## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - I Year - I Sem**

## MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

**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, Bernand 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 Chakraborthy and B K Sarkar, Oxford, 2011.

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - I Year - I Sem**

### COMPUTER ORGANIZATION

**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.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**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.Borror, 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.

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - I Year - I Sem**

## 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: B**udgeting–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.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - I Sem**

**IT WORKSHOP**

**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).

### JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**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-I**
**a)** Write a C program to find the sum of individual digits of a positive integer.
**b)** 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.
**c)** 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**
**a)** Write a C program to calculate the following Sum:
$$Sum = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! - x^{10}/10!$$
**b)** Write a C program to find the roots of a quadratic equation.

**Exercise-3**
**a)** 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/sec$^2$). 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'.
**b)** 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**
**a)** Write C programs that use both recursive and non-recursive functions
    i) To find the factorial of a given integer.
    ii) To find the GCD (greatest common divisor) of two given integers.

**Exercise-5**
**a)** Write a C program to find the largest integer in a list of integers.
**b)** Write a C program that uses functions to perform the following:
    i) Addition of Two Matrices.
    ii) Multiplication of Two Matrices.

**Exercise-6**
**a)** Write a C program that uses functions to perform the following operations:
    i) To insert a sub-string in to a given main string from a given position.
    ii) To delete n Characters from a given position in a given string.
**b)** Write a C program to determine if the given string is a palindrome or not.

**Exercise-7**
**a)** 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.
**b)** Write a C program to count the lines, words and characters in a given text.

**Exercise-8**
**a)** Write a C program to generate Pascal's triangle.
**b)** 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:
$1+x+x^2+x^3+\ldots\ldots\ldots+x^n$
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.

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - I Year - I Sem**

### COMPUTER ORGANIZATION LAB

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

  a. Considering 8-bit, 16 bit and 32 bit binary numbers as b, c, d, e.

  b. 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.

a) Arrange in ascending and descending order.

b) Find max and minimum

c) 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.

a) Find the length .          b) Find it is Palindrome or not.

c) Find whether given string substring or not.

d) Reverse a string.          e) 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.

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

b) 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.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MASTER OF COMPUTER APPLICATIONS
COURSE STRUCTURE AND SYLLABUS**

**I Year – II Semester**

| Category | Course Title | Int. marks | Ext. marks | L | P | C |
|---|---|---|---|---|---|---|
| Core Course I | Data Structures and Algorithms | 25 | 75 | 4 | -- | 4 |
| Core Course II | Object Oriented programming using C++ | 25 | 75 | 4 | -- | 4 |
| Core Course III | Scripting Languages | 25 | 75 | 4 | -- | 4 |
| Core Course IV | Operations Research | 25 | 75 | 4 | -- | 4 |
| Core Course V | Software Engineering | 25 | 75 | 4 | -- | 4 |
| Laboratory I | Data Structures Lab Using C++ | 25 | 75 | -- | 4 | 2 |
| Laboratory II | Software Engineering Lab | 25 | 75 | -- | 4 | 2 |
| Laboratory III | English Language Communication Skills Lab | 25 | 75 | -- | 4 | 2 |
| **Total Credits** | | | | **20** | **6** | **26** |

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - I Year - II Sem.**

## DATA STRUCTURES AND ALGORITHMS

**Objectives:**
- To understand the basic concepts such as Abstract Data Types, Linear and Non Linear Data structures.
- To understand the notations used to analyze the Performance of algorithms.
- To understand the behavior of data structures such as stacks, queues, trees, hash tables, search trees, Graphs and their representations.
- To choose the appropriate data structure for a specified application.
- To understand and analyze various searching and sorting algorithms.
- To write programs in C to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables, search trees.

**Outcomes:**
- Learn how to use data structure concepts for realistic problems.
- Ability to identify appropriate data structure for solving computing problems in respective language.
- Ability to solve problems independently and think critically.
- Be able to analyze algorithms and improve the efficiency of algorithms.
- Apply different designing methods for development of algorithms to realistic problems, such as divide and conquer, greedy and etc.
- Ability to understand and estimate the performance of algorithm

## UNIT I
Basic concepts-Data types, Abstract Data Types, Data structures, Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Analysis-Big O, Omega and Theta notations.
Linear data structures- Linear Lists, Sequential and Linked allocation The list ADT, array and linked Implementations, Singly Linked Lists-Operations-Insertion, Deletion, Doubly Linked
Lists-
Operations- Insertion, Deletion, Stack ADT, definitions, operations, array and linked implementations, applications-infix to postfix conversion, recursion implementation, Queue ADT, definitions and operations ,array and linked Implementations.

## UNIT II
Non Linear data structures- Trees – Basic Terminology, Binary tree ADT, array and linked representations, traversals, threaded binary trees, Disjoint Sets, Union and Find algorithms, Priority Queues-Definition, ADT, Realizing a Priority Queue using Heap.
Graphs – Introduction, Basic Terminology, Graph Representations- Adjacency matrix, Adjacency lists, Adjacency multi lists, Graph traversals- DFS and BFS.

## UNIT III
Searching- Linear Search, Binary Search, Hashing-Introduction, hash tables, hash functions, collision resolution methods, Comparison of Searching methods.
Sorting- Bubble Sort, Insertion Sort, Selection Sort, Heap Sort, Radix Sort. Algorithm Design method-Divide and Conquer method-applications- Quick sort, Merge sort, Comparison of Sorting methods, Model for External Sorting.

## UNIT IV
Search Trees-Binary Search Trees, Definition, ADT, Implementation, Operations- Searching, Insertion and Deletion, AVL Trees, Definition, Operations – Insertion and Searching,
B-Trees, Definition, B-Tree of order m, operations- insertion and deletion, Introduction to Red-Black and Splay Trees(Elementary treatment), Comparison of Search Trees.

**UNIT V**
Algorithm Design methods-Greedy method-applications-Kruskal's Algorithm for Minimum cost Spanning trees, Job Sequencing with deadlines, Single Source Shortest path problem, Dynamic Programming method-applications-Ordering matrix multiplications, Optimal Binary Search Trees, APSP problem.
**Text Processing -** Pattern matching algorithm- The Knuth-Morris-Pratt algorithm, Tries- Standard Tries, Compressed Tries, Suffix tries.

**TEXT BOOKS:**
1. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, 3$^{rd}$ edition, Pearson Education. Ltd.,
2. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and  D.Mount, Wiley Student edition, seventh edition, John Wiley and Sons.
3. Data structures, Algorithms and Applications in C++, S.Sahani, Universities Press.

**REFERENCE BOOKS:**
1. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Cengage Learning.
2. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
3. Data structures, algorithms and OOP, G.L.Heileman, TMH edition.
4. Data Structures using C++, D.S. Malik, Cengage Learning, India Edition.
5. Data structures with C++ Using STL, 2$^{nd}$ edition, W.H.Ford and W.R.Topp, Pearson/PH.I
6. Mastering Algorithms with C,K.Loudon, O'Reilly, SPD pvt. Ltd.
7. Advanced Data structures & Algorithms in C++, V.V.Muniswamy, Jaico Publishing House.
8. An Introduction to Data structures and Algorithms, J.A.Storer, Springer.

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - I Year - II Sem.**

### OBJECT ORIENTED PROGRAMMING USING C++

**UNIT I**

Different paradigms for problem solving, need for OOP paradigm, classes and instances, fundamental characteristics of OOP (Alan key), differences between OOP and Procedure Oriented Programming.
C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statements- if, switch, while, for, do, break, continue, goto statements.

**UNIT II**

C++ Functions-Scope of variables, Parameter passing methods, Default arguments, inline functions, Recursive functions, Pointers to functions.
C++ Classes And Data Abstraction: Class definition, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction, ADT and information hiding.

**UNIT III**

Dynamic memory allocation and deallocation operators-new and delete, Dynamic creation and destruction of objects, Preprocessor directives, command line arguments,name spaces.
Polymorphism: Function overloading, Operator overloading, Generic programming-necessity of templates, Function templates and class templates.

**UNIT IV**

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.
Virtual Functions And Run Time Polymorphism: Overriding, Static and Dynamic bindings, Base and Derived class virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes,Virtual destructors.

**UNIT V**

C++ I/O: I/O using C functions, C++ Stream classes hierarchy, Stream I/O, File streams and String streams,File Operations, Overloading << and >> operators, Error handling during file operations, Formatted I/O.
Exception Handling: Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

**TEXT BOOKS:**
1.      C++, The Complete Reference, 4th Edition, Herbert Schildt, TMH.
2.      Object Oriented Programming in C++, 4th Edition, R.Lafore, Pearson Education
**REFERENCE BOOKS:**
1. An Introduction to OOP, 3rd Edition, T. Budd, Pearson Education,2008.
2. Programming Principles and Practice Using C++, B.Stroutstrup, Pearson Education.
3. Problem solving with C++, 6th Edition, Walter Savitch, Pearson Education,2007.
4. Mastering C++, K.R.Venu Gopal, Raj Kumar and T.Ravi Shankar, TMH.
5. OOP in C++, 3rd Edition, T.Gaddis, J.Walters and G.Muganda, Wiley DreamTech Press.
6. An Introduction to OOP in C++ with applications in Computer Graphics, 2nd Edition, G.M.Seed, Springer.
7. Programming with ANSI C++, B.Trivedi,Oxford Press.
8. Programming in C++,M.T.Somasekara,PHI.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - II Sem.**

## SCRIPTING LANGUAGES

**Objectives:**
The course demonstrates an in depth understanding of the tools and the scripting languages necessary for design and development of applications dealing with Bio-information/ Bio-data. The instructor is advised to discuss examples in the context of Bio-data/ Bio-information application development.

**UNIT  I**
**Introduction to PERL and Scripting** Scripts and Programs, Origin of Scripting , Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

**UNIT  II**
**Advanced perl:** Finer points of looping, pack and unpack, file system, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.
**PHP Basics**
PHP Basics- Features, Embedding PHP Code in your Web pages, Outputting the data to the browser, Data types, Variables, Constants, expressions, string interpolation, control structures, Function, Creating a Function, Function Libraries, Arrays, strings and Regular Expressions.

**UNIT  III**
**Advanced PHP Programming:** PHP and Web Forms, Files, PHP Authentication and Methodologies - Hard Coded, File Based, Database Based, IP Based, Login Administration, Uploading Files with PHP, Sending Email using PHP, PHP Encryption Functions, the Mcrypt package, Building Web sites for the World.

**UNIT  IV**
**TCL :** TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.
Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding , Perl-Tk.

**UNIT  V**
**Python:** Introduction to Python language, python-syntax, statements, functions, Built-in-functions and Methods, Modules in python, Exception Handling.
Integrated Web Applications in Python – Building Small, Efficient Python Web Systems, Web Application Framework.

**TEXT BOOKS:**
1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Python Web Programming, Steve Holden and David Beazley, New Riders Publications.
3. Beginning PHP and MySQL, 3$^{rd}$ Edition, Jason Gilmore, Apress Publications (Dream tech.).

**REFERENCE BOOKS:**
 1.      Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and PHP, J.Lee and B.Ware (Addison Wesley) Pearson Education.
 2.      Programming Python, M.Lutz, SPD.
 3.      PHP 6 Fast and Easy Web Development, Julie Meloni and Matt Telles, Cengage Learning Publications.

