

Objectives:

To develop problem-solving techniques and explore topics in a variety of areas of discrete mathematics, including but not limited to logic, graph theory, set theory, recursions, combinatorics, and algorithms. Students will learn to express statements in the language of formal logic and draw conclusions, model situations in terms of graph and set theory, find and interpret recursive definitions for mathematical sequences, use combinatorial methods to approach counting problems.

Outcomes:

- Ability to Illustrate by examples the basic terminology of functions, relations, and sets and demonstrate knowledge of their associated operations.
- Ability to Demonstrate in practical applications the use of basic counting principles of permutations, combinations, inclusion/exclusion principle and the pigeonhole methodology.
- Ability to represent and Apply Graph theory in solving computer science problems.

UNIT I

Mathematical Logic: Statements and notations, Connectives, Well formed formulas, Truth Tables, tautology, equivalence implication, Normal forms. Theory of inference for the statement calculus: Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving. Predicate calculus: Predicative logic, Free and Bound variables, The Universe of Discourse. Inference theory of predicate calculus involving quantifiers.

UNIT II

Relations: Properties of binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Lattices, Hasse diagram. Functions: Composition of functions, Inverse Function, Hashing functions, Natural numbers, recursive functions.

Algebraic structures: Algebraic systems, Examples and general properties, Semi groups and monoids, groups and sub groups, homomorphism. Lattice as partially ordered sets, Boolean algebra.

UNIT III

Elementary Combinatorics: Basics of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial and Multinomial theorems, the principles of Inclusion – Exclusion, Pigeon hole principles and its application.

UNIT IV

Recurrence Relations: Generating Functions, Function of Sequences, Calculating Coefficients of generating functions, Recurrence relations, Solving recurrence relation by substitution and Generating functions, the method of Characteristic roots, solution of Inhomogeneous Recurrence Relations.

UNIT V

Graphs: Basic Concepts, Isomorphism and Sub graphs, Trees and their properties, Spanning Trees, Directed trees, Binary trees, Planar Graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

TEXT BOOKS:

1. Discrete Mathematics with Applications to Computer Science, J P Trembley and R Manohar, TMH, 2008. (Units I and II)
2. Discrete Mathematics for Computer Scientists and Mathematicians, second edition, J.L.Mott, A. Kandel, T.P. Baker, PHI

REFERENCE BOOKS:

1. Elements of Discrete Mathematics- A Computer Oriented Approach, C.L.Liu, D.P. Mohapatra, 3rd edition, TMH.
2. Discrete and Combinatorial Mathematics- An Applied Introduction-5th Edition–Ralph. P.Grimaldi, Pearson Education.
3. Discrete Mathematics with applications, Thomas Koshy, Elsevier.
4. Discrete Mathematical Structures, Mallik and Sen, Cengage Learning.
5. Discrete Mathematical Structures, Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, PHI.
6. Discrete Mathematics and its Applications, 6th edition, K.H.Rosen, TMH.
7. Discrete Mathematics, Lovasz, Springer.
8. Logic and Discrete Mathematics, Grass Man and Tremblay, Pearson Education.
9. Discrete Mathematics, S K Chakraborty and B K Sarkar, Oxford, 2011.

Objectives:

- To understand how computers are constructed out of a set of functional units.
- To understand how these functional units operate, interact and communicate.
- To understand the factors and trade-offs that affect computer performance.
- To understand concrete representation of data at the machine level.
- To understand how computations are actually performed at the machine level.
- To understand how problems expressed by humans are expressed as binary strings in a machine.
- Understand the system interconnection and the different I/O techniques.
- Explain the functioning and programming of the INTEL-8086.
- Understand the design of processors, the structure and operation of memory and virtual memory, cache, storage, and pipelining, system integration, and peripherals.
- Identify the different architectural and organizational design issues that can affect the performance of a computer such as Instruction Sets design, Pipelining, RISC architecture, and Superscalar architecture.
- Design an interconnection networks and multiprocessors.

Outcomes: After this course students understand in a better way the I/O and memory organization in depth. They should be in a position to write assembly language programs for various applications.

