a factorial of a number using recursive and non recursive functions.
2 Write a C Program to implement set operations using arrays - union, intersection, difference, symmetric difference.
3 Write a C program to perform various string operations using arrays such as copy, length, substring, palindrome, etc. (without built in functions)
4 Write a C program to perform various string operations using pointers such as length, concat, reverse, compare etc. (without built in functions)
5 Write a C Program to represent single variable polynomial using array and perform addition, multiplication of them.
6 Write a C program to implement matrix operations like addition, multiplication, transpose, etc.
7 Write a program to create a magic square matrix.
8 Write a C program to represent sparse matrix using array and perform sparse matrix addition, simple & fast transpose.
9 Write a C program to implement sorting methods- bubble, insertion, bucket, selection sort. Display results after every pass.
10 Write a C Program to implement quick sort and merge sort.
11 Write a C program to implement binary search method - with and without recursion.

12 Write a C Program to create a database (such as employee, student) using array of structures with options like create, insert, delete, modify, print reverse, display etc.
13 Write a C program to implement stack using array and perform Push and Pop operations on it.
14 Write a C program to check the parenthesis validity of the given expression using stack.
15 Write a C program to convert infix expression to postfix and evaluate it using stack.
16 Write a C program to convert infix expression to prefix and evaluate it using stack.
17 Write a C program to implement multiple stacks using array.
18 Write a C program to implement circular queue using array and perform add and delete operations on it.
19 Write a C program to implement priority queue.
20 Write a C program to implement double ended queue.

Outcomes:

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

5. Understand of how several fundamental algorithms work, particularly those concerned with sorting and searching
6. Understand of the fundamental data structures used in computer science
7. Analyze the space and time efficiency of most algorithms
8. Design new algorithms or modify existing ones for new applications and reason about the efficiency of the result

Text Books:

1. Y. Langsam, M.J.Augenstein, A.M.Tenenbaum, 'Data structures using C and C++', Pearson Education, Second Edition, 2002.

2. Balaguruswamy,' C and Data Structures' TMH Publication, 2003

Reference Books :

1. M. Weiss, 'Data structures and Algorithm Analysis in C++', Pearson Education, 2nd Edition, 2002
2. J. Tremblay, P. soresan, 'An Introduction to data Structures with applications', TMH Publication, 2nd Edition, 1984.
3. A. Drozdek, 'Data Structures in C++', 2nd Edition, Thomson Brookes / COLE Books, 2002.
4. E. Horwitz , S. Sahani, D. Mehta, 'fundamentals of Data Structures in C++', Galgotia book source, New Delhi, 1995.

CS5122: Numerical Methods and Simulation

Objectives:

The course explains the importance and symbiosis between Mathematics and Engineering. This course module has helped to develop skills of numerical techniques, probability, queuing models. The objective of the course is to achieve fluency with Numerical tools which is an essential weapon in modern Graduate Engineer's Armory and the balance between the development of understanding and mastering of numerical techniques.

List of Practicals

1) Program for Lagrange Interpolation Method
2) Program for Cubic Spline Interpolation for arbitrary set of points . The second derivative at the end points are assumed as zeros (natural spline).
3) Program to solve an IVP , dy/dx =f(x,y), y(x0)=y0, using Euler Method. The initial values x0, y0, the final value xf and the step size are to be read. F(x,y) is written as a function subprogram Program to solve an IVP , dy/dx =f(x,y), y(x0)=y0, using Milne Simpson Method The initial values x0, y0, the final value xf and the step size h are to be read. Starting values are calculated using classical fourth order Runge Kutta method. Adams-Bashforth method of third order is used as a predictor and Milne-Simpson method is iterated till abs(yold-ynew)<=err where err is error tolerance.
4) Program to solve the linear two point boundary value problem

   U^n = p[x] (du/dx) + q[x]u+r[x] = G (x, u, du/dx) , u[a]=s1, u[b]=s2 by shooting method using the superposition principle. The initial value problem is solved by the fourth order Runge Kutta Method for 2x2 system. It requires two approximations of the slope of the solution curve at the starting point of integration. The linear function G is given as a function subprogram.
5) Write a program to generate Huffman variable length code.
6) Write a program which will find the

   1) minimum, maximum entropy

2) Joint entropy

3) Conditional entropy for the given information

7) For given data find the measures of central tendency and dispersion. Also find coefficient of skewness.

8) Write a program to find the binomial and poison probabilities by receiving input data from the user.

9) Study of Pro-Model simulation software used for queuing problems.

10) Demonstrate any simulation software for queuing applications to explore the software capability.