4.      PHP 5.1, I.Bayross and S.Shah, The X Team, SPD.
5.      Core Python Programming, Chun, Pearson Education.
6.      Guide to Programming with Python, M.Dawson, Cengage Learning.
7.      Perl by Example, E.Quigley, Pearson Education.
8.      Programming Perl, Larry Wall, T.Christiansen and J.Orwant, O'Reilly, SPD.
9.      Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
10.PHP and MySQL by Example, E.Quigley, Prentice Hall (Pearson).
11.Perl Power, J.P.Flynt, Cengage Learning.
12.      PHP Programming solutions, V.Vaswani, TMH.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - II Sem.**

## OPERATIONS RESEARCH

**Objectives:**
- To introduce the methods of Operations Research.
- Emphasize the mathematical procedures of non linear programming search techniques.
- Introduce advanced topics such as Probabilistic models and dynamic programming.

**Prerequisites:** Probability and Statistics

**Outcomes:**
- Ability to understand the optimization methods and also learn where to use in computer programming.
- Ability to apply the OR methods in different domains like operating system, algorithms and etc.

### UNIT I
Introduction to Operations Research: Basics definition, scope, objectives, phases, models and limitations of Operations Research. Linear Programming Problem – Formulation of LPP, Graphical solution of LPP. Simplex Method, Artificial variables, big-M method, two-phase method, degeneracy and unbound solutions.

### UNIT II
Transportation Problem:  Formulation, solution, unbalanced Transportation problem. Finding basic feasible solutions – Northwest corner rule, least cost method and Vogel's approximation method. Optimality test: the stepping stone method and MODI method.
Assignment model: Formulation. Hungarian method for optimal solution. Solving unbalanced problem. Traveling salesman problem as assignment problem.

### UNIT III
Sequencing models: Solution of Sequencing Problem – Processing n Jobs through 2 Machines – Processing n Jobs through 3 Machines – Processing 2 Jobs through m machines – Processing n Jobs through m Machines.
Replacement Models: Replacement of Items that Deteriorate whose maintenance costs increase with time without change in the money value. Replacement of items that fail suddenly: individual replacement policy, group replacement policy.

### UNIT IV
Dynamic programming: Characteristics of dynamic programming. Dynamic programming approach for Priority Management employment smoothening, Stage Coach/Shortest Path and Reliability problems.
Games Theory: Competitive games, rectangular game, saddle point, minimax (maximin) method of optimal strategies, value of the game. Solution of games with saddle points, dominance principle. Rectangular games without saddle point – mixed strategy for 2 X 2 games.

### UNIT V
Inventory models: Inventory costs. Models with deterministic demand – model (a) demand rate uniform and production rate infinite, model (b) demand rate non-uniform and production rate infinite, model (c) demand rate uniform and production rate finite.
Queuing Theory: Essential Features of a queuing system. Performance measures of a queuing system. Model 1: {(M/M/1) : (∞/FCFS)} Single server, Unlimited Queue model. Model 2: {(M/M/1) : (∞/SIRO)} Single server, Unlimited Queue model. Model III: {(M/M/1): (N/FCFS)} Single server, Finite Queue model.

### TEXT BOOKS:
1. J K Sharma. "Operations Research Theory & Applications 4e", Macmillan India Ltd.
2. P. K. Gupta and D. S. Hira, "Operations Research", S. Chand & co., 2007.

**REFERENCE BOOKS:**

1. Pradeep Prabhakar Pai, Operations Research – principles and Practice, Oxford University Press, 2012.
2. A.M. Natarajan, P. Balasubramani, A. Tamilarasi, "Operations Research", Pearson Education.
3. P Sankara Iyer,"Operations Research", Tata McGraw-Hill, 2008.
4. N.V.S. Raju, "Operations Research", HI-TECH, 2002.
5. Col. D. S. Cheema, "Operations Research", Laxmi Publications Ltd., 2005.
6. F.S. Hillier, G.J. Lieberman, "Introduction to Operations Research – 8ed", TMH.
7. H.S. Kasana & K.D. Kumar, "Introductory Operations Research – Theory and applications", Springer, 2003, rp2005.
8. Billy E. Gillett, "Introduction to Operations Research – A Computer-Oriented Algorithmic Approach", Tata McGraw-Hill, 1979, rp2004.
9. A.B.Rao, Operations Research, Jaico .
10. Ravindran, Phillips, Solberg, Operations Research, 2nd edition, Wiley India.
11. W.L.Winston, Operations Research, 4th edition, Cengage Learning.
12. R. Panneerselvam, "Operations Research", PHI-2e, 2006, rp2008.
13. ANITHA H S, "Operations Research", EXEL books, 2011.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - II Sem.**

## SOFTWARE ENGINEERING

**Objectives:**
- Understanding of software process models such as waterfall and evolutionary models.
- Understanding of software requirements and SRS document.
- Understanding of different software architectural styles.
- Understanding of software testing approaches such as unit testing and integration testing.
- Understanding on quality control and how to ensure good quality software.

**Outcomes:**
- Ability to identify the minimum requirements for the development of application.
- Ability to develop, maintain, efficient, reliable and cost effective software solutions.
- Ability to think critically and evaluate assumptions and arguments.

### UNIT I
Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, legacy software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, specialized process models, The Unified process.

### UNIT II
Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

### UNIT III
Design Engineering: Design process and Design quality, Design concepts, the design model, pattern based software design.

Creating an architectural design: software architecture, Data design, Architectural styles and patterns, Architectural Design, assessing alternative architectural designs, mapping data flow into a software architecture.

Modeling component-level design: Designing class-based components, conducting component-level design, Object constraint language, designing conventional components.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

### UNIT IV
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Frame work for Product metrics, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

### UNIT V
Risk management: Reactive Vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

**TEXT BOOKS:**
1. Software Engineering A practitioner's Approach, Roger S Pressman, sixth edition, McGraw Hill International Edition.
2.　　Software Engineering, Ian Sommerville, seventh edition, Pearson education.

**REFERENCE BOOKS:**
1.　　Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
2.　　Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008.
3.　　Fundamentals of Software Engineering, Rajib Mall, PHI, 2005.
4.　　Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5.　　Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6.　　Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition , 2006.
7.　　Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8.　　Software Engineering Principles and Practice, Hans Van Vliet, $3^{rd}$ edition, John Wiley &Sons Ltd.
9.　　Software Engineering 3: Domains, Requirements,and Software Design, D.Bjorner, Springer International Edition.
10.　Introduction to Software Engineering, R.J.Leach, CRC Press.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - II Sem.**

**DATA STRUCTURES LAB USING C++**

**List of Sample Problems:**

1.Write a C++ program to perform the following operations :
   a) Create a Singly linked list of elements.
   b) Search for a given element in the above list.
   c) Delete an element from the above Singly linked list.
   d) Display the contents of the list.

2. Write a C++ program to perform the following operations:
   a) Create a doubly linked list of elements.
   b) Delete an element from the above doubly linked list
   c) Display the contents of the list.

3.Write C++ programs to implement the following using an array.
a)              Stack ADT b) Queue ADT

4.Write C++ programs to implement the following using a singly linked list.
a)              Stack ADT b) Queue ADT

5. Write a C++ program to convert i) a given infix expression into postfix form using stack
     and ii) evaluate the Postfix expression using stack.

6.Write C++ programs to implement the deque (double ended queue) ADT using
        a)Singly linked list  b) Doubly linked list c) an array.

7. Write C++ programs that use recursive functions to traverse the given
   binary tree in
               a) Preorder b) Inorder and c) Postorder

8.Write a C++ program to perform the following operations:
   a) Construct a binary search tree of elements.
   b) Search for a key element in the above binary search tree.
   c) Delete an element from the above binary search tree.

9.Write C++ programs for implementing the following Searching methods:
               a) Linear Search b) Binary Search

10.Write C++ programs for implementing the following sorting methods:
               a) Bubble Sort b) Selection Sort c) Insertion Sort

11.Write C++ programs for implementing the following sorting methods:
 a)Merge sort b)Quick sort c) Heap sort d)Radix sort.

12.Write a C++ program to perform the following operation:
a)              Insertion into a B-tree

13.Write a C++ program to implement all the functions of a dictionary (ADT)
    using hashing.

14. Write a C++ program for implementing Knuth-Morris- Pratt pattern matching algorithm.

15. Write C++ programs that use non-recursive functions to traverse the given binary tree in
a)                    Preorder b) inorder and c) postorder.

16. Write C++ programs for the depth first and breadth first traversals of a graph.


 **(Note: You may use STL(Standard Template Libray) in writing the above programs)**

**TEXT BOOKS :**

1. Data Structures A Pseudocode Approach with C++, India Edition, R.F.Gilberg
   and B.A.Forouzan,Cengage Learning.
2. Data structures with C++,J.R.Hubbard,Schaum's Outlines,TMH.
3. Data Structures and STL, W.J.Collins,Mc Graw Hill,International edition.
4. Data structures and Algorithms with OODesign patterns in C++,B.R.Priess,John Wiley& sons.
5. Advanced Data structures & Algorithms in C++,V.V.Muniswamy,Jaico Publishing House.
6. C++ for Programmers,P.J.Deitel and H.M.Deitel,Pearson/C++: How to Program,Deitel
   and Dietel,PHI.
7. Data structures via C++,A.M.Berman, Oxford University Press.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - II Sem.**

**SOFTWARE ENGINEERING LAB**

**Objectives:**

• To have hands on experience on dealing with a software project development activity by using various software Engineering principles and methods.
• At the end of the course the student shall have clear idea about Software Engineering and will be in a position to do all the phases of a software project development individually for small projects.

**LIST OF EXPERIMENTS**

**Do the following 8 exercises for any two projects given in sample projects or any other projects:**

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
4. Preparation of Software Configuration Management and Risk Management related documents.
5. Study and usage of any Design phase CASE tool
6. Performing the Design by using any Design phase CASE tools.
7. Develop test cases for unit testing and integration testing
8. Develop test cases for various white box and black box testing techniques.

**Sample Projects:**
1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - I Year - II Sem.**

**ENGLISH LANGUAGE COMMUNICATION SKILLS LAB**

The **Language Lab** focuses on the production and practice of sounds of language and familiarizes the students with the use of English in everyday situations and contexts.

**Objectives:**
&#9758;    To facilitate computer-aided multi-media instruction enabling individualized and independent language learning
&#9758;    To sensitize the students to the nuances of English speech sounds, word accent, intonation and rhythm
&#9758;    To bring about a consistent accent and intelligibility in their pronunciation of English by providing an opportunity for practice in speaking
&#9758;    To improve the fluency in spoken English and neutralize mother tongue influence
&#9758;    To train students to use language appropriately for interviews, group discussion and public speaking

**Syllabus: English Language Communication Skills Lab shall have two parts:**
**a.      Computer Assisted Language Learning (CALL) Lab**
**b.      Interactive Communication Skills (ICS) Lab**
The following course content is prescribed for the **English Language Communication Skills Lab**

**Exercise – I**
**CALL Lab**: Introduction to Phonetics – Speech Sounds – Vowels and Consonants
**ICS Lab**: Ice-Breaking activity and JAM session
Articles, Prepositions, Word formation- Prefixes & Suffixes, Synonyms & Antonyms

**Exercise – II**
**CALL Lab**: Structure of Syllables - Past Tense Marker and Plural Marker – Weak Forms and Strong Forms - Consonant Clusters.
**ICS Lab**: Situational Dialogues – Role-Play- Expressions in Various Situations – Self-introduction and Introducing Others – Greetings – Apologies – Requests – Social and Professional Etiquette - Telephone Etiquette.
Concord (Subject in agreement with verb) and Words often misspelt- confused/misused

**Exercise - III**
**CALL Lab:** Minimal Pairs- Word accent and Stress Shifts- Listening Comprehension.
**ICS Lab**: Descriptions- Narrations- Giving Directions and guidelines.
Sequence of Tenses, Question Tags and One word substitutes.