UNIT I

Number Systems and Computer Arithmetic: Signed and unsigned numbers, Addition and subtraction, multiplication, division, Floating point representation, logical operation, Gray code, BCD codes, Error detecting codes, Boolean algebra, Simplification of Boolean expressions, K-Maps, Combinational and Sequential Circuits- decoders, Encoders, Multiplexers, Half and Full adders, Shift registers, Sequential circuits- flip-flops.

UNIT II

Memory Organization: Memory hierarchy, Main memory-RAM, ROM chips, Memory address map, memory contention to CPU, Associative Memory-Hardware logic, match, read and write logic, Cache Memory-Associative mapping, Direct mapping, Set-associative mapping, hit and miss ratio.

UNIT III

Basic CPU Organization: Introduction to CPU, Instruction formats-INTEL-8086 CPU architecture-Addressing modes - generation of physical address- code segment registers, Zero, one, two, and three address instructions. INTEL 8086 Assembly Language Instructions-Data transfer instructions-input- output instructions, address transfer, Flag transfer, arithmetic, logical, shift, and rotate instructions. Conditional and unconditional transfer, iteration control, interrupts and process control instructions, assembler directives, Programming with assembly language instructions.

UNIT IV

Input -Output Organization: Peripheral devices, input-output interface-I/O Bus and interface modules, I/O versus Memory bus, isolated versus memory mapped I/O, Modes of transfer-Programmed I/O, Interrupt-initiated I/O, priority interrupts-Daisy chaining, parallel priority, interrupt cycle, DMA- DMA control, DMA transfer, Input output processor-CPU-IOP communication.

UNIT V

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

Multi Processors: Characteristics or Multiprocessors, Interconnection Structures, Interprocessor Arbitration, InterProcessor Communication and Synchronization, Cache Coherence, Shared Memory Multiprocessors.

TEXT BOOKS:

1. Computer System Architecture, M. Morris Mano, 3rd Edition, Pearson Education, 2008.
2. Microprocessors and Interfacing, Douglas Hall, Tata McGraw-Hill.

REFERENCE BOOKS:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5th Edition, McGraw Hill.
2. Fundamentals of Computer Organization and Design, Sivarama P. Dandamudi, Springer Int. Edition.
3. Computer Organization and Architecture, William Stallings, 8th Edition, Pearson, 2007.
4. Digital Design, M. Morris Mano, Pearson Education.
5. Computer Organization and Design, D. A. Paterson and John L. Hennessy, Elsevier.
6. Computer Architecture and Organization, M. Murdocca and V. Heuring, Wiley India.

MCA - I Year - I Sem

COMPUTER PROGRAMMING USING C

Objectives:

- To understand the various steps in Program development.
- To understand the basic concepts in C Programming Language.
- To learn how to write modular and readable C Programs
- To learn to write programs (using structured programming approach) in C to solve problems.
- To introduce the students to basic data structures such as lists, stacks and queues.
- To make the student understand simple sorting and searching methods.

Outcomes:

- Demonstrate the basic knowledge of computer hardware and software.
- Ability to apply solving and logical skills to programming in C language and also in other languages.

UNIT I

Introduction to Computers: Computer Systems, Computing Environments, Computer Languages, Creating and running programs, Program Development.

Introduction to the C Language: Background, C Programs, Identifiers, Types, Variables, Constants, Input / Output, Operators(Arithmetic, relational, logical, bitwise etc.), Expressions, Precedence and Associativity, Expression Evaluation, Type conversions.

UNIT II

Statements- Selection Statements(making decisions) – if and switch statements, Repetition statements (loops)-while, for, do-while statements, Loop examples, other statements related to looping – break, continue, goto, Simple C Program examples.

Functions: Designing Structured Programs, Functions, user defined functions, inter function communication, Standard functions, Scope, Storage classes-auto, register, static, extern, scope rules, type qualifiers, recursion- recursive functions, Limitations of recursion, example C programs, Preprocessor commands.

UNIT III

Arrays: Concepts, using arrays in C, inter function communication, array applications, two – dimensional arrays, multidimensional arrays, C program examples.

Pointers: Introduction (Basic Concepts), Pointers for inter function communication, pointers to pointers, compatibility, Pointer Applications-Arrays and Pointers, Pointer Arithmetic and arrays, Passing an array to a function, memory allocation functions, array of pointers, programming applications, pointers to void, pointers to functions.

UNIT IV

Strings – Concepts, C Strings, String Input / Output functions, arrays of strings, string manipulation functions, string / data conversion, C program examples.