OUTCOMES:

By the end of this module students are expected to demonstrate the knowledge of

1. Difference operators, Different Interpolation Methods and the curve fitting.
2. Euler's Method, Euler's Modify Method, Runge Kutta's Fourth Order Method and Adom Bashforth Method for First Order Differential Equations with their Stability Analysis
3. Shooting Method for Initial Value Problem.
4. Finite Difference Method for Boundary Value Problem and Rayleigh Ritz's method, Galarkin Method, Finite Element Ritz's Method (Second Order Differential Equations only)
5. Random Variables and Random Processes and various Probability Distributions.
6. Sample mean, Sample variance, Sampling Distribution, Student's t Test, Chi Square test, Measuring Information, Entropy, Interpretation of Entropy, Joint Entropy, Conditional Entropy, Huffman Codes.
7. Little's Theorem, Poisson Arrival Models, Markov Chain Formulation, Different Markov Systems**,** The M/G/1 System, M/G/1 Queues with Vacations**,** Reservations and Polling**,** Priority Queuing.

Text Books

1. M. K. Jain, S.R.K Iyengar & R .K. Jain ,'Numerical Methods for scientific & engineering computations', Wiley Eastern Ltd., New Delhi,2nd Edition,1991.
2. Gross D. and Harris C., John Wiley & Sons,' Fundamentals of queuing theory', New York, 1985
3. Robert B. Ash,' Information Theory', Dover Publications, Inc. New York, 2007.

Reference Books

1. S.M.Ross, 'Probability models for computer science',Academic Press, 2006.

2. Albert Leon-Gracia, University of Toronto,'Probability and Random Processes for electrical engineering', by Pearson Education, 2nd Edition, 2004.

3. S. M. Ross ,'Simulation ', Academic Press, 2002.

4. Allen, A.O., "Probability, Statistics and Queuing theory", Academic Press, 1981.

5. R.B.Cooper,' Introduction to Queuing theory', McMillan , New York, 1972 (Introductory)

EC9992 : Digital Electronics And Logic Design

Objectives:

To build and verify simple combinational and sequential circuits.

List of Practicals

I Combinational Logic Design

1. Verification of Logical Gates and Boolean Algebra.
2. Code converters, e.g. Excess-3 to BCD and vice versa using logical gates.
3. Multiplexer - e.g. 16:1 Mux using 4:1 Mux (IC 74153).
4. Decoder – e.g. 2 bit comparator (IC 74138).
5. BCD adder –using IC 7483.

II Sequential Circuit Design (Any six)

1. Conversion of flip-flops. e.g. JK to D,T.
2. Ripple (asynchronous) mod –N counter using J-K F-F.
3. Ripple (asynchronous) mod –N counter using IC 7490
4. Synchronous 2 bit Up /down counter using JK flip-flop .
5. Sequence generator using JK flip-flop
6. Pseudo random number generator using 74194.(universal shift register)
7. Sequence detector (Moore ckt) using JK flip-flop
8. Sequence detector (Mealy ckt) using JK flip-flop

III ASM (Any one)

1. Simple ASM using multiplexer controller method using Simulator.
2. Design of simple combinational circuit: half adder and subs tractor using VHDL language

Outcomes: At the end of the course students will:

1. Implement code conversion
2. Use K-map for Simplification and designing of different combinational circuit.
3. Implement 3-bit,4 bit counters, shift register, sequence generator, sequence detector
4. Design ASM using simulator.
5. Be able to design simple combinational circuit: half adder and subs tractor using VHDL language.

Text Books

1. R.P. Jain, "Modern Digital Electronics", Tata McGraw-Hill, 3rd Edition ,2003, ISBN 0 - 07 - 049492 - 4
2. J. Wakerly, "Digital Design - Principles and Practices", Pearson Education,

Reference Books

1. M. Mano, "Digital Design", Pearson Education, 3rd Edition ,2002, ISBN - 81 - 7808 - 555 - 0
2. J. Yarbrough, "Digital Logic: Applications and Design", Thomson Brooks/Cole (Vikas Publishing House, New Delhi), 2002, ISBN 981 - 240 - 062 - 1
3. J. Bignell, R. Donovan, "Digital Electronics", DELMAR Thomson Learning, 4th Edition, 2001, ISBN 981 - 240 - 352 - 3
4. A. Malvino, D. Leach, "Digital Principles and Applications", Tata McGraw Hill, 5th Edition, 2003, ISBN 0 - 07 - 047258 – 0

Second Year

Computer Engineering

Second Semester

Theory

CS0162 : Data Structures And Algorithms - II

Prerequisites :
Data structures and algorithms - I

Aim :
This course addresses techniques for programming with collections of data, and the data structures and algorithms that are needed to implement these collections. The course emphasizes the practical application of techniques for writing and analyzing programs.

Objectives:
1 To study the representation, implementation & applications of data structures.
2 To compare alternative implementations of data structures.
3 To compare the benefits of static and dynamic data structures.
4 To choose the appropriate data structure for modeling a given problem.

Unit 1 : Linked Representation (8 Hrs)

Dynamic Memory allocation, Array representation using dynamic memory allocation, Concept of linked organization, singly linked list, doubly linked list, circular linked list, Insertion, Deletion and traversal on above data structures. Representation and manipulations of polynomials using linked lists. Generalized Linked List, operations on GLL like copy, equality.