**Exercise – IV**
**CALL Lab**: Intonation and Common errors in Pronunciation.
**ICS Lab**: Extempore- Public Speaking
Active and Passive Voice, –Common Errors in English, Idioms and Phrases

**Exercise – V**
**CALL Lab:** Neutralization of Mother Tongue Influence and Conversation Practice
**ICS Lab**: Information Transfer- Oral Presentation Skills
Reading Comprehension and Job Application with Resume preparation.
**Minimum Requirement of infra structural facilities for ELCS Lab:**
**1.      Computer Assisted Language Learning (CALL) Lab:**
**The Computer aided Language Lab** for 40 students with 40 systems, one master console, LAN facility and English language software for self- study by learners.

**System Requirement (Hardware component):**
*Computer network with LAN with minimum 60 multimedia systems with the following specifications:*
i) P – IV Processor
a)          Speed – 2.8 GHZ
b)          RAM – 512 MB Minimum
c)          Hard Disk – 80 GB
ii)         Headphones of High quality

**2.          Interactive Communication Skills (ICS) Lab :**
**The Interactive Communication Skills Lab:** A Spacious room with movable chairs and audio-visual aids with a Public Address System, a T. V., a digital stereo –audio & video system and camcorder etc.

**Prescribed Lab Manual:** A Manual entitled "***English Language Communication Skills (ELCS) Lab Manual- cum- Work Book***", published by Cengage Learning India Pvt. Ltd, New Delhi. 2013.

**Suggested Software:**
❖          **Cambridge Advanced Learners' English Dictionary with CD.**
❖          **Grammar Made Easy by Darling Kindersley**
❖          **Punctuation  Made Easy by Darling Kindersley**
❖          Clarity Pronunciation Power – Part I
❖          Clarity Pronunciation Power – part II
❖          **Oxford Advanced Learner's Compass, 7th Edition**
❖          ***DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.***
❖          Lingua TOEFL CBT Insider, by Dreamtech
❖          TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
❖          **English in Mind (Series 1-4), Herbert Puchta and Jeff Stranks with Meredith Levy, Cambridge**
❖          English Pronunciation in Use, Cambridge University Press
❖          Technical Communication, OUP
❖          Communication Skills, OUP

**SUGGESTED READING:**
1.          Rama Krishna Rao, A. *et al. English Language Communication Skills – A Reader cum Lab Manual Course Content and Practice.* Chennai: Anuradha Publishers
2.          Suresh Kumar, E. & Sreehari, P. 2009. *A Handbook for English Language Laboratories.* New Delhi: Foundation
3.          *Speaking English Effectively* 2nd Edition by Krishna Mohan and N. P. Singh, 2011. Macmillan Publishers India Ltd. Delhi.
4.          Sasi Kumar, V & Dhamija, P.V. *How to Prepare for Group Discussion and Interviews.* Tata McGraw Hill
5.          Hancock, M. 2009. *English Pronunciation in Use. Intermediate.* Cambridge: CUP
6.          Spoken English: A Manual of Speech and Phonetics by R. K. Bansal & J. B. Harrison. 2013. Orient Blackswan. Hyderabad.
7.          Hewings, M. 2009. *English Pronunciation in Use. Advanced.* Cambridge: CUP
8.          Marks, J. 2009. *English Pronunciation in Use. Elementary.* Cambridge: CUP
9.          Nambiar, K.C. 2011. *Speaking Accurately. A Course in International Communication.* New Delhi : Foundation
10.         Soundararaj, Francis. 2012. *Basics of Communication in English. New Delhi: Macmillan*
11.         **Spoken English** (CIEFL) in 3 volumes with 6 cassettes, OUP.
12.         **English Pronouncing Dictionary** Daniel Jones Current Edition with CD.
13.         **A text book of English Phonetics for Indian Students** by T.Balasubramanian (Macmillan)

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - I Sem.**

**DATA BASE MANAGEMENT SYSTEMS**

**Objectives:**
The objective of the course is to learn the foundation of Database Technologies. In this course, student will learn to design a database, normalize it by eliminating the redundancies, Querying, Improve the database performance by Indexing, Hashing and finally optimize the queries and Transaction Management.

**Outcomes:** Demonstrate the basic elements of a relational database management system.
- Ability to identify the data models for relevant problems.
- Ability to design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respective data.
- Apply normalization and professional attitudes for the development of application software's.

**UNIT I**
Database System Applications, Purpose of Database Systems, View of Data – Data Abstraction, Instances and Schemas, Data Models – the ER Model, Relational Model, Other Models – Database Languages – DDL, DML, Database Access from Applications Programs, Transaction Management, Data Storage and Querying, Database Architecture, Database Users and Administrators, History of Data base Systems. Introduction to Data base design, ER diagrams, Beyond ER Design, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises. Relational Model: Introduction to the Relational Model – Integrity Constraints over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design, Introduction to Views – Destroying /altering Tables and Views.

**UNIT II**
Relational Algebra and Calculus: Relational Algebra – Selection and Projection, Set operations, Renaming, Joins, Division, Examples of Algebra Queries, Relational calculus – Tuple relational Calculus – Domain relational calculus – Expressive Power of Algebra and calculus.
Form of Basic SQL Query – Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set – Comparison Operators, Aggregate Operators, NULL values – Comparison using Null values – Logical connectives – AND, OR and NOT – Impact on SQL Constructs, Outer Joins, Disallowing NULL values, Complex Integrity Constraints in SQL Triggers and Active Data bases.

**UNIT III**
Introduction to Schema Refinement – Problems Caused by redundancy, Decompositions – Problem related to decomposition, Functional Dependencies - Reasoning about FDS, Normal Forms – FIRST, SECOND, THIRD Normal forms – BCNF –Properties of Decompositions- Loss less- join Decomposition, Dependency preserving Decomposition, Schema Refinement in Data base Design – Multi valued Dependencies – FOURTH Normal Form, Join Dependencies, FIFTH Normal form, Inclusion Dependencies.

**UNIT IV**
Overview of Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control, Deadlocks – Performance of Locking – Transaction Support in SQL.
Concurrency Control: Serializability, and recoverability – Introduction to Lock Management – Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques – Concurrency Control without Locking.
Crash recovery: Introduction to Crash recovery, Introduction to ARIES, the Log, Other Recovery related Structures, the Write-Ahead Log Protocol, Check pointing, recovering from a System Crash, Media recovery.

**UNIT V**

Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing – Clustered Indexes, Primary and Secondary Indexes, Index data Structures – Hash Based Indexing, Tree based Indexing, Comparison of File Organizations.

Storing data: Disks and Files -The Memory Hierarchy – Redundant Arrays of Independent Disks.

Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM). B+ Trees: A Dynamic Index Structure, Search, Insert, and Delete.

Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendible Vs. Linear Hashing.

**TEXT BOOKS:**
1. Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, TMH, 3rd Edition, 2003.
2. Data base System Concepts, A.Silberschatz, H.F. Korth, S.Sudarshan, McGraw hill, VI edition, 2006.
3. Fundamentals of Database Systems 5th edition, Ramez Elmasri, Shamkant B.Navathe, Pearson Education, 2008.

**REFERENCE BOOKS:**
1. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, PHI.
2. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning, 2008.
3. Database Systems, A Practical approach to Design Implementation and Management Fourth edition, Thomas Connolly, Carolyn Begg, Pearson education.
4. Database-Principles, Programming, and Performance, P.O'Neil, E. O'Neil 2nd ed., ELSEVIER.
5. Fundamentals of Relational Database Management Systems, S.Sumathi, S.Esakkirajan, Springer.
6. Introduction to Database Management, M.L.Gillenson and others, Wiley Student Edition.
7. Database Development and Management, Lee Chao, Auerbach publications, Taylor & Francis Group.
8. Introduction to Database Systems, C.J.Date, Pearson Education.
9. Database Management Systems, G.K.Gupta, TMH.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - II Year - I Sem.**

## COMPUTER NETWORKS

**Objectives:**
- To introduce the fundamentals of various types of computer networks.
- To demonstrate the TCP/IP and OSI models with merits and demerits.
- To explore the various layers of OSI Model.
- To introduce UDP and TCP Models.

**Outcomes:**
- After this course student could understand and explore the basics of Computer Networks and Various Protocols. She/he will be in a position to understand the World Wide Web concepts.
- After this course he/she will be in a position to administrate a network and flow of information.

**UNIT I**
Overview of the Internet: Protocol, Layering Scenario, TCP/IP Protocol Suite: The OSI Model,
Internet history standards and administration; Comparison of the OSI and TCP/IP reference model
**Data Link Layer** – design issues, CRC Codes, Elementary Data link Layer protocols, sliding window protocol

**UNIT II**
Multiple Access Protocols –ALOHA, CSMA, Collision free protocols, Ethernet- Physical Layer, Ethernet Mac Sub layer, data link layer switching & use of bridges, learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.

**UNIT III**
**Network Layer:** Network Layer Design issues, store and forward packet switching connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Count to Infinity Problem, Hierarchical Routing, Congestion control algorithms, admission control.

**UNIT IV**
Internetworking: Tunneling, Internetwork Routing, Packet fragmentation, IPv4, Ipv6 Protocol, IP addresses, CIDR, IMCP, ARP, RARP, DHCP.
**Transport Layer:** Services provided to the upper layers elements of transport protocol-addressing connection establishment, connection release, Crash Recovery.

**UNIT V**
The Internet Transport Protocols UDP-RPC, Real Time Transport Protocols, The Internet Transport Protocols- Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Connection Management Modeling, The TCP Sliding Window, The TCP Congestion Control, The future of TCP.
Application Layer-Introduction, providing services, Applications layer paradigms, Client server model, Standard client-server application-HTTP, FTP, electronic mail, TELNET, DNS, SSH.

**TEXT BOOKS:**
1) Computer Networks, Andrew S. Tanenbaum, David J Wetherall, Pearson Education, 5th Edition.
2) Computer Networks A Top-Down Approach, Behrouz A Forouzan, Firouz Mosharraf, TMH.

**REFERENCE BOOKS:**
1. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W.A.Shay, Cengage Learning.
3. Computer Networks, L.L.Peterson and B.S.Davie, 4th edition, ELSEVIER.
4. Computer Networking: A Top-Down Approach Featuring the Internet, James F.Kurose, K.W.Ross, 3rd Edition, Pearson Education.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - I Sem.**

## OPERATING SYSTEMS

**Objectives:**
- Understand main components of OS and their working.
- To study the operations performed by OS as a resource manager.
- Understand the scheduling policies of OS.
- Understand the different memory management techniques.
- Understand process concurrency and synchronization.
- Understand the concepts of input/output, storage and file management.
- To study different OS and compare their features.

**Outcomes:**
- Apply optimization techniques for the improvement of system performance.
- Ability to understand the synchronous and asynchronous communication mechanisms in their respective OS.
- Learn about minimization of turnaround time, waiting time and response time and also maximization of throughput with keeping CPU as busy as possible.
- Ability to compare the different OS.

**UNIT I**
Operating System Introduction: Operating Systems objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special -Purpose Systems, Operating System services, User OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Virtual Machines.

**UNIT II**
Process and CPU Scheduling - Process concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies: Linux, Windows.
Process Coordination – Process Synchronization, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

**UNIT III**
Memory Management and Virtual Memory - Logical & Physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames, Thrashing, Case Studies: Linux, Windows.

**UNIT IV**
File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance, Case Studies: Linux, Windows.
Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management

**UNIT V**
Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

Protection – System Protection, Goals of Protection, Principles of  Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection, Case Studies: Linux, Windows.