Enumerated, Structure ,and Union Types– The Type Definition(typedef), Enumerated types, Structures –Declaration, initialization, accessing structures, operations on structures, Complex structures, structures and functions, Passing structures through pointers, self referential structures, unions, bit fields, C programming examples, command –line arguments.

UNIT V

Input and Output – Concept of a file, streams, text files and binary files, Differences between text and binary files, State of a file, Opening and Closing files, file input / output functions (standard library input / output functions for files), file status functions (error handling)Positioning functions, C program examples.

TEXT BOOKS:

1. A Structured Programming Approach Using C, B.A.Forouzan and R.F. Gilberg, Third Edition, Cengage Learning.
2. Programming in C. P. Dey and M Ghosh , Oxford University Press.

REFERENCE BOOKS:

1. C& Data structures – P. Padmanabham, Third Edition, B.S. Publications.
2. Problem Solving and Program Design in C.J.R. Hanly and E.B. Koffman, 7th Edition, Pearson education.
3. Programming in C. Ajay Mittal, Pearson.
4. Programming with C. B.Gottfried, 3rd edition, Schaum's outlines, TMH.
5. Problem solving with C,M.T.Somasekhara, PHI.
6. Programming with C R.S.Bickar, Universities Press.
7. Computer Programming & Data Structures, E.Balagurusamy, 4th edition, TMH.
8. Programming in C – Stephen G. Kochan, III Edition, Pearson Education.

9. The C Programming Language, B.W. Kernighan and Dennis M.Ritchie, PHI.
10. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

MCA - I Year - I Sem

PROBABILITY AND STATISTICS

Objectives:

- Understand the concepts of Probability, random variables and their distributions.
- Understand the concepts of estimation and hypothesis testing for population averages and percentages.
- Select and produce the appropriate tabular and graphical formulas for displaying bivariate data sets and carry out correlation, regression and chi-square analyses.

UNIT I

Probability: Sample space and events – Probability – The axioms of probability - Some elementary theorems - Conditional probability – Baye's theorem.

UNIT II

Random variables – Discrete and continuous – Distribution – Distribution function.
Distribution - Binomial, Poisson and normal distribution – related properties.

UNIT III

Sampling distribution: Populations and samples - Sampling distributions of mean (known and unknown) proportions, sums and differences.
Estimation: Point estimation – interval estimation - Bayesian estimation.

UNIT IV

Test of Hypothesis – Means and proportions – Hypothesis concerning one and two means – Type I and Type II errors.
One tail, two-tail tests. Tests of significance – Student's t-test, F-test, t^2 test. Estimation of proportions.

UNIT V

Curve fitting: The method of least squares – Inferences based on the least squares estimations - Curvilinear regression – multiple regressions – correlation for univariate and bivariate distributions.

TEXT BOOKS:

1. Probability and statistics for engineers (Erwin Miller and John E.Freund), R A Johnson and C.B.Gupta.7th edition, PHI.
2. Introduction to Probability and Statistics, 12th edition, W.Mendenhall, R.J.Beaver and B.M.Beaver, Cengage Learning.

REFERENCE BOOKS:

1. Text book of Probability and Statistics Dr.Shahnaz Bathul, V.G.S.Publishers 2003.
2. Probability and Statistics in Engineering, 4th Edition, William W.Hines, Douglas C.Montgomery, David M.Goldsman, Connie M.Borrer, Wiley Student Edition.
3. Probability, Statistics and Queuing Theory, 2nd Edition, Trivedi, John Wiley and Sons
4. Introduction to Probability and Statistics, J.S.Milton, Jesse C.Arnold, 4th edition, TMH.
5. Probability, Statistics and Random Processes, Dr.K.Murugesan, P.Guruswamy, Anuradha Agencies, Deepti Publications.
6. Probability and Statistics for Engineers, G.S.S.Bhisma Rao, Sitech, 2nd edition,2005.
7. Probability and Statistics for Engineers and Scientists, R.E.Walpole, S.L.Myers,K.Ye, Pearson.
8. Probability, Statistics and Reliability for Engineers and Scientists, B.M.Ayyub, R.H.McCuen, 2nd edition, Chapman&Hall/CRC, Special Indian Edition.
9. Probability and Statistics for Engineers and Scientists, Sheldon M Ross, 4th ed., Elsevier, 2011.