Unit 2 : Basic Trees (8 Hrs)

Review of basic terminology, binary trees and its representation using sequential and linked organization, full and complete binary trees, creation of a binary tree, binary tree traversals (recursive and non recursive), operations such as copy, equal etc. Binary search tree, creation of binary Search tree, finding height and counting leaf nodes of a binary search tree (with and without recursion), Finding mirror image of the binary search tree with and without recursion, Deletion of a node from a binary search tree, printing a tree level wise and depth wise.

Unit 3 : Advance Trees and indexing Techniques (9 Hrs)

Threaded binary trees, Creation of in-order threaded binary tree and traversing it by in-order, pre-order and post-order way, Insertion and deletion of nodes in threaded binary tree, applications of binary trees: Gaming, Expression tree, Heap sorting; Symbol table, OBST, AVL trees, Huffman's algorithm, Indexing techniques, B- tree, B+tree, red-black tree, AA trees

Unit 4 : Graphs (8 Hrs)

Review of basic terminology, Representation of graphs using adjacency matrix, adjacency list, traversals: Depth First and Breadth First, Connected components and spanning trees, Kruskal's and Prim's algorithms for minimum spanning tree, Algorithm for shortest path-Dijkstra's algorithm

Unit 5 : Hashing (5 Hrs)

Hash tables, hash functions: Division, folding and mid square methods, collusion resolution strategies: linear probing, rehashing, open addressing and chaining, chaining with and without replacement, table overflow, extendible hashing**.**

Unit 6 : Files (4 Hrs)

External storage devices, files: definition and concepts, file organizations: Sequential, Random, Linked. Processing of sequential, Index-sequential and direct files.

Outcomes:

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

1. Use principles for good program design, especially the uses of data abstraction and modular program composition;
2. Use major techniques for implementing the fundamental data types (linked lists, binary search trees, hashing, heaps, etc.)
3. Understand how the choice of data structures and algorithm design methods impacts the performance of programs.
4. Implement specific data structures such as binary trees, heaps, binary search trees, and graphs.

.**Text Books:**

1. Y. Langsam, M.J.Augenstein, A.M.Tenenbaum, 'Data structures using C and C++',

   Pearson Education, Second Edition, 2002.

2. E. Horwitz , S. Sahani, D. Mehta, 'fundamentals of Data Structures in C++', Galgotia book source, New Delhi, 1995.

Reference Books :

1. M. Weiss, 'Data structures and Algorithm Analysis in C++', Pearson Education, 2nd Edition, 2002
2. J. Tremblay, P. soresan, 'An Introduction to data Structures with applications', TMH Publication, 2nd Edition, 1984.
3. A. Drozdek, 'Data Structures in C++', 2nd Edition, Thomson Brookes / COLE Books, 2002.

CS0172 : Microprocessor Interfacing Techniques

Prerequisites:
Understanding of Digital Logic Circuits: Buffers, Flip-flops, Gates, Shifters.. Basic Electronics Transistors Diodes, LED's.

Aim:
To acquire knowledge of Microprocessors fundamentals.

Objectives:
To explore internal architecture of microprocessor, signal and interface with memory and I/O devices.

Unit 1: Introduction to 8086 microprocessor (7 Hrs)

Introduction to32 bit 8086 microprocessor, Internal Architecture, Pin Diagram, Min Max Mode, study of 8086 supporting chips 8282, 8284, 8286, 8288. Timing Diagram Read Write Machine Cycles.

Unit 2: Assembly language programming of 8086 (7 Hrs)

Address Decoding, Even-Odd Memory Banks and Accessing Memory and I/O ports. Programmer's model, Addressing Modes, Instruction sets, Assembler Directives, Far near Procedures

Unit 3: Interrupt structure of 8086 (6 Hrs)

Interrupt Structure , Interrupt service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR ,NMI , Interrupt Response, Priority if interrupts.

Unit 4: Interfacing with 8086-I (8 Hrs)

Priority interrupt controller (8259)-block diagram, cascade connections of PIC's
Programmable interrupt timer/counter (8254)-block diagram, control word & interfacing

Programmable peripheral interface (8255)-block diagram, control word, interfacing ADC, DAC, stepper motor in mode 0

Unit 5: Interfacing with 8086-II (8 Hrs)

Keyboard display interface (8279)-Block Diagram , encoded & decoded mode, Interfacing & programming Asynchronous , Synchronous Communication, RS 232C
Universal synchronous asynchronous receiver transmitter 8251-Block Diagram, Interfacing, Programming

Unit 6: DOS and TSR's (6 Hrs)

DOS, Internals, Loading, Memory Map, POST sequence, PSP, .EXE, .COM. BIOS Calls, INT 10H, DOS calls-INT 25H. TSR's, Type –active, passive, TSR Structure, Loading & Writing TSR's

Outcomes:

At the end of the course student will be able to design microprocessor based systems. Students have a scope to further enhance their abilities in learning subjects like embedded systems.