**TEXT BOOKS:**
1.	Operating System Principles, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley Student Edition.
2.	Operating Systems – Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

**REFERENCE BOOKS:**
1.	Modern Operating Systems, Andrew S Tanenbaum,  3rd Edition, PHI.
2.	Operating Systems A concept-based Approach, 2nd Edition, D.M.Dhamdhere, TMH.
3.	Principles of Operating Systems, B.L.Stuart, Cengage learning, India Edition.
4.	Operating  Systems, A.S.Godbole,2nd Edition, TMH.
5.	An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6.	Operating Systems, S.Haldar and A.A.Aravind, Pearson Education.
7.	Operating Systems, R.Elmasri, A.G.Carrick and D.Levine, Mc Graw Hill.
8.	Operating Systems in depth,T.W. Doeppner, Wiley.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - I Sem.**

### OBJECT ORIENTED ANALYSIS AND DESIGN USING UML

**Objectives**
- Concisely define the following key terms: class, object, state, behavior, object class, class diagram, object diagram, operation, encapsulation, constructor operation, query operation, update operation, scope operation, association, association role, multiplicity, association class, abstract class, concrete class, class-scope attribute, abstract operation, method, polymorphism, overriding, multiple classification, aggregation, and composition.
- Describe the activities in the different phases of the object-oriented development life cycle.
- State the advantages of object-oriented modeling vis-à-vis structured approaches.
- Compare and contrast the object-oriented model with the E-R and EER models.
- Model a real-world application by using a UML class diagram.
- Provide a snapshot of the detailed state of a system at a point in time using a UML (Unified Modeling Language) object diagram.
- Recognize when to use generalization, aggregation, and composition relationships.
- Specify different types of business rules in a class diagram.

**Outcomes:**
Master student can able to take up the case studies and model it in different views with respect user requirement such as use case, logical, component and deployment and etc, and preparation of document of the project for the unified Library application.

**UNIT I**
Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

**UNIT II**
Basic Structural Modeling: Classes, Relationships, common Mechanisms, diagrams.
Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.
Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object   Diagrams.

**UNIT III**
Basic Behavioral Modeling-I: Interactions, Interaction diagrams.
Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

**UNIT IV**
Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams.
Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

**UNIT V**
Patterns and Frameworks, Artifact Diagrams. Case Study: The Unified Library application

**TEXT BOOKS:**
1.    Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education 2nd Edition.
2.    Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

**REFERENCE BOOKS:**
1.    Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.

2.  Pascal Roques: Modeling Software Systems Using UML2, WILEY-Dreamtech India Pvt. Ltd.
3.  Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4.  Mark Priestley: Practical Object-Oriented Design with UML,TMH.
5.  Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.
6.  Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning.
7.  UML and C++,R.C.Lee, and W.M.Tepfenhart, PHI.
8.  Object Oriented Analysis, Design and Implementation, B.Dathan, S.Ramnath, Universities Press.
9.  OODesign with UML and Java, K.Barclay, J.Savage, Elsevier.
10.  Learning UML 2.0,Russ Miles and Kim Hamilton, O'Reilly, SPD.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - I Sem.**

**JAVA PROGRAMMING**

**Objectives**

- Understand the concept of OOP as well as the purpose and usage s of inheritance, polymorphism, and encapsulation principles.
- Identify classes, objects, members of a class and the relationships among them needed for a specific problem.
- Develop Java application programs using sound OOP practices(ex. Interfaces and APIs)
- Develop programs using the Java collection APIs as well as Java standard class library.

**Outcomes:**

- Understanding of OOP concepts and basics of java programming (Console and GUI based).
- The skills to apply OOP and Java programming in problem solving.
- Should have the ability to extend his knowledge of Java programming further on his/her own.

**UNIT I**

**Java Basics -** History of Java, Java buzzwords, comments, data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow-block scope, conditional statements, loops, break and continue statements, simple java program, arrays, input and output, formatting output, Review of OOP concepts, encapsulation, inheritance, polymorphism,  classes, objects, constructors, methods, parameter passing, static fields and methods, access control, this reference, overloading methods and constructors, recursion, garbage collection, building strings, exploring string class, Enumerations, autoboxing and unboxing, Generics.

**UNIT II**

**Inheritance** – Inheritance concept, benefits of inheritance, Super classes and Sub classes, Member access rules, Inheritance hierarchies, super uses, preventing inheritance: final classes and methods, casting, polymorphism- dynamic binding, method overriding, abstract classes and methods, the Object class and its methods. **Interfaces** – Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interface. **Inner classes**– Uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples.
**Packages-**Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

**UNIT III**

**Data structures creation and manipulation in java –** Introduction to Java Collections, Overview of Java Collection frame work, Commonly used Collection classes– ArrayList, LinkedList, HashSet, HashMap, TreeMap, Collection Interfaces – Collection, Set, List, Map, Legacy Collection classes – Vector, Hashtable, Stack, Dictionary(abstract), Enumeration interface, Iteration over Collections – Iterator interface, ListIterator interface. Other Utility classes – String Tokenizer, Formatter, Random, Scanner, Observable, java.util. **Files** – streams- byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class, java.io. **Networking** – Introduction, Manipulating URLs, Ex. Client/Server Interaction with Stream Socket Connections, Connectionless Client/Server Interaction with Datagrams, java.net.

**UNIT IV**

**Exception handling**  – Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

Guide lines for proper use of exceptions. **Multithreading -** Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, interthread communication, thread groups, daemon threads.

**UNIT V**

**GUI Programming with Java** - The AWT class hierarchy, Introduction to Swing, Swing vs. AWT, MVC architecture, Hierarchy for Swing components, Containers – Top-level containers – JFrame, JApplet, JWindow, JDialog, Light weight containers – JPanel, A simple swing application, Overview of several swing components- Jbutton, JToggleButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JList, JComboBox, JMenu, Java's Graphics capabilities – Introduction, Graphics contexts and Graphics objects, color control, Font control, Drawing lines, rectangles and ovals, Drawing arcs, Layout management - Layout manager types – border, grid, flow, box. **Event Handling -** Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Semantic and Low-level events, Examples: handling a button click, handling mouse and keyboard events, Adapter classes.

**Applets** – Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet - Four methods of an applet, Developing applets and testing, passing parameters to applets, applet security issues..

**TEXT BOOKS:**

1.     Java: the complete reference, $8^{th}$ edition, Herbert Schildt, TMH.

2.     Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education / Java: How to Program P.J.Deitel and H.M.Deitel, $8^{th}$ edition, PHI.

**REFERENCE BOOKS:**

1.     Java Programming, D.S.Malik, Cengage Learning.

2.     Core Java, Volume 1-Fundamentals, eighth edition, Cay S.Horstmann and Gary Cornell, Pearson Education.

3.     An introduction to Java programming and object oriented application development, R.A. Johnson-Cengage Learning.

4.     Advanced Programming in Java2, K.Somasundaram, Jaico Publishing House.

5.     Programming in Java, S.Malhotra and S.Choudhary, Oxford Univ. Press.

6.     Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.

7.     Object Oriented Programming through Java, P. Radha Krishna, Universities Press.

8.     An introduction to programming and OO design using Java, J.Nino, F.A.Hosch, John Wiley & Sons.

9.     Java and Object Orientation, an introduction, John Hunt, second edition, Springer.

10.    Maurach's Beginning Java2, D.Lowe, J.Murach, A.Steelman, SPD.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - I Sem.**

**DATABASE MANAGEMENT SYSTEMS LAB**

**Objectives**:
This lab enables the students to practice the concepts learnt in the subject Database management
systems.
 **Outcomes:**
o        Ability to design and implement a database schema for given problem.
o        Be capable to Design and build a GUI application.
o        Apply the normalization techniques for development of application software to realistic problems.
o        Ability to formulate queries using SQL DML/DDL/DCL commands.

**List of Sample Problems:**

1)   Creation, altering and dropping of tables and inserting rows into a table (use constraints while
creating tables) examples using SELECT command.
2)   Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET,
Constraints.
Example:- Select the roll number and name of the student who secured fourth rank in the class.
3)   Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and
Creation and dropping of Views.
4)   Queries using Conversion functions (to_char, to_number and to_date), string functions
(Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions
(Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char,
to_date).
5)   i)Creation of simple PL/SQL program which includes declaration section, executable section and
exception –handling section (Ex. Student marks can be selected from the table and printed for those who
secured first class and an exception can be raised if no records were found)
      ii)Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
6)   Develop a program that includes the features NESTED IF, CASE and CASE expression. The
program can be extended using the NULLIF and COALESCE functions.
7)   Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR
Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.
8)   Programs development using creation of procedures, passing parameters IN and OUT of
PROCEDURES.
9)   Program development using creation of stored functions, invoke functions in SQL Statements and
write complex functions.
10) Program development using creation of package specification, package bodies, private objects,
package variables and cursors and calling stored packages.
11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE
CURRENT of clause and CURSOR variables.
12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD
OF Triggers.

**Example Problems:**

**1. Creating tables for various relations (in SQL):**

CLIENT- MASTER  :( CLENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE).
PRODUCT_MASTER   :     (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNITMEASURE,
QTY_ON_HAND, REORDERLVL,
                    SELLPRICE, COSTPRICE)
SALESMAN_MASTER: (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY,

PINCODE,
STATE, SLAMT, TGTTOGET, YTDSALES, RESALES)

SALES_ORDER     : (OREDRENO, CLIENTNO, ORDERDATE, DELYADDR,
SALESMANNO, DELYTYPE,   BILLYN, DELYDATE, ORDERSTATUS)

SALES_ORDER_DETAILS :( ORDERNO, PRODUCTNO, QTYORDERED, QTYDISP, PRODUCTRATE)

i) Insert data into their respective table.

ii) Exercise on retrieving records from table.
   a. Find out the names of all clients.
   b. Retrieve the entire contents of the client_master table.
   c. Retrieve the list of names, city and the state of all clients.
   d. List the various products available from the Product_master table.
   e. List all the clients who are located in Mumbai.
   f. Find the names of salesmen who have a salary equal to Rs.3000.

iii. Exercise on updating records in a table
   a. Change the city of client No 'C00005' to 'Bangalore'.
   b. Change the BalDue of Client No 'C00001' to Rs.1000.
   c. Change the cost price of 'Trousers' to Rs.950.00.
   d. Change the city of the salesman to Pune.

iv. Exercise on deleting records in a table
   a. Delete all sales men from the salesman_master whose salaries are equal to Rs.35000.
   b. Delete all products from product_master where the quantity on hand is equal to 100.
   c. Delete from Client _Master where the column state holds the value 'Tamilnadu'.

v. Exercise on altering the table
a.      Add a column called 'Telephone' of data type 'number' and size='10' to the Client Master table.
   b. Change the size of Sell Price column in Product_Master to 10, 2.

vi. Exercise on deleting the table structure along with the data
   a. Destroy the table Client_Master along with the data.

vii. Exercise on renaming the table
  a. Change the name of the Salesman_Master table to Sman_mast.