ACCOUNTING AND FINANCIAL MANAGEMENT

This course is designed to introduce students to the principles, concepts, and applications of financial accounting and management.

An introduction to accounting emphasizing how general purpose financial statements communicate information about the business corporation's performance and position for users internal and external to management. Approximately one third of the course emphasizes how the accountant processes and presents the information and includes exposure to recording transactions, adjusting balances and preparing financial statements for service and merchandise firms according to established rules and procedures.

This course presents the underlying framework and concepts of Financial Accounting in the context of how accounting fits into the overall business environment of contemporary society. Financial accounting is the basic means of recording and reporting financial information in a business. Students will learn how accounting functions as an information development and communication system that supports economic decision making and provides value to entities and society. Students will discover the uses and limitations of financial statements and related information and apply analytical tools in making both business and financial decisions. Topics examined include those related to corporate financial position, operating results, cash flows, and financial strength. Students will study the basic accounting system and will be shown how the various accounting alternatives for recording financial transactions impact on the usefulness of the information provided for decision-making. During coverage of relevant topics reference will be made to recent lapses in ethical reporting and the resulting impact on the financial markets and society.

In Financial Management students are introduced to concepts and tools that enable them to think critically about the financial opportunities and challenges faced by an organization. In working through the material, they build upon the understanding they already have of accounting, business, and mathematics. They learn how to use financial statements such as balance sheets, income statements, and statements of cash flow. They prepare budgets, analyze investment options, and determine the best means of financing business endeavors. They discover ways of assessing both the return and the risk involved in a firm's financial decisions. The focus of this course is on solving practical business problems similar to those encountered in the workplace.

Objectives

- Explain financial accounting terminology and the recording process.
- Discuss and apply the basic principles of accounting, the accounting model, and the accounting cycle.
- Demonstrate and show how items such as assets, liabilities, share capital, etc., are reported in the financial statements.
- Carry out journal entries and adjustments.
- Prepare and analyze a trial balance and financial statements.
- Analyze a cash flow statement.
- Analyze financial statements using ratio analysis.
- Understand the elements, uses, and limitations of each financial statement and the relationships among the statements
- Understand how financial information, primarily that provided by the financial statements, can be used to analyze business operations and make economic decisions
- Identify the basic principles used in safeguarding assets and insuring the accuracy of accounting records
- Appreciate the role accounting software applications play in gathering, recording, reporting and interpreting financial accounting information
- Interpret and create standard financial statements.
- Perform financial statement analysis for the purposes of evaluating and forecasting in financial management.
- Evaluate a firm's working capital position.
- Estimate the components of cost of capital by applying time value of money principles.
- Perform net present value analysis for capital budgeting purposes.
- Evaluate risk in the capital budgeting process.
- Evaluate the cost of debt, preferred stock, and common stock as sources of capital.
- The Most important learning objective is how to do accounts in Computers.

Prerequisites

Some basic accounting knowledge and mathematics level is recommended.

Ability to understand the knowledge of cost estimation towards the business sceneries.

Ability to analyze market competency in the global village

To become a Master in accounting and financial management.

UNIT I

Introduction to Accounting: Principles, concepts and conventions, double entry system of accounting, introduction to basic books of accounts, Journal, ledger- Trial Balance - Preparation of Final accounts: Trading Account, Profit and Loss Account and Balance Sheet.

UNIT II

Financial Management - Meaning and scope, role of Financial Manager, Objectives of time value of money - Goals of Financial Management, Leverages: Operating, Financial Leverage and Combined Leverage Cost of Capital: Cost of Equity, Preference Shares, Bonds- Weighted Average Cost of Capital – Capital Gearing- Overcapitalization and Undercapitalization, Sources of Finance.

UNIT III

Tools and Techniques for Financial Statement Analysis: Ratio Analysis – Classification of Ratios –Short term solvency and long term solvency – Profitability ratios - Analysis and Interpretation of Financial Statements through ratios of Liquidity, Solvency and Profitability ratios. **Fund Flow Statement** - Meaning, Importance, Statement of changes in working capital and statement of Sources and application of funds. Cash flow Analysis: cash flow Statements: Preparation, Analysis and interpretation.