Text Books:

1. Douglas Hall, **"**Microprocessors and Interfacing", 2nd Edition, McGraw Hill, 2003.
2. Peter Abel " Assembly Language Programming" , Pearson Education , 5th Edition,2001

Reference Books:

1. Ray Duncan, " Advanced MS DOS Programming",2nd Edition, BPB publications,2002
2. A. ray, K. Bhurchandi, "Advanced microprocessors and peripherals :Architecture ,programming & interfacing",4th edition, Tata McGraw-Hill,2004
3. INTEL – Microprocessor and Peripheral Handbook VOL I
4. Intel data manual on microprocessors Vol. II

CS0182 : Object Oriented Programming And Computer Graphics

Aim :
The aim of this Syllabus is to teach students the basic concepts of Object Oriented Programming (OOP) and also about fundamentals of Computer Graphics.

Objectives:
1. To understand basic concepts of Object Oriented Programming.
2. To understand inheritance, polymorphisms, templates, file handling.
3. To understand basic concepts of computer graphics.
4. To understand algorithms to draw various graphics primitives.
5. To understand 2-D and 3-D transformations.

Unit 1 : Introduction to Object Oriented Programming (7 Hrs)

Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming, fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism, overloading. Extensions to C: Variable declarations, global scope, 'const', reference variables, comments, default parameters, inline functions, default and constant arguments.
Classes and Objects: Defining a class, data members and methods, access specifiers, Local classes, inline member functions, default and constant arguments, Scope Resolution Operator

Memory Management Operators, Manipulators, constructors & types, destructors, array of objects, pointers and classes, class as ADTs.

Unit 2 : Inheritance (7 Hrs)

Function overloading: Function overloading, function with default arguments.
Concept and need, single inheritance, base and derived classes, friend classes, types of inheritance, member access control-private, public & protected, friend function, static data members, static member functions, static class, constructors in derived classes, multiple inheritance- ambiguity, virtual base class, container classes.

Unit 3 : Polymorphism (7 Hrs)

Introduction, compile time & run time polymorphism.
Operator Overloading: Introduction, Need of operator overloading, overloading the assignment, binary and unary operators, 'this' pointer, overloading using friends, rules for operator overloading, type conversions.
Virtual functions: Void pointers, pointer to function and objects, pointers and memory Management, pointers to derived classes, abstract base class, pure virtual functions, and virtual destructors. Introduction, Function template and class template, function overloading vs. function templates, member function templates and template arguments.

Unit 4 : Managing Console I/O Operations (7 Hrs)

Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators Files and Streams: Concept of a file, file operations, streams, opening and closing a file, detecting end-of-file, file modes, file pointer, structures and files, classes and files, sequential file processing Updating a file-Random access, Error handling , Command Line Arguments

Unit 5 : Fundamentals of Graphics (7 Hrs)

Introduction to computer graphics, lines, line segments, vectors, pixels and frame buffers, vector generation, Generation of line using DDA algorithm and Bresenham's line drawing algorithm, Generation of circle using Bresenham's algorithm, DDA algorithm and Mid point algorithm.Polygon filling: Seed fill scan line algorithm, Edge fill, Boundary fill, Flood fill.

Window clipping: Line clipping using Cohen Sutherland algorithm, Polygon Clipping using Sutherland-Hodgman algorithm.

Unit 6 : 2D & 3D Transformations (7 Hrs)

2D Transformations: Translation, Rotation, Scaling, Reflection and Shear.
3D Transformations: Introduction, 3-D geometry, primitives ,translation, scaling, rotation about any axis not parallel x, y or z axis, aliasing and antialiasing techniques in graphics

Curves and Fractals : Introduction, Curve generation, Interpolation, interpolating algorithms, interpolating polygons, B-Spline and corners, Bezier curves, Fractals, fractal lines and surfaces.

Outcomes:

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

1. Correctly implement a class in the target language, including the appropriate use of public and private methods and data.

2. Implement a subclass of the existing class in order to extend the functionality of the base class in order to solve the stated problem.
3. Design and implement a reusable collection class (templates), creation and manipulation of files.
4. Implementation of graphics primitives such as line drawing, polygon filling, clipping.
5. Implementation of 2D & 3D concepts in computer graphics.

Text Books :

1. E. Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Publishing Company Ltd, New Delhi ISBN 0 - 07 - 462038 - X.
2. J. Foley, V. Dam, S. Feiner, J. Hughes, "Computer Graphics Principles and Practice", Pearson Education, 2nd Edition.,2003, ISBN 81 - 7808 - 038 – 9.