2. Using the tables created previously generate the SQL statements for the operations mentioned below.
The tables in user are as follows:
 Client_Master
 Product_Master
 Salesman_Master
 Sales_Order
 Sales_Order_Details

i) Perform the following computations on table data:
a.     List the names of all clients having 'a' as the second letter in their names.
b.     List the clients who stay in a city whose first letter is 'M'.
c.     List all clients who stay in 'Bangalore' or 'Mangalore'.
d.     List all clients whose BalDue is greater than value 10000.
e.     List all information from the Sales_Order table for orders placed in the month of June.
f.     List the order information for Client No'C00001' and 'C00002'.
g.     List products whose selling price is greater than 500 and less than or equal to 750.
h.     List products whose selling price is more than 500.Calculate a new selling price as, original selling
price * .15. Rename the new column in the output of the above query as new_price.
i.     List the names, city and state of clients who are not in the state of  'Maharashtra'.
j.     Count the total no of orders.
k.     Calculate the average price of all the products.
l.     Determine the maximum and minimum products prices. Rename the output as max_price and
min_price respectively.
m.     Count the no of products having price less than or equal to 500.
n.      List all the products whose Qty On Hand is less than reorder level.

ii) Exercise on Date Manipulation

a.         List the order number and day on which clients on placed their order.

b.         List the months (in alphabets) and date when the orders must be delivered.

c.         List the Order Date in the format 'DD-Month-YY'. E.g.12-February-02.

d.         List the date, 15 days after today's date.

iii). Exercises on using Having and Group by Clauses:

a. Print the   description and total qty sold for each product.

b. Find the value of each product sold.

c. Calculate the average qty sold for each client that has a maximum order value
 of 15000.00.

       d. Find out the total of all the billed orders for the month of June.

iv). Exercises on Joins and Correlation:

a)         Find out the products, which have been sold to 'Ivan Bay Ross'.

b)         Find out the products and their quantities that will have to be delivered in the current month.

c)         List the product no and description of constantly sold products (i.e. rapidly moving products).

d)         Find the names of clients who have purchased 'Trousers'.

e)         List the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.

f)         Find the products and their quantities for the orders placed by 'Ivan Bay Ross' and 'Mamta Muzumdar'.

g)         Find the products and their quantities for the orders placed by Client No 'C00001' and 'C00002'.

v). Exercice on Sub-queries:

a. Find the Product No and Description of non_moving products i.e.
   Products not being sold.

b. List the customer Name, Address1, Address2, City and Pin Code for the client who has
   placed order no 'O19001'.

c. List the client names that have placed orders before the month of May'02.

d. List if the product 'Lycra Top' has been ordered by any client and print the
   Client_no, Name to whom it was sold.

e. List the names of clients who have placed orders worth Rs.10, 000 or more.

**3) Creating Views**
**4) Writing Assertions**
**5) Writing Triggers**
**6) Implementing Operations on relations (tables) using PL/SQL**
   Ex: Write a PL/SQL code block   to calculate the area of a circle for a value
      of radius  varying  from 5 to  9. Store the radius and the corresponding
      values of calculated area in an empty table named Areas(radius, area).

7) Creating FORMS.
8) Generating REPORTS.

**TEXT BOOKS:**

1.      Introduction to SQL, Mastering the relational DB languages, IV th edition, Rick F. Vander Lans, Pearson ed., 2007.

2.      Oracle PL/SQL, 3rd edition, Benjamin Rosenzweig, Elena Silvestrova, Pearson ed., 2004.

3.      Oracle, Forms Developer's Hand Book. Albert Lulushi, Pearson ed.,

4.      Oracle Database 11g PL/SQL Programming, M. McLaughlin, TMH.

5.      Oracle Pl/Sql. Programming, IV Edition, Steven Feuerstein. SPD,rp-2007.

6.      SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.

7.      Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL, Shah, PHI.

**Additional Problems:**
**i)Databases :**
**Objective**: This lab enables the students to practice the concepts learnt in the subject Databases by developing a database for an example company named "Roadway Travels" whose description is as follows. The student is expected to practice the designing, developing and querying a database in the context of example database "Roadway travels". Students are expected to use "Mysql" database.

**Roadway Travels**
**"Roadway Travels"** is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad.

The company wants to **computerize its operations** in the following areas:
•Reservations and Ticketing
•Cancellations

**Reservations & Cancellation:**
Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger.  One Passenger/person can book many tickets (to his/her family).

Cancellations are also directly handed at the booking office.

In the process of **computerization** of **Roadway Travels**   you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model   3.   Relational Model  4. Normalization   5. Creating the database 6. Querying.  **Students are supposed to work on these steps week wise and finally create a complete "Database System" to Roadway Travels.**   Examples are given at every experiment for guidance to students.

**Experiment 1: E-R Model**

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.
Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example:   **Entities:**
1.      BUS
2.      Ticket
3.      Passenger
   **Relationships:**
                1. Reservation
                2. Cancellation
**PRIMARY KEY ATTRIBUTES:**

1.              Ticket ID (Ticket Entity)
2.              Passport ID (Passenger Entity)
3.              Bus_NO(Bus Entity)

Apart from the above mentioned entities you can identify more. The above mentioned are few.
**Note:** *The student is required to submit a document by writing the Entities and Keys to the lab teacher.*

**Experiment 2: Concept design with E-R Model**

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

**Example: E-R diagram for bus**
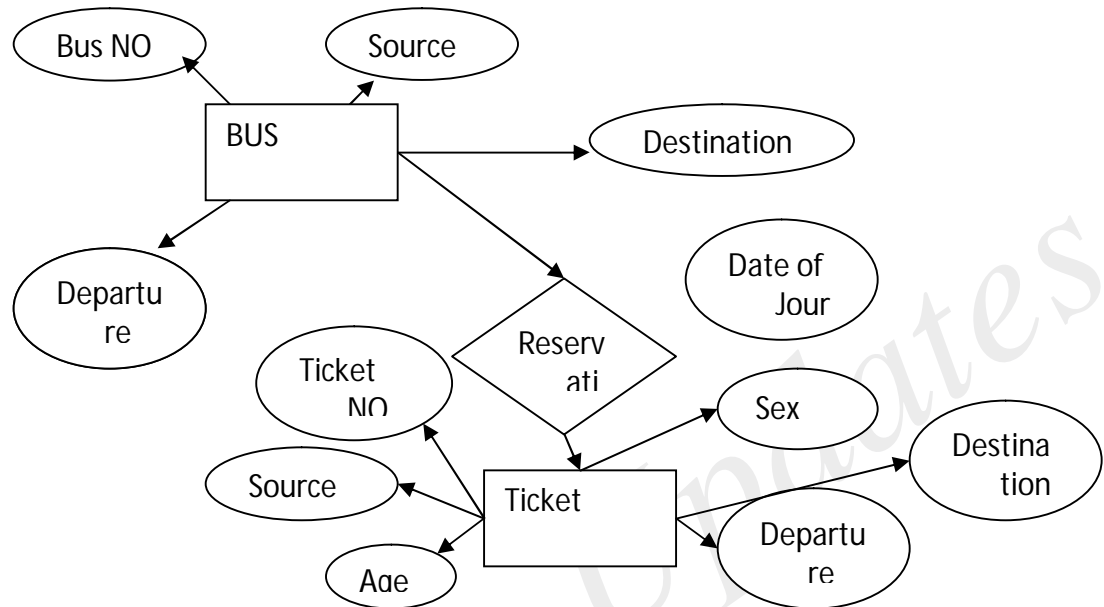


**Note:** *The student is required to submit a document by drawing the* E-R Diagram *to the* lab teacher.

**Experiment 3: Relational Model**

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

Passenger

| Name | Age | Sex | Address | | Passport ID |
|------|-----|-----|---------|-----------|-------------|
| | | | | Ticket_id | |
| | | | | | |
| | | | | | |

**Note:** *The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.*

**Experiment 4: Normalization**

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

For the above table in the First normalization we can remove the multi valued attribute Ticket_id and place it  in another table along with the primary key of passenger.

**First Normal Form: The above table can be divided into two tables as shown below.**

Passenger

| Name | Age | Sex | Address | Passport ID |
|---|---|---|---|---|
|  |  |  |  |  |
|  |  |  |  |  |

| Passport ID | Ticket_id |
|---|---|
|  |  |
|  |  |

You can do the second and third normal forms if required.  Anyhow Normalized tables are given at the end.

**Experiment 5: Installation of Mysql and practicing DDL commands**

Installation of MySql. In this week you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized "Passenger" table.

```
CREATE TABLE Passenger (
    Passport_id    INTEGER   PRIMARY KEY,
    Name   VARCHAR (50) Not NULL,
    Age    Integer Not NULL,
    Sex    Char,
    Address VARCHAR (50) Not NULL);
```

Similarly create all other tables.
**Note: Detailed creation of tables is given at the end.**

**Experiment 6: Practicing DML commands**
DML commands are used to for managing data within schema objects. Some examples:
- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

**Inserting values into "Bus" table:**

Insert into Bus values (1234,'hyderabad', 'tirupathi');
Insert into Bus values (2345,'hyderabd','Banglore');
Insert into Bus values (23,'hyderabd','Kolkata');
Insert into Bus values (45,'Tirupathi,'Banglore');
Insert into Bus values (34,'hyderabd','Chennai');

**Inserting values into "Passenger" table:**
Insert into Passenger values (1, 45,'ramesh', 45,'M','abc123');
Insert into Passenger values (2, 78,'geetha', 36,'F','abc124');
Insert into Passenger values (45, 90,'ram', 30,'M','abc12');
Insert into Passenger values (67, 89,'ravi', 50,'M','abc14');
Insert into Passenger values (56, 22,'seetha', 32,'F','abc55');

**Few more Examples of DML commands:**
Select * from Bus; (selects all the attributes and display)
UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;

**Experiment 7: Querying**
In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

**Practice the following Queries:**
1.      Display unique PNR_no of all passengers.
2.      Display all the names of male passengers.
3.      Display the ticket numbers and names of all the passengers.
4.      Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.
5.      Find the names of passengers whose age is between 30 and 45.
6.      Display all the passengers names beginning with 'A'
7.      Display the sorted list of passengers names

**Experiment 8 and Experiment 9: Querying (continued…)**

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
1.      Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
2.      Display the number of days in a week on which the 9W01 bus is available.
3.      Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
4.      Find the distinct PNR numbers that are present.
5.      Find the number of tickets booked by a passenger where the number of seats is greater than 1.
**Hint:** Use    GROUP BY, WHERE and HAVING CLAUSES.
6.      Find the total number of cancelled seats.

**Experiment 10: Triggers**

In this week you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger**
     **FOR EACH ROW**
     **BEGIN**
      **IF NEW.TickentNO > 60 THEN**
        **SET New.Tickent no = Ticket no;**
     **ELSE**
            **SET New.Ticketno = 0;**
     **END IF;**

    **END;**

## Experiment 11: Procedures

In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

**Eg: CREATE PROCEDURE myProc()**
**BEGIN**
**SELECT COUNT(Tickets) FROM Ticket WHERE age>=40;**
**End;**

## Experiment 12: Cursors

In this week you need to do the following: Declare a cursor that defines a result set.
Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done

**CREATE PROCEDURE myProc(in_customer_id INT)**
 **BEGIN**
 **DECLARE v_id  INT;**
 **DECLARE v_name VARCHAR (30);**
 **DECLARE  c1  CURSOR  FOR  SELECT  stdId,stdFirstname  FROM  students  WHERE stdId=in_customer_id;**
**OPEN c1;**
**FETCH c1 into v_id, v_name;**
**Close c1;**
**END;**

**Tables**

**BUS**
Bus No: Varchar: PK (public key)
Source : Varchar
Destination : Varchar

**Passenger**
PPNO: Varchar(15)) : PK
Name: Varchar(15)
Age: int (4)
Sex:Char(10) : Male / Female
Address: VarChar(20)

**Passenger_Tickets**
PPNO: Varchar(15)) : PK
Ticket_No: Numeric (9)

**Reservation**
PNR_No: Numeric (9): FK
Journey_date : datetime (8)
No_of_seats : int (8)
Address : Varchar (50)
Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer
Status: Char (2): Yes / No

**Cancellation**
PNR_No: Numeric (9) : FK
Journey_date : datetime(8)
No_of_seats : int (8)
Address : Varchar (50)
Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer
Status: Char (2): Yes / No

**Ticket**
Ticket_No: Numeric (9): PK
Journey_date : datetime(8)
Age : int (4)
Sex:Char(10) : Male / Female
Source: Varchar
Destination: Varchar
Dep_time : Varchar

**TEXT BOOKS:**
1. Introduction to SQL, Rick F.Vander Lans, Pearson education.
2. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, Pearson education.
3.Oracle PL/SQL Programming,Steven Feuerstein,SPD.
4. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S.Deshpande, Dream Tech.
5. Oracle Database 11g PL/SQL Programming, M. Mc Laughlin, TMH.
6. SQL Fundamentals, J.J.Patrick, Pearson Education.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - II Year - I SEM**

## JAVA PROGRAMMING LAB

**Objectives:**
- Developing Java programs for solving problems using OOP concepts.
- To implement some standard Unix utilities using system calls.
- To develop shell scripts to solve problems.
- To produce programs in C for network-based applications.