UNIT IV

Break-even Analysis: Concept of Break Even Point, Cost-Volume-Profit Analysis, Determination of Break Even Point, Margin of Safety and PV ratio, Impact of changes in Cost or selling price on BEP Practical applications of Break-even Analysis.

Budgeting: Budgeting–cash budget, sales budget – flexible Budgets and master budgets.

UNIT V

Capital Budgeting: Capital and its significance, Types of Capital, Estimation of Fixed and Working capital requirements, Methods and sources of raising capital.. Capital Budgeting: features of capital budgeting proposals, Methods of Capital Budgeting: Payback Method, Accounting Rate of Return (ARR) and Net Present Value Method (simple problems).

TEXT BOOKS:

1. Aryasri: Accounting And Financial Management,, TMH, 2009
2. Van Horne, James, C: Financial Management and Policy, Pearson, 2009

REFERENCE BOOKS:

1. Dr. G. Vidyanath G. Lakshmi Accounting and Financial Management.
2. Prasanna Chandra, Financial Management, TMH, 2009
3. S.N.Maheshwari, Financial Accounting, Sultan Chand, 2009.
4. Tulsian, Financial Accounting, S Chand, 2009.
5. Khan and Jain: Financial Management, TMH, 2009
6. Gokul Sinha: Financial Statement Analysis, PHI, 2009
7. Bhat Sundhindra: Financial Management, Excel:2009
8. Jawaharlal: Accounting for Management, Himalaya, 2009
9. Paresh Shah : Basic Financial Accounting for Management, Oxford 2010.

Objectives:

The IT Workshop for engineers is a training lab course spread over 54 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel and Power Point.

PC Hardware introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. **The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible.**

Internet & World Wide Web module introduces the different ways of hooking the PC on to the Internet from home and workplace and effective usage of the internet. Usage of web browsers, email, newsgroups and discussion forums would be covered. In addition, awareness of cyber hygiene, i.e., protecting the personal computer from getting infected with the viruses, worms and other cyber attacks would be introduced.

Productivity tools module would enable the students in crafting professional word documents, excel spread sheets and power point presentations using the Microsoft suite of office tools and LaTeX. **(Recommended to use Microsoft office 2007 in place of MS Office 2003)**

PC Hardware

Task 1 : Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2 : Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3 : Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4 : Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva.

Task 5: Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Task 6: Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Internet & World Wide Web

Task 1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no Internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

Task 2: Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

Task 3: Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

Task 4: Cyber Hygiene: Students would be exposed to the various threats on the Internet and would be asked to configure their computer to be safe on the Internet. They need to first install antivirus software, configure their personal firewall and windows update on their computer. Then they need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

Task 5: Develop your home page using HTML Consisting of your photo, name, address and education details as a table and your skill set as a list.

Productivity tools:

LaTeX and Word:

Word Orientation: The mentor needs to give an overview of LaTeX and Microsoft (MS) office 2007/ equivalent (FOSS) tool word: Importance of LaTeX and MS office 2007/ equivalent (FOSS) tool Word as word Processors, Details of the three tasks and features that would be covered in each, using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter.

Task 1: Using LaTeX and Word to create project certificate. Features to be covered:-Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both LaTeX and Word.

Task 2 : Creating project abstract Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

Task 3 : Creating a Newsletter : Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

Excel:

Excel Orientation: The mentor needs to tell the importance of MS office 2007/ equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the two tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

Task 1: Creating a Scheduler - Features to be covered: - Gridlines, Format Cells, Summation, auto fill, Formatting Text.

Task 2 : Calculating GPA - .Features to be covered:- Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP, Sorting, Conditional formatting.

LaTeX and MS/equivalent (FOSS) tool Power Point:

Task1: Students will be working on basic power point utilities and tools which help them create basic power point presentation. Topic covered during this week includes :- PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in both LaTeX and Power point. Students will be given model power point presentation which needs to be replicated (exactly how it's asked).

Task 2: Second week helps students in making their presentations interactive. Topic covered during this week includes: Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

Task 3: Concentrating on the in and out of Microsoft power point and presentations in LaTeX. Helps them learn best practices in designing and preparing power point presentation. Topic covered during this week includes: - Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), Inserting – Background, textures, Design Templates, Hidden slides.