Reference Books:

1. R. Lafore, "The Waite Group's Object oriented Programming in C++", Galgotia Publications, 3rd Edition ,2001, ISBN 81-7515-269-9.
2. B. Stroustrup, "C++ Programming Language", Pearson Education, , 3rd Edition.,1997, ISBN 0 - 201 – 327554.
3. Herbert Schildt," C++ Complete Reference", TMH
4. D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publication, 2nd Edition, 2001, ISBN 0 - 07 - 047371 – 4,.
5. F. Hill, "Computer Graphics: Using OpenGL", Pearson Education, 2nd Edition, 2003 ISBN 81 - 297 – 0181 – 2,
6. D. Hearn, M. Baker, "Computer Graphics - C Version", Pearson Education, 2nd Edition, 2002, ISBN 81 7808 - 794 – 4
7. S. Harrington, "Computer Graphics McGraw-Hill Publications, 2nd Edition, 1987, ISBN 0 – 07 -100472- 6,
8. D. Rogers, J. Adams, "Mathematical Elements for Computer Graphics", Tata McGraw-Hill Publication, 2nd Edition.,2002, ISBN 0 - 07 - 048677 – 8.

ES0132 : ENGINEERING MATHEMATICS - III

Prerequisites: Engineering Mathematics- I and Engineering Mathematics - II

Objectives:

The course explains the importance and symbiosis between Mathematics and Engineering. The objective of the course is to achieve a fluency with Mathematical tools which is an essential weapon in modern Graduate Engineer's Armory and the balance between the development of understanding and mastering of solution techniques with emphasis being on the development of student's ability to use Mathematics with understanding to solve Engineering problems by retaining the philosophy of "learning by doing".

Unit-1. Logic and Proofs (7 Hrs)

Propositions, Conditional Propositions, Logical Connectivity, Prepositional calculus, Universal and Existential Quantifiers, Proofs, Proof Techniques, Mathematical Induction. Set Theory - Set, Combination of sets, Finite and Infinite sets, Un-countably infinite sets, Principle of inclusion and exclusion

Unit-2. Relations and Functions (7 Hrs)

Definitions, Properties of Binary Relations, Equivalence Relations and partitions, Partial ordering relations and lattices, Chains and Anti chains. Definitions, domain, Range, One-to-One and On-To, Inverse and Composition, Pigeonhole Principle, Discrete Numeric functions and Generating functions, Job scheduling Problem. Recurrence Relation, Linear Recurrence Relations With constant Coefficients, Homogeneous Solutions, Total solutions, solutions by the method of generating functions

Unit-3. Algebra and Number Theory (7 Hrs)

Groups, Ring, Field, GF Fields. Modulo Arithmetic: Modulo Operator, Set of Residues, Modular Exponentiation, Congruence, Operations in Zn, Multiplicative and Additive Inverses. Fermat and Euler Theorems. Euclid's Algorithm, Primitive roots, Chinese Remainder Algorithm.

Unit-4. Linear Differential equations of higher order (7 Hrs)

Homogeneous Linear Differential Equations of Second Order, Higher Order Homogeneous & Non Homogeneous Linear Differential Equations with Constant Coefficients, Solutions by undetermined coefficients and Variation of Parameters method, Euler – Cauchy Equation, Application of system of ordinary differential equations.

Unit-5. Complex Variables and Z Transform (7 Hrs)

Derivative, Analytical function, Cauchy-Riemann equations, Complex Integration, Cauchy's Integral Theorem and formula, Residue Theorem and applications to Engineering Problems, Introduction to Z Transform, properties of Z-Transform, Inverse Z-Transform, application of Z-transform to difference equations.

Unit-6. Fourier and Laplace Transform (7 Hrs)

Complex Fourier series and frequency spectrum, Fourier integrals, Fourier cosine and sine transforms, Fourier transforms. Introduction to Laplace Transform and its properties. Laplace Transform of Unit step function, Delta function and periodic function. Inverse Laplace Transform and its evaluation. Application of Fourier transform and Laplace Transform to Engineering Problems.

OUTCOMES:

By the end of this module students are expected to demonstrate the knowledge of

1. Mathematical Statements and connectivities, Quantifiers, Set Theory, Un-countably infinite sets, Principle of inclusion and exclusion
2. Binary Relations & its Properties, Different types of Relations, partitions, Partial ordering relations and lattices, Functions, Pigeonhole Principle, Discrete Numeric functions and Generating functions.
3. Elementary knowledge of Group theory, Euclid's Algorithm, Primitive roots, Chinese Remainder Algorithm
4. Linear differential equations for modeling of a linear systems and its solutions by classical, transform techniques and numerical methods.
5. Cauchy-Riemann equations, Contour Integration.
6. Solutions of difference equations by using Z-transforms.
7. Complex Fourier series and frequency spectrum, Fourier transforms and its properties, Laplace Transform and its properties. Application to solve Engineering problem.

TEXT BOOKS:

1. Erwin Kreyszig, 'Advanced Engineering Mathematics', John Wiley and sons, inc., 8th Edition, 2003.
2. B.S. Grewal, 'Higher Engineering Mathematics', Khanna Publishers, Delhi, 38th Edition, 2000.