**List of Sample Problems:**

1. a)  Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

b)  The Fibonacci sequence is defined by the following rule:

The fist two values in the sequence are 1 and 1.

Every subsequent value is the sum of the two values preceding it.

Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.

2. a)  Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.

   b)  Write a Java program to multiply two given matrices.

   c)  Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)

3.  Write a Java program to find both the largest and smallest number in a list of integers.

4.  Write a Java program to illustrate method overloading.

5.  Write a Java program that implements the Sieve of Eratosthenes to find prime numbers.

6.  Write a Java program to sort a list of names in ascending order.

7.  Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:

a)      Reading a matrix.              d) Addition of matrices.

b)   Printing a matrix.              c)  Multiplication of matrices.

8.  Write a Java Program to solve Towers of Hanoi problem .

9.  Write a Java Program that uses a recursive function to compute ncr. (Note: n and r values are given.)

10. Write a Java program  to perform the following operations:

a)      Concatenation of two strings.

b)      Comparison of two strings.

11. Implement the complex number ADT in Java using a class. The complex ADT is used to represent complex numbers of the form c=a+ib, where a and b are real numbers. The operations supported by this ADT are:

a)      Reading a complex number.       c) Multiplication of complex numbers.

b)      Writing a complex number.        d) Addition of Complex numbers.

12. Write a Java program that makes frequency count of letters in a given text.

13. Write a Java program that uses functions to perform the following operations :

a)      Inserting a sub-string in to the given main string from a given position.

b)   Deleting n characters from a given position in a given string.

14. a) Write a Java program that checks whether a given string is a palindrome or not.  Ex: MADAM is a palindrome.

   **b**) Write a  Java program to make frequency count of words in a given text.

15 .a) Write a Java program that reads a file name from the user, then displays information  about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

   b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.

c) Write a Java program that displays the number of characters, lines and words in a text file.

d) Write a Java program to change a specific character in a file.

Note: Filename , number of the byte in the file to be changed and the new character are specified on the command line.

16. Write a Java program that illustrates the following

a)    Creation of simple package.  b) Accessing a package. c)   Implementing interfaces.

17. a) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.

b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

18. Write Java programs that illustrates the following

a) Handling predefined exceptions

b) Handling user defined exceptions

19. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.

20. a) Develop an applet in Java that displays a simple message.

b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.

21. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field  when the Divide button is clicked. If Num1 or Num2 were not and integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the  exception in a message dialog box.

22. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console.  For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)

23. a) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time.No light is on when the program starts.

b) Write a Java program that allows the user to draw lines, rectangles and ovals.

24 . a) Write a Java program to create an abstract class named Shape that contains an empty method named numberOfSides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one  of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides ( ) that shows the number of sides in the given geometrical  figures.

b) Suppose that a table named Table.txt is stored in a text file. The firstline in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Jtable component.

25. Write a Java program for handling Key events.

26. Write a Java program for handling mouse events. (Use  Adapter classes).

27. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:

a) Linear search             b) Binary search

28. Write Java programs to implement the List ADT using arrays and linked lists.

29. Write Java programs to implement the Stack ADT and Queue ADT using  arrays.

30. Write Java programs to implement the following using a singly linked list.

a) Stack ADT                b) Queue ADT .

31. Write Java programs for implementing the following sorting methods:

a) Bubble sort              c) Quick sort

b) Selection sort

32. Write a Java Program to perform the following:

a) Create a binary search tree of elements.

b) Search the above binary search tree for a key value.

c) Traverse the above binary search tree in inorder.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - II Year - II Sem.**

## WEB TECHNOLOGIES

**Objectives:**
- To introduce PHP language for server side scripting.
- To introduce XML and processing of XML Data with Java.
- To introduce Server side programming with Java Servlets and JSP .
- To introduce Client side scripting with Javascript and AJAX.

**Outcomes:**
- gain knowledge of client side scripting, validation of forms and AJAX programming .
- have understanding of server side scripting with PHP language .
- have understanding of what is XML and how to parse and use XML Data with Java.
- To introduce Server side programming with Java Servlets and JSP.

**UNIT I**
**Introduction to PHP**: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads, Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies.
**File Handling in PHP**: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

**UNIT II**
**XML**: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, XHTML.
**Parsing XML Data** - DOM and SAX Parsers in java.

**UNIT III**
**Introduction to Servlets**: Common Gateway Interface (CGI), Lifecycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

**UNIT IV**
**Introduction to JSP**: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

**UNIT V**
**Client side Scripting**: Introduction to Javascript: Javascript language - declaring variables, scope of variables, functions, event handlers (onclick, onsubmit etc.), Document Object Model, Form validation. Simple AJAX application.

**TEXT BOOKS**:
1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill .

**REFERENCE BOOKS**:
1. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dreamtech.
2. Java Server Pages –Hans Bergsten, SPD O'Reilly.
3. Java Script, D.Flanagan, O'Reilly, SPD.
4. Beginning Web Programming-Jon Duckett WROX.
5. Programming World Wide Web, R.W.Sebesta, Fourth Edition, Pearson.
6. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - II Year - II Sem.**

## LINUX PROGRAMMING

**Objectives:**
- To understand the LINUX system structure.
- To understand and use command line shell.
- To make effective use of Unix utilities and Shell scripting language such as bash.
- To produce programs similar to standard unix utilities such as ls, mv, cp etc. using Unix system calls.
- To develop the skills necessary for Unix systems programming including file system programming, process and signal management, and interprocess communication.
- To develop the basic skills required to write network programs using Sockets.

**Prerequisites:** Familiarity with using Unix Programming environment and having a good working knowledge of the C programming language.

**Outcomes:**
- Work confidently in Linux environment.
- Work with shell script to automate different tasks as Linux administration.

**UNIT I**

Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

Sed-Scripts, Operation, Addresses, Commands, Applications, awk- Execution, Fields and Records, Scripts, Operation, Patterns, Actions, Associative Arrays, String and Mathematical functions, System commands in awk, Applications..

Introduction, shell responsibilities, pipes and  Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

Review of C programming concepts-arrays, strings (library functions), pointers, function pointers, structures, unions, libraries in C.

**UNIT II**

Files and Directories- File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, create, read, write, close, lseek, dup2, file status information-stat family, file and record locking-lockf and fcntl functions, file permissions - chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links – symlink, link, unlink.

Directories-Creating, removing and changing Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir, seekdir, telldir functions.

**UNIT III**

Process – Process concept, Layout of a C program image in main memory, Process environment-environment list, environment variables, getenv, setenv, Kernel support for process,  process identification, process hierarchy, process states, process control - process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, system, I/O redirection, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes.

Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.

**UNIT IV**

Interprocess Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using

unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions.
Message Queues- Kernel support for messages, APIs for message queues, client/server example.
Semaphores-Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

**UNIT V**
Shared Memory- Kernel support for shared memory, APIs for shared memory, shared memory example.
Sockets- Introduction to Berkeley Sockets, IPC over a network, Client-Server model, Socket address structures (Unix domain and Internet domain),Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection, Multiple simultaneous clients, Comparison of IPC mechanisms.

**TEXT BOOKS:**
1. Unix System Programming using C++, T.Chan, PHI.
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH, 2006.
3. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones, Wrox, Wiley India Edition, rp-2008.
4. Unix Network Programming, W.R.Stevens, PHI.
5. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.

**REFERENCE BOOKS:**
1. Linux System Programming, Robert Love, O'Reilly, SPD, rp-2007.
2. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education, 2003.
3. Advanced Programming in the Unix environment, 2nd Edition, W.R.Stevens, Pearson Education.
4. System Programming with C and Unix, A.Hoover, Pearson.
5. Unix System Programming, Communication, Concurrency and Threads, K.A.Robbins and S.Robbins, Pearson Education.
6. Unix shell Programming, S.G.Kochan and P.Wood, 3rd edition, Pearson Education.
7. Shell Scripting, S.Parker, Wiley India Pvt. Ltd.
8. C Programming Language, Kernighan and Ritchie, PHI.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem.**

### MANAGEMENT INFORMATION SYSTEMS
### (Core Elective-I)

**Objectives:**
One of the main aims of this subject is to analyze the system in such a way that a programme description can be made.  The student will become conversant with the various ways of analyzing the system.  The ultimate aim of the student is to write a system design.  There after he is supposed to become conversant with the implementation of the software in the organization and the problems encountered by him.  These elements will be useful to the student in career.

- To understand MIS within organization.
- To understand Information Systems and Strategy Formulation
- To understand Conceptual System Design.  Understand the basic functioning of the organization. Writing a report.
- To understand Detailed System Design   Understand the minute working of the functions of the organization.  Writing a report.
- To understand Implementation of software.
- To understand the pitfalls in MIS development.

**Prerequisites** Organization working which the student will pursue in earlier classes.

### UNIT I
The meaning and role of MIS: What is MIS? Decision support systems, systems approach, the systems view of business, MIS Organizational within the Company.
Management, Organizational Theory, and the Systems Approach: Development of Organizational Theory, Management and Organizational Behavior, Management, Information and the Systems Approach

### UNIT II
Conceptual system design: Define the problems, set system objectives, establish system constraints, determine information needs, determine information sources, develop alternative conceptual designs and select one, document the system concept, prepare the conceptual design report. Organizing data and information: Data warehouses, Data mart and data mining

### UNIT III
Detailed system design: Inform and involve the organization, aim of detailed design, project management of MIS detailed design, identify dominant and trade off criteria, define the subsystems, sketch the detailed operating subsystems and information flows, determine the degree of automation of each operation, inform and involve the organization again, inputs, outputs, and processing, early system testing, software, hardware and tools, propose an organization to operate the system, document the detailed design, revisit the manager-user.

### UNIT IV
Implementation, evaluation and maintenance of the MIS: Plan the implementation, acquire floor space and plan space layouts, organize for implementation, develop procedures for implementation, train the operating personnel, computer related acquisitions, develop forms for data collection and information dissemination, develop the files, test the system, cutover, document the system, evaluate the MIS, control and maintain the system.

### UNIT V
Pitfalls in MIS development: Fundamental weaknesses, soft spots, in planning, design problems, implementation: the TAR PIT.

### TEXT BOOKS:
1. Information systems for modern management, 3rd Edition by R.G Murdick, J.E Ross and J. R clagget, PHI-1994.

2. Management Information Systems, Managing the Digital Firm Edition by Kenneth C. Laudon, Jane P. Laudon, Pearson Education, 10th Edition.