REFERENCE BOOKS:

1. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
2. LaTeX Companion – Leslie Lamport, PHI/Pearson.
3. Introduction to Computers, Peter Norton, 6/e Mc Graw Hill.
4. Upgrading and Repairing, PC's 18th e, Scott Muller QUE, Pearson Education.
5. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dreamtech.
6. IT Essentials PC Hardware and Software Companion Guide Third Edition, David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. PC Hardware and A+Handbook – Kate J. Chase PHI (Microsoft).

MCA - I Year - I Sem

COMPUTER PROGRAMMING USING C LAB

Objectives:

- To write programs in C to solve the problems.
- To implement linear data structures such as lists, stacks, queues.
- To implement simple searching and sorting methods.

Outcomes:

Ability to write programs for different kinds of problems in C.
Work confidently in compilers like C and others.
Ability to identify appropriate data type or data structure to given problem.

Recommended Systems/Software Requirements:

- Intel based desktop PC.
- ANSI C Compiler with Supporting Editors.

Exercise-1

- Write a C program to find the sum of individual digits of a positive integer.
- A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
- Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

Exercise-2

- Write a C program to calculate the following Sum:
$$\text{Sum} = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! - x^{10}/10!$$
- Write a C program to find the roots of a quadratic equation.

Exercise-3

- The total distance travelled by vehicle in 't' seconds is given by distance $= ut + 1/2at^2$ where 'u' and 'a' are the initial velocity (m/sec.) and acceleration (m/sec²). Write C program to find the distance travelled at regular intervals of time given the values of 'u' and 'a'. The program should provide the flexibility to the user to select his own time intervals and repeat the calculations for different values of 'u' and 'a'.
- Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use Switch Statement).

Exercise-4

- Write C programs that use both recursive and non-recursive functions
 - To find the factorial of a given integer.
 - To find the GCD (greatest common divisor) of two given integers.

Exercise-5

- Write a C program to find the largest integer in a list of integers.
- Write a C program that uses functions to perform the following:
 - Addition of Two Matrices.
 - Multiplication of Two Matrices.

Exercise-6

- Write a C program that uses functions to perform the following operations:
 - To insert a sub-string in to a given main string from a given position.
 - To delete n Characters from a given position in a given string.
- Write a C program to determine if the given string is a palindrome or not.

Exercise-7

- Write a C program that displays the position or index in the string S where the string T begins, or - 1 if S doesn't contain T.
- Write a C program to count the lines, words and characters in a given text.

Exercise-8

- Write a C program to generate Pascal's triangle.
- Write a C program to construct a pyramid of numbers.

Exercise-9

Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression:

$$1+x+x^2+x^3+\dots+x^n$$

For example: if n is 3 and x is 5, then the program computes 1+5+25+125.

Print x, n, the sum

Perform error checking. For example, the formula does not make sense for negative exponents – if n is less than 0.

Have your program print an error message if n<0, then go back and read in the next pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too.

Exercise-10

a) 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the 2's complement of a binary number.

b) Write a C program to convert a Roman numeral to its decimal equivalent.

Exercise-11

Write a C program that uses functions to perform the following operations:

i) Reading a complex number.

ii) Writing a complex number.

iii) Addition of two complex numbers.

iv) Multiplication of two complex numbers.

(Note: represent complex number using a structure.)

Exercise-12

a) Write a C program which copies one file to another.

b) Write a C program to reverse the first n characters in a file.

(Note: The file name and n are specified on the command line.)

Exercise-13

a) Write a C program to display the contents of a file.

b) Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

Exercise-14

a) Write a C program that uses non recursive function to search for a Key value in a given list of integers using Linear search.

b) Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using Binary search.

Exercise-15

Write a C program to implement the Lagrange interpolation.

Exercise-16

Write C program to implement the Newton- Gregory forward interpolation.

Exercise-17

Write a C program to implement Trapezoidal method.

Exercise-18

Write a C program to implement Simpson method.

TEXT BOOKS:

1. C programming and Data Structures, P. Padmanabham, Third Edition, BS Publications.
2. Computer Programming in C, V. Rajaraman, PHI.
3. C Programming, E.Balagurusamy, 3rd edition, TMH.
4. C Programming, M.V.S.S.N.Prasad, ACME Learning Pvt. Ltd.
5. C and Data Structures, N.B.Venkateswarlu and E.V.Prasad, S.Chand.
6. Mastering C, K.R. Venugopal and S.R. Prasad, TMH Publications.