REFERENCE BOOKS:

1. Murray R. Spiegel 'Advanced Calculus' by [Schaum's out line series],1981

2. Thomas, G. B. and Finney,'Calculus and analytic Geometry, Wesley/Narosa, (6th Edition) 1985.
3. Dennis G. Zill and Michael R. Cullen, 'Advanced Engineering Mathematics', CBS New Delhi, (2nd Edition), 2000.
4. Michael D.Greenberg, 'Advanced Engineering Mathematics', Prentice Hall International, (second Edition), 1998.
5. C. Ray Wylie, Louis C Barrett R, 'Advanced Engineering Mathematics', McGraw-Hill Book Company, 6th Ed,2003.
6. Pipes and Harvill, 'Applied Mathematics for Engineers and Physicists', McGraw-Hill Book Company, 3rd Ed. 1984.
7. Larry C., Andrews & Ronald L. Philips, 'Mathematical Techniques for Engineers & scientists', PHI Pvt. Ltd., New Delhi, Indian reprint by SPIE, 2005.
8. Alan Jeffrey, 'Advanced Engineering Mathematics', Academic Press, 1st Ed., 2002.
9. David M. Burton, 'Elementary Number Theory', Mcgraw Hill Inc New York, 2nd edition,1995.
10. Ivan Niven, Herbert S. Zuckerman and Hugh L. Montgomery, 'The Theory of Numbers' John Wiley & Sons Inc. New York, Fifth Edition, 2001
11. Tremblay and Manohar, 'Discrete Mathematical Structures with applications to Computer Science', Tata Mcgraw-Hill, 23rd Reprint,2005.

EC4882 : Principles of Communication Engineering

Aim :
To gain proficiency in electronic communication concepts to be applied in advanced networking domains

Objectives:
1. To understand the basics of signals , modulation
2. Understand multiplexing techniques
3. Application of the above mentioned concepts in the real world
4. Introduction to Computer Networks

Unit 1 : Introduction to Electronic Communication (7 Hrs)

The importance of Communication, Elements of communication system, Types of electronics communication, survey of communication applications, Electromagnetic spectrum, Bandwidth
Introduction to Fourier Series, Fourier Transform- DFT, IFT and properties of Fourier transform.

Unit 2 : Modulation Techniques (7 Hrs)

Principles of Amplitude Modulation, Modulation index and percentage of modulation, AM power distribution, Single sideband communication, ISB modulation, AM transmitters and Receivers
Frequency modulation principles, Phase modulation, FM with binary signals, sideband and modulation index, FM vs AM
Examples : TV signal, TV receiver
Receivers : Superheterodyne receiver, Frequency conversion, Intermediate frequency selection and Images.

Unit 3 : Multiplexing and Telephony (7 Hrs)

Introduction, FDM , TDM, Pulse code modulation, Delta modulation, PAM, Sampling theorem
Transmission lines
Telephone System, Fascimile, Cellular Telephone System, Paging System.

Unit 4 : Data Communication (7 Hrs)

Digital Communication concepts, Modems, Null modems, ASK, PSK, FSK
Error detection and correction: LRC, VRC, CRC, Hamming code, Checksum
Introduction to Computer Networks : OSI model, Spread Spectrum, Fibre Optic communication

Outcomes:

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

Text Books :

1. Frenzel, Communication Electronics Principles and Applications, Tata McGraw Hill Publication., Third Edition.
2. Kennedy, Principles of Communication Systems, McGraw Hill
3. Roddy Coolen, Electronic Communication

Reference Books :

1. Taub and Schilling, Principles of Communication
2. Behrouz Ferouzan, Data and Computer Communication

Second Year

Computer Engineering

Second Semester

Practicals

CS5162 : Data Structures And Algorithms - II

Prerequisites :
Data structures and algorithms - I

Aim :
This course addresses techniques for programming with collections of data, and the data structures and algorithms that are needed to implement these collections. The course emphasizes the practical application of techniques for writing and analyzing programs.

Objectives:

3 To study the representation, implementation & applications of data structures.
4 To compare alternative implementations of data structures.
3 To compare the benefits of static and dynamic data structures.
4 To choose the appropriate data structure for modeling a given problem.