**REFERENCE BOOKS:**
1. Management information Systems, 4th edition by Robert Schultheis, Mary Sumner, PHI- Seventeenth Reprint 2007.
2 Principles of Information systems, Sixth edition by Ralph M.Stair, George W.Reynolds, Cengage learning.
3 Management Information Systems, J.A.O'brien, G.M.Marakas, R.Behl, 9<sup>th</sup> Edition, TMH.
4 Management Information Systems, Effy Oz, Cengage Learning.
5 Managing and Using Information Systems, K.E.Pearlson, C.S.Saunders, Wiley India.
6 Management information Systems, M.Jaiswal & M.Mital, Oxford University Press.
7 MIS, Rahul De, First edition, Wiley India.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem.**

## ORGANIZATION STRUCTURE AND PERSONNEL MANAGEMENT
### (Core Elective-I)

**Objectives:**
One of the main objectives of this course is to provide the students with an in depth understanding of organizations. Organizations are a main tool that modern society applies to meet the challenges of innovation and resolution of complex tasks. To understand how to organize to resolve the challenges organizations meet is a main focus of this course. The course reviews the main contributions to organization research.

This is an introductory course in Organization Theory. In this course we will explore the basic theories and principles around which contemporary organizations are structured in complex, dynamic, uncertain, and competitive environments. The course examines the effects of both the internal and external environment on managerial choices for the structuring of organizations, as well as the implications that stem from these choices.

This course also covers Personnel Management and Communication.  In this course we will explore the basic theories of Personnel Management and its functions.  The object of the study is to do the analysis of various functions of the organization in functional area of Personnel Management so that the student should be in a position to create data bases and programmes in the content of Personnel Department.

**Outcomes:**
 The course will help each student to better:
• Understand the importance of organizational structure and design on internal organizational processes and overall effectiveness.
• Understand the relationships between organization structure and the behavior of those who work in them or otherwise interact with them.
• Recognize the managerial implications of organization design and change and how these are informed by the relevant theories.
• Appreciate the impact of advanced technologies on the strategy and structure of organizations and how to address the changes implied by the adoption of these technologies.
• Acquire the knowledge and skills needed to analyze the design and structure of organizations through a combination of lectures, discussions, and cases.
• Understand the Personnel Functions like position of the personnel department in the organization.
• Understand manpower planning, job description, interviewing techniques, transfers, promotion and its policies.
• Understand the training and development and career planning and Performance Appraisal.
• Obtain and practice effective written and oral business communications skills.

**UNIT I**
Classical  Theories of organization : Functional  approach, classical theories of organization, division of labour, levels of authority, span of control, authority & responsibility, efficiency  of management. Behavioral theories  of organization, limitations of formal organization, human relation, group behavior, committee and group making, motivation and morale.

**UNIT II**
Personnel Function: Evaluation, objectives, principles, philosophies and policies, duties & responsibilities of the manager, position of the personnel department in the organization, line and staff relationship & the changing concept of personnel management in India.

**UNIT III**
Manpower planning : Uses benefits problems and limitations, manpower inventory, manpower forecasting, job description, recruitment, Job specification and job selection, interviewing techniques, transfers, promotion and its policies.
Training and development : Objectives and policies planning, organizing the training department, training manager and his job, on and off the job training, techniques, career planning, objectives of performance appraisal.

**UNIT IV**
Strategic management: Objectives, importance policies, concept of core competence capability of organizational learning, strategic levels and planning, business level strategy and functional level, PHASES OF PLANNING,SWOT, develop strategies and prepare strategic plan.

**UNIT V**
Communication : Importance of communication, inter personnel communication barriers of communication, communication in organizations, using communication skills to manage conflicts. Impact of informational technology and fostering effective communication

**TEXT BOOKS:**
1.      L.M.Prasad, Principles and Practice of Management, Sultan Chand & Sons.
2.      A.R.Aryasri, Organizational Structure and Personnel Management, TMH, 2009

**REFERENCE BOOKS:**
1.      Hellriegel, Jackson and Slocum, Edition 9, Management-A competency – Based Approach
2.      L.M.Prasad, Human Resource Management.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem.**

## INFORMATION SECURITY
### (Core Elective-I)

**Objectives:**

- Explain the objectives of information security.
- Explain the importance and application of each of confidentiality, integrity, authentication and availability.
- Understand various cryptographic algorithms.
- Understand the basic categories of threats to computers and networks.
- Describe public-key cryptosystem.
- Describe the enhancements made to IPv4 by IPSec.
- Understand Intrusions and intrusion detection.
- Discuss the fundamental ideas of public-key cryptography.
- Generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
- Discuss Web security and Firewalls.

**UNIT I**

**Attacks on Computers and Computer Security:** Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security.

**Cryptography: Concepts and Techniques:** Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

**UNIT II**

**Symmetric key Ciphers:** Block Cipher principles & Algorithms(DES, AES, Blowfish), Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers, RC4,Location and placement of encryption function, Key distribution **Asymmetric key Ciphers:** Principles of public key cryptosystems, Algorithms(RSA, Diffie-Hellman,ECC), Key Distribution.

**UNIT III**

**Message Authentication Algorithms and Hash Functions:** Authentication requirements, Functions, Message authentication codes, Hash Functions, Secure hash algorithm, Whirlpool, HMAC, CMAC, Digital signatures, knapsack algorithm **Authentication Applications:** Kerberos, X.509 Authentication Service, Public – Key Infrastructure, Biometric Authentication.

**UNIT IV**

**E-Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, key management

**UNIT V**

**Web Security:** Web security considerations, Secure Socket Layer and Transport Layer Security, Secure electronic transaction **Intruders, Virus and Firewalls:** Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls **Case Studies on Cryptography and security:** Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability, Virtual Elections.

**TEXT BOOKS:**

1. Cryptography and Network Security : William Stallings, Pearson Education,4th Edition.
2. Cryptography and Network Security : Atul Kahate, Mc Graw Hill, 2nd Edition.

**REFERENCE BOOKS:**
1.      Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1<sup>st</sup> Edition.
2.      Cryptography and Network Security : Forouzan Mukhopadhyay, Mc Graw Hill, 2<sup>nd</sup> Edition.
3.      Information Security, Principles and Practice: Mark Stamp, Wiley India.
4.      Principles of Computer Security: WM.Arthur Conklin, Greg White, TMH.
5.      Introduction to Network Security: Neal Krawetz, CENGAGE Learning.
6.      Network Security and Cryptography: Bernard Menezes, CENGAGE Learning.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem.**

### DISTRIBUTED SYSTEMS
### (Core Elective II)

**Objectives:**

- Understand the need for distributed systems and their applications.
- Understand the concepts of remote procedure calls, remote file systems, distributed agreement, clock synchronization, and security.

**UNIT I**

Characterization of Distributed Systems-Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models-Introduction, Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication.

Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

**UNIT II**

Operating System Support- Introduction, OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture, Distributed File Systems-Introduction, File Service architecture, case study- SUN network file systems.

Name Services-Introduction, Name Services and the Domain Name System, Case study of the Global Name Service, Case study of the X.500 Directory Service.

**UNIT III**

Peer to Peer Systems–Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies-Squirrel, OceanStore, Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement-Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

**UNIT IV**

Transactions and Concurrency control-Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control. Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication-Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data.

**UNIT V**

Security-Introduction, Overview of Security techniques, Cryptographic algorithms, Digital signatures, Case studies-Kerberos, TLS, 802.11 Wi-Fi.

Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy case study, Release consistency and Munin case study, Other consistency models, CORBA case study-Introduction, CORBA RMI, CORBA Services.

**TEXT BOOKS:**

1. Distributed Systems Concepts and Design, G Coulouris, J Dollimore and T Kindberg, Fourth Edition, Pearson Education.
2. Distributed Systems, S.Ghosh, Chapman& Hall/CRC, Taylor & Francis Group, 2010.

**REFERENCE BOOKS:**

1. Distributed Computing, S.Mahajan and S.Shah, Oxford University Press.
2. Distributed Operating Systems Concepts and Design, Pradeep K.Sinha, PHI.
3. Advanced Concepts in Operating Systems, M Singhal, N G Shivarathri, TMH.

4.      Reliable Distributed Systems, K.P.Birman, Springer.
5.      Distributed Systems – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, Pearson Education.
6.      Distributed Operating Systems and Algorithm Analysis, R.Chow, T.Johnson, Pearson.
7.      Distributed Operating Systems, A.S.Tanenbaum, Pearson education.
8.      Distributed Computing, Principles, Algorithms and Systems, Ajay D.Kshemakalyani and Mukesh Singhal, Cambridge, rp 2010.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem**

**SOFTWARE PROCESS AND PROJECT MANAGEMENT**
**(Core Elective-II)**

The main goal of software development projects is to create a software system with a predetermined functionality and quality in a given time frame and with given costs. For achieving this goal, models are required for determining target values and for continuously controlling these values. This course focuses on principles, techniques, methods & tools for model-based management of software projects, assurance of product quality and process adherence (quality assurance), as well as experience-based creation & improvement of models (process management). The goals of the course can be characterized as follows:
1.  Understanding the specific roles within a software organization as related to project and process management.
2.  Understanding the basic infrastructure competences (e.g., process modeling and measurement).
3.  Understanding the basic steps of project planning, project management, quality assurance, and process management and their relationships.
**Objectives:**
    At the end of the course, the student shall be able to:
1. Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
2. Compare and differentiate organization structures and project structures.
3. Implement a project to manage project schedule, expenses and resources with the application of suitable project management tools.

**UNIT I**
Conventional Software Management: The waterfall model, conventional software
Management performance. Evolution of Software Economics: Software Economics, pragmatic software cost estimation.

**UNIT II**
Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.
The old way and the new: The principles of conventional software engineering, principles of modern software management, transitioning to an iterative process.

**UNIT III**
Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.
Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts. Model based software architectures: A Management perspective and technical perspective.

**UNIT IV**
Work Flows of the process: Software process workflows, Inter trans workflows. Checkpoints of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments.  Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Interaction planning process, Pragmatic planning.
Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.
Process Automation: Automation Building Blocks, The Project Environment.

**UNIT V**
Project Control and Process instrumentation: The server care Metrics, Management indicators, quality indicators, life cycle expectations pragmatic Software Metrics, Metrics automation. Tailoring the Process: Process discriminants, Example.

Future Software Project Management: Modern Project Profiles Next generation, Software economics, modern Process transitions.
Case Study : The Command Center Processing and Display System-Replacement(CCPDS-R)

**TEXT BOOKS:**
1.   Software Project Management, Walker Royce, Pearson Education.
2.   Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, Tata Mc Graw Hill.

**REFERENCE BOOKS:**
1.   Applied Software Project Management, Andrew Stellman & Jennifer Greene,O'Reilly, 2006.
2.   Head First PMP, Jennifer Greene & Andrew Stellman, O'Reilly,2007.
3.   Software Engineering Project Management, Richard H. Thayer & Edward Yourdon, second edition, Wiley   India, 2004.
4.   Agile Project Management, Jim Highsmith, Pearson education, 2004.
5.   The art of Project management, Scott Berkun, O'Reilly, 2005.
6.   Software Project Management in Practice, Pankaj Jalote, Pearson Education, 2002.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem.**

### SOFT COMPUTING
### (Core Elective-II)

**Objectives:**
To give students knowledge of soft computing theories fundamentals, ie. Fundamentals of artificial and neural networks, fuzzy sets and fuzzy logic and genetic algorithms.

**UNIT I**
AI Problems and Search: AI problems, Techniques, Problem Spaces and Search, Heuristic Search Techniques- Generate and Test, Hill Climbing, Best First Search Problem reduction, Constraint Satisfaction and Means End Analysis. Approaches to Knowledge Representation- Using Predicate Logic and Rules.

**UNIT II**
Artificial Neural Networks: Introduction, Basic models of ANN, important terminologies, Supervised Learning Networks, Perceptron Networks, Adaptive Linear Neuron, Back propagation Network. Associative Memory Networks. Training Algorithms for pattern association, BAM and Hopfield Networks.