Objectives:

- To write assembly language programs (8086 processor) for solving problems.

List of Sample Problems:

Write assembly language programs for the following using MASAM:

1. Write assembly language programs to evaluate the expressions:

i) $a = b + c - d * e$

ii) $z = x * y + w - v + u / k$

- Considering 8-bit, 16 bit and 32 bit binary numbers as b, c, d, e.
- Considering 2 digit, 4digit and 8 digit BCD numbers.

Take the input in consecutive memory locations and results also.

Display the results by using "int xx" of 8086. Validate program for the boundary conditions.

2. Write an ALP of 8086 to add two exponential numbers which are in IEEE 754 notation. Display the results by using "int xx" of 8086. Validate program for the boundary Conditions.

3. Write an ALP of 8086 to take N numbers as input. And do the following operations on them.

- Arrange in ascending and descending order.
- Find max and minimum
- Find average

Considering 8-bit, 16 bit binary numbers and 2 digit, 4digit and 8 digit BCD numbers.

Display the results by using "int xx" of 8086. Validate program for the boundary conditions.

4. Write an ALP of 8086 to take a string of as input (in 'C' format)and do the Following Operations on it.

- Find the length .
- Find it is Palindrome or not.
- Find whether given string substring or not.
- Reverse a string.
- Concatenate by taking another sting.

Display the results by using "int xx" of 8086.

5. Write the ALP to implement the above operations as procedures and call from the main procedure.

6. Write an ALP of 8086 to find the factorial of a given number as a Procedure and call from the main program which display the result.

7. Write an assembly language program to encrypt digits as shown below:

Input digit: 0 1 2 3 4 5 6 7 8 9

Encrypted digit: 4 6 9 5 0 3 1 8 7 2

Your program should accept a string consisting of digits. The encrypted String should be displayed using "int xx" of 8086.

8. Write a procedure to locate a character in a given string. The procedure receives a pointer to a string and character to be located. When the first occurrence of the character is located, its position is returned to main. If no match is found, a negative value is returned. The main procedure requests a character string and a character to be located and displays the result.

9. Write an assembly language program to read a string of characters from the user and that prints the vowel count . Display the results by using "int xx" of 8086.

ex. Input: Advanced Programming in UNIX

Out put:

Vowel	count
a or A	3
e or E	1
i or I	3
o or O	1
u or U	1

10. A computer uses RAM chips of 1024 X 1 capacity.

- How many chips are needed, and how should their address lines be connected to provide a memory capacity of 1024 bytes?
- How many chips are needed to provide a memory capacity of 16K bytes?

11. A computer employs RAM chips of 256X8 and ROM chips of 1024 X 8. The computer needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the

- address bus are assigned 00 for RAM, 01 for ROM, 10 for interface registers.
- a. How many RAM and ROM chips are needed?
 - b. Draw a memory-address map for the system.
 - c. Give the address range in hexadecimal for RAM, ROM and interface.
12. Obtain the complement function for the match logic of one word in an associative memory. Draw the logic diagram for it and compare with the actual match logic diagram.
 13. A two-way set associative cache memory uses blocks of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128K X 32.
 - a. Formulate all pertinent information required to construct the cache memory.
 - b. What is the size of the cache memory?
 14. A digital computer has a memory unit of 64K X 16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.
 - a. How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.
 - b. How many bits are there in the tag, index, block, and word fields of the address format?
 - c. How many blocks can the cache accommodate?
 15. An address space is specified by 24 bits and the corresponding memory space by 16 bits.
 - a. How many words are there in the address space?
 - b. How many words are there in the memory space?
 - c. If a page consists of 2K words, how many pages and blocks are there in the system.
 16. A virtual memory has a page size of 1K words. There are eight pages and four blocks. The associative memory page table contains the following entries. Make a list of all virtual addresses(in decimal) that will cause a page fault.

Page	Block
0	3
1	1
4	2
6	0

TEXT BOOKS:

1. IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI.
2. Introduction to Assembly Language Programming, Sivarama P.Dandamudi, Springer Int. Edition, 2003.
3. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Application, 4th edition, W.A.Triebel, A.Singh, N.K.Srinath, Pearson Education.