List of Practicals

1. Write a C program to create a database (such as employee, student) using single linked list with options like create, insert, delete, modify, search, print reverse, display etc.
2. Write a C program to accept binary numbers in doubly linked list & perform addition of them and store the result in another list.
3. Write a C program to create two sorted singly linked lists, and merge these two lists into third list without creating a new linked list.
4. Write a C program to create a GLL and perform accept, copy and display functions on it.
5. Write a C program to create a binary tree and traverse it in preorder, postorder and inorder way, both by recursion and non-recursion.
6. Write a C program to create a binary search tree and find height & number of leaf nodes with and without recursion and print leaf nodes also.
7. Write a C program to create a binary search tree. Find its mirror image. Print original and mirror image level wise.
8. Write a C program to create an inorder threaded binary tree and perform all three traversals.
9. Write a C program to represent a given graph using adjacency list and perform DFS and BFS.
10. Write a C program to represent a given graph using adjacency array and find the shortest path using Dijkstra algorithms.
11. Write a C program to represent a given graph using adjacency list and generate a minimum spanning tree using Kruskal's / Prim's algorithms.
12. Write a C program to create a hash table and handle the collisions using linear probing with or without replacement.
13. Write a C Program to create a hash table and handle the collisions using chaining with or without replacement.
14. Write a C program to create a text file, read it and compute frequency of vowels, count words, characters, lines and white spaces and write these results into another text file. Use command line arguments.
15. Write a C program for implementation of sequential file and perform insertion of record, display, delete and modify, search operations on it.
16. Write a C program for implementation of simple index file and perform insertion of record, display, delete and modify operations on it.
17. Write a C program for implementation of direct access file – Insertion and deletion of a record from a direct access file chaining with or without replacement.
18. Mini project which will make use of different data structures learnt in this subject.

Outcomes:

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

1. Use principles for good program design, especially the uses of data abstraction and modular program composition;
2. Use major techniques for implementing the fundamental data types (linked lists, binary search trees, hashing, heaps, etc.)

3. Understand how the choice of data structures and algorithm design methods impacts the performance of programs.
4. Implement specific data structures such as binary trees, heaps, binary search trees, and graphs.

Text Books:

1. Y. Langsam, M.J.Augenstein, A.M.Tenenbaum, 'Data structures using C and C++', Pearson Education, Second Edition, 2002.

2. E. Horwitz , S. Sahani, D. Mehta, 'fundamentals of Data Structures in C++', Galgotia book source, New Delhi, 1995.

Reference Books :

1. M. Weiss, 'Data structures and Algorithm Analysis in C++', Pearson Education, 2nd Edition, 2002
2. J. Tremblay, P. soresan, 'An Introduction to data Structures with applications', TMH Publication, 2nd Edition, 1984.
3. A. Drozdek, 'Data Structures in C++', 2nd Edition, Thomson Brookes / COLE Books, 2002.

CS5172: Microprocessor Interfacing Laboratory

Prerequisites:
Understanding of Digital Logic Circuits: Buffers, Flip-flops, Gates, Shifters.. Basic Electronics Transistors Diodes, LED's

Objectives:
To study programming model of 8086 and assembly language programming.

List of Practicals

Group A:- (Any 6)

1. Study of 8086 Architecture and Execution of sample programs.

2. Write 8086 Assembly language program to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.

3. Write 8086 ALP to perform block transfer operation.(Don't use string operations) Data bytes in a block stored in one array transfer to another array..
Use debugger to show execution of program.

4 Write 8086 ALP to find and count negative number from the array of signed number stored in memory.

5. Write 8086 Assembly language program (ALP) to arrange the numbers stored in the array in ascending as well as descending order. Assume that the first location in the array holds the number of elements in the array and successive memory location will have actual array elements. Write a separate subroutine to arrange the numbers in ascending and descending order. Accept a key from the user.

a. If user enters 0, Arrange in ascending
b. If user enters 1, Arrange in descending

6. Write 8086 Alp to convert 2_digit HEX number into equivalent BCD number.

7. Write 8086 ALP to convert 2_digit BCD number into equivalent HEX number.

8. Write 8086 Assembly language program (ALP) for following operations on the string entered by the user.

a. Concatenation of two strings
b. Find number of words, characters

Group B:- (Any 6)

9. Write 8086 ALP to convert an analog signal in the range of 0V to 5V to its corresponding digital signal using successive approximation ADC.

10. Write 8086 ALP to interface DAC & generate following waveforms on oscilloscope. Comment on types of DAC's and write detailed specifications of the DAC used

i) Square wave -- Variable Duty Cycle & frequency.
ii) Stair case wave
iii) Triangular wave

11. Write 8086 ALP to rotate a stepper motor for

a) one clockwise rotation

b) one anti clockwise rotation

Write routines to accelerate and de-accelerate the motor

Modify your program to rotate stepper motor for given angle and given direction.

12. Write 8086 ALP to program 8253 in Mode 0 . Generate a square wave with a pulse of 10 mS. .

13. Write 8086 ALP to initialize 8279 & to display characters in right entry mode. Provide also the facility to display "SECOMP".

1. Character in left entry mode
2. Rolling Display
3. Flashing Display

14. Perform an experiment to establish communication between two USART's Initialize USART-A in asynchronous transmitter mode and interface USART-B by initializing it in asynchronous receiver mode.

15. Consider 8086 based microprocessor system with following specifications:

1. Memory:16Kb Monitor program in ROM, 8Kb Scratch Pad RAM
2. Peripherals: 8255, 8253.
3. I/O Devices: six 7 Segment Displays and Hex Key Pad through 8279.

Draw the entire system block diagram showing the details of memory decoding, chip selection logic, latches, buffers, system clock, power on RESET, address data and Control lines.