**UNIT III**
Unsupervised Learning Network- Introduction, Fixed Weight Competitive Nets, Maxnet, Hamming Network, Kohonen Self-Organizing Feature Maps, Learning Vector Quantization, Counter Propagation Networks, Adaptive Resonance Theory Networks. Special Networks-Introduction to various networks.

**UNIT IV**
Introduction to Classical Sets ( crisp Sets)and Fuzzy Sets- operations and Fuzzy sets. Classical Relations -and Fuzzy Relations- Cardinality, Operations, Properties and composition. Tolerance and equivalence relations.
Membership functions- Features, Fuzzification, membership value assignments, Defuzzification.

**UNIT V**
Fuzzy Arithmetic and Fuzzy Measures, Fuzzy Rule Base and Approximate Reasoning Fuzzy Decision making
Fuzzy Logic Control Systems. Genetic Algorithm- Introduction and basic operators and terminology, Applications: Optimization of TSP, Internet Search Technique

**TEXT BOOKS:**
1       Principles of Soft Computing- S N Sivanandam, S N Deepa, Wiley India, 2007.
2       Soft Computing and Intelligent System Design -Fakhreddine O Karray, Clarence D Silva, Pearson Edition, 2004.

**REFERECE BOOKS :**
1.      Artificial Intelligence and SoftComputing- Behavioural and Cognitive Modeling of the Human Brain- Amit Konar, CRC press, Taylor and Francis Group.
2.      Artificial Intelligence – Elaine Rich and Kevin Knight, TMH, 1991, rp2008.
3.      Artificial Intelligence – Patric Henry Winston – Third Edition, Pearson Education.
4.      A first course in Fuzzy Logic-Hung T Nguyen and Elbert A Walker, CRC. Press Taylor and Francis Group.
5.      Artificial Intelligence and Intelligent Systems, N.P.Padhy, Oxford Univ. Press.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - II Year - II Sem.**

**WEB TECHNOLOGIES LAB**

**Objectives:**
The primary objective of the course is to learn web programming by designing and developing some web based applications.

**List of Sample Problems**

1. Develop static pages (using Only HTML) of an online Book store. The pages should resemble: www.amazon.com. The website should consist the following pages.
Home page, Registration and user Login
User Profile Page, Books catalog
Shopping Cart, Payment By credit card
Order Conformation
2.       Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
3.       Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
4.        Bean Assignments
a.       Create a JavaBean which gives the exchange value of INR(Indian Rupees) into equivalent American/Canadian/Australian Dollar value.
b.       Create a simple Bean with a label - which is the count of number of clicks. Than create a BeanInfo class such that only the "count" property is visible in the Property Window.
c.       Create two Beans-a)KeyPad .b)DisplayPad .After that integrate the two Beans to make it work as a Calculator.
d.       Create two Beans Traffic Light (Implemented as a Label with only three background colors-Red, Green, Yellow) and Automobile(Implemented as a TextBox which states its state/movement). The state of the Automobile should depend on the following Light Transition Table.

| Light Transition | Automobile State |
|---|---|
| Red   ---> Yellow | Ready |
| Yellow ---> Green | Move |
| Green --> Red | Stopped |

5.       Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
6.       Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.
7.        Implement the "Hello World!" program using JSP Struts Framework.
8.       Redo the problem 5 using PHP.

**Additional Assignment Problems for the WT  Lab.:**

Write an HTML page including any required Javascript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show "out of range" and if it is not a number, it should show "not a number" message in the result box.

Write a java swing application that takes a text file name as input and counts the characters, words and lines in the file. Words are separated with white space characters and lines are separated with new line character.

Write a simple calculator servlet that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operation performed on the operands. It should check in a database if the same expression is already computed and if so, just return the value from database. Use MySQL or PostgreSQL.(Do the same problem using PHP).

Write an HTML page that contains a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).

Write a servlet that takes name and age from an HTML page. If the age is less than 18, it should send a page with "Hello <name>, you are not authorized to visit this site" message, where <name> should be replaced with the entered name. Otherwise it should send "Welcome <name> to this site" message. (Do the same problem using PHP).

Write a calculator program in HTML that performs basic arithmetic operations (+, -, /, * and %). Use CSS to change the foreground and background color of the values, buttons and result display area separately. Validate the input strings using JavaScript regular expressions. Handle any special cases like division with zero reasonably. The screen may look similar to the following:



Write a Java program that creates a calculator GUI, as shown in figure. Extra components may be added for convenience:



The Color Scheme may be Black on White or Blue on Yellow (selectable) and accordingly all components colors must be changed. The values can be either entered or increased or decreased by a step of 10. The operators are +, -, / and * (selectable). Once any change takes place, the result must be automatically computed by the program.

Write a Java Application that will read an XML file that contains personal information (Name, Mobile Number, age and place. It reads the information using SAX parser. After reading the information, it shows two input Text Fields in a window, one for tag name and the other for value. Once these two values are given, it should list all the records in the XML file that match the value of the given field in a text area (result box). For example, if the two text boxes are entered with "name" and "ABCD" then it should show all the records for which name is "ABCD"? An Illustration is given below that takes a mobile number and lists all the records that have the same mobile number.

Consider the following web application for implementation:

The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.

If name and password matches, serves a welcome page with user's full name.

If name matches and password doesn't match, then serves "password mismatch" page.

If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application using:

1.      Pure JSP
2.      Pure Servlets
3.      Struts Framework
4.      PHP

Implement a simple arithmetic calculator with +, -, /, *, % and = operations using Struts Framework The number of times the calculator is used should be displayed at the bottom (use session variable).

### JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - II Year - II Sem.**

### LINUX PROGRAMMMING LAB

**Note: Use Bash for Shell scripts.**

1.      Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.

2.      Write a shell script that deletes all lines containing a specified word in one or more files  supplied as arguments to it.

3.      Write a shell script that displays a list of all the files in the current directory to which the  user has read, write and execute permissions.

4.      Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly.  Whenever the argument is a file, the number of lines on it is also reported.

5.      Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.

6.      Write a shell script to list all of the directory files in a directory.

7.      Write a shell script to find factorial of a given integer.

8.      Write an awk script to count the number of lines in a file that do not contain vowels.

9.      Write an awk script to find the number of characters, words and lines in a file.

10.      Write a C program that makes a copy of a file using standard I/O and system calls.

11.            Implement in C the following Unix commands using System calls

     a). cat            b) mv

12.            Write a  C program to list files in a directory.

13.      Write a C program to emulate the Unix ls –l command.

14.      Write a C program to list for every file in a directory, its inode number and file name.

15.      Write a C program that redirects standard output to a file.Ex: ls > f1.

16.      Write a C program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen.

17.      Write a C program to create a Zombie process.

18.      Write a C program that illustrates how an orphan is created.

19.      Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls –l | sort

20.      Write  C programs that illustrate communication between two unrelated processes using named pipe.

21.      Write a C program in which a parent writes a message to a pipe and the child reads the message.

22.            Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.

23.                  Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (22)) and displays them.

24.                  Write a C programs to transfer a large amount of data between processes,using

     a) a pipe   b)a FIFO   c)a message queue.

25.                  Write a C program to allow cooperating processes to lock a resource for exclusive use, using:      a)Semaphores      b)flock or lockf system calls.

26.                  Write a C program that illustrates suspending and resuming processes using signals.

27.                  Write a C program that implements a producer-consumer system with two processes.

                  (using Semaphores).

28.                  Write client and server programs(using c) for interaction between server and client processes using Unix Domain sockets.

29.           Write client and server programs(using c) for interaction between server and client processes using Internet Domain sockets.

30.           Write C programs that illustrate two processes communicating using shared memory.

**TEXT BOOKS:**
1.        Advanced Unix Programming, N.B.Venkateswarulu, BS Publications.
2.        Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.
3.        Unix and Shell Programming, M.G. Venkatesh Murthy, Pearson Education, 2005.
4.        Unix Shells by Example, 4th Edition, Elllie Quigley, Pearson Education.
5.        Sed and Awk, O.Dougherty&A.Robbins,2nd edition

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**
**ANDROID MOBILE APPLICATION DEVELOPMENT**

**Unit I:**
*Introduction to Android Operating System:*
Android OS design and Features – Android development framework, SDK features, Installing and running applications on Eclipse platform, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools
Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes
Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

**Unit II:**
*Android User Interface:*
Measurements – Device and pixel density independent measuring units
Layouts – Linear, Relative, Grid and Table Layouts
User Interface (UI) Components – Editable and non editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers
Event Handling – Handling clicks or changes of various UI components
Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

**Unit III**
*Intents and Broadcasts:*
Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS
Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity
Notifications – Creating and Displaying notifications, Displaying Toasts

**Unit IV**
*Persistent Storage:*
Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory
Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference
Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

**Unit V**
*Advanced Topics:*
Alarms – Creating and using alarms
Using Internet Resources – Connecting to internet resource, using download manager
Location Based Services – Finding Current Location and showing location on the Map, updating location

**Text Books:**
1.      Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012
2.      Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
**References:**
1.      Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**

**SOFTWARE TESTING METHODOLOGIES**

**Objectives:**
To understand different software testing techniques and strategies.

**UNIT I**
Introduction: - Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs.
Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

**UNIT II**
Transaction Flow Testing:-transaction flows, transaction flow testing techniques. Dataflow testing: - Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

**UNIT III**
Domain Testing:-domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domains and testability.
Paths, Path products and Regular expressions: - path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

**UNIT IV**
Logic Based Testing:- overview, decision tables, path expressions, kv charts, specifications.
State, State Graphs and Transition testing:- state graphs, good & bad  state graphs, state testing, Testability tips.

**UNIT V**
Graph Matrices and Application:-Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

**TEXT BOOKS:**
1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.

**REFERENCE BOOKS:**
1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing,3rd edition, P.C.Jorgensen, Aurbach Publications(Dist. by SPD).
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.
5. Art of Software Testing – Meyers, John Wiley.
6. Software Testing, N.Chauhan, Oxford University Press.
7. Software Testing, M.G.Limaye, TMH.
8. Software Testing, S.Desikan, G.Ramesh, Pearson.
9. Foundations of Software Testing, D.Graham & Others, Cengage Learning.
10. Foundations of Software Testing, A.P.Mathur, Pearson.
11. Software Testing, S.Desikan, G.Ramesh, Pearson.
12. Foundations of Software Testing, D.Graham & Others, Cengage Learning.

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - III Year - I Sem.**

## WEB SERVICES
### (Core Elective-III)

**Objectives:**
- To understand the details of Web services technologies: SOAP, WSDL, UDDI.
- To learn how to implement and deploy web service clients and servers.
- To explore interoperability between different frameworks.
- To learn basic concepts of SOA.

### UNIT I
**Evolution and Emergence of Web Services -** Evolution of distributed computing, Core distributed computing technologies – client/server, CORBA, JAVA RMI, Microsoft DCOM, MOM, Challenges in Distributed Computing, role of J2EE and XML in distributed computing, emergence of Web Services and Service Oriented Architecture (SOA).
**Introduction to Web Services** – The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services.
**Web Services Architecture** – Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication models, basic steps of implementing web services.

### UNIT II
**Core fundamentals of SOAP** – SOAP Message Structure, SOAP encoding, SOAP message exchange models, SOAP communication and messaging, SOAP security.
**Developing Web Services using SOAP** – Building SOAP Web Services, developing SOAP Web Services using Java and Axis, limitations of SOAP.

### UNIT III
**Describing Web Services** – WSDL – WSDL in the world of Web Services, Web Services life cycle, anatomy of WSDL definition document, WSDL bindings, WSDL Tools, limitations of WSDL.
**Discovering Web Services** – Service