16. Write TSR program in 8086 ALP to handle the "Divide by zero" interrupts. Test your program with a small code, which causes the divide by zero interrupt.

17 Write TSR program in 8086 ALP to implement Real Time Clock (RTC). Read the Real Time from CMOS chip by suitable INT & FUNCTION & display the RTC at the bottom right corner on the screen. Access the video RAM directly in your routine.

Note: - Students should perform any 6 assignments from group A and any 6 assignments from group B

Outcomes:

At the end of course, student should get well acquaint with assembly language programming and concepts of TSR's.

Text books:

1 Douglas Hall,'Microprocessors and Interfacing', McGraw Hill, 2nd Edition.

2 Ray Duncan, 'Advanced MS DOS Programming', Latest Edition.

Reference Books:

1 Peter Abel, "Assembly Language Programming", Pearson Education, 5th Edition.

2 INTEL – Microprocessor and Peripheral Handbook VOL I

CS5182 : Object Oriented Programming And Computer Graphics

Aim :

The aim of this Syllabus is to teach students the basic concepts of Object Oriented Programming (OOP) and also about fundamentals of Computer Graphics.

Objectives:

1. To understand basic concepts of Object Oriented Programming.
2. To understand inheritance, polymorphisms, templates, file handling.
3. To understand basic concepts of computer graphics.
4. To understand algorithms to draw various graphics primitives.
5. To understand 2-D and 3-D transformations.

List of Practicals

Group A

1. Write a program for implementation of constructors, destructors, new, delete operators.

2. Write a program for implementing function overloading and function with default argument.

3. Write a program to implement friend function, function overloading, function with default arguments, constructors etc.

4. Write a program to create a database of any information system and provide the facilities for Insertion, Deletion, Modification, Search and Display using multiple inheritance.

5. Write a program to implement hybrid inheritance.

6. Write a program to implement class for overloading '-(hyphen)', '++','--', '+', '-', '<<','=','==' and '>>' operators.

7. Write a program to implement virtual function.

8. Write a program to perform various operations by using Templates.

9. Declare class 'STACK' and 'QUEUE' and handle the run time anomalies .

10. Write a program to maintain the details of any system using files. Implement functions to add, delete, modify, search and display records.

11. Write a program for file handling by using virtual function.

Group B

1. Write a program to draw any shape using Graphics Primitives such as Pixel, Line, Circle, Ellipse, Polygons, and Line styles.

2. Write a program to draw any shape using DDA & Bresenham's line algorithm.

3. Write a program to draw any shape using DDA,Bresenham's & Mid point circle algorithm.

4. Write a program to implement algorithm for filling a polygon using
   a) scan line method.
   b) Flood fill
   c) Edge fill
   d) Boundary fill

5. Write a program to implement Line clipping using Cohen Sutherland algorithm.

6. Write a program to implement Polygon clipping using Sutherland Hodgman algorithm.

7. Write a program to implement 2-D transformations such as translation, rotation, scaling shearing.

8. Write a program to implement 3-D transformations such as translation, rotation, scaling shearing.

Outcomes:

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

1. Correctly implement a class in the target language, including the appropriate use of public and private methods and data.
2. Implement a subclass of the existing class in order to extend the functionality of the base class in order to solve the stated problem.
3. Design and implement a reusable collection class (templates), creation and manipulation of files.
4. Implementation of graphics primitives such as line drawing, polygon filling, clipping.
5. Implementation of 2D & 3D concepts in computer graphics

Text Books :

1.E. Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Publishing Company Ltd, New Delhi ISBN 0 - 07 - 462038 - X.

2. S. Harrington, "Computer Graphics McGraw-Hill Publications, 2nd Edition, 1987, ISBN 0 – 07 -100472- 6,

Reference Books:

1. R. Lafore, "The Waite Group's Object oriented Programming in C++", Galgotia Publications,
3rd Edition ,2001, ISBN 81-7515-269-9.
2. B. Stroustrup, "C++ Programming Language", Pearson Education, , 3rd Edition.,1997, ISBN 0 - 201 – 327554.
3. Herbert Schildt," C++ Complete Reference", TMH
4. D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publication,
2nd Edition ,2001, ISBN 0 - 07 - 047371 – 4,.
5. J. Foley, V. Dam, S. Feiner, J. Hughes, "Computer Graphics Principles and Practice", Pearson Education, 2nd Edition.,2003, ISBN 81 - 7808 - 038 – 9.
6. F. Hill, "Computer Graphics: Using OpenGL", Pearson Education, 2nd Edition., 2003 ISBN
81 - 297 – 0181 – 2,
7. D. Hearn, M. Baker, "Computer Graphics - C Version", Pearson Education, 2nd Edition'

,2002, ISBN 81 7808 - 794 – 4,
8. D. Rogers, J. Adams, "Mathematical Elements for Computer Graphics", Tata McGraw-Hill
Publication, 2nd Edition.,2002, ISBN 0 - 07 - 048677 – 8.
9. Kanetkar,' Let us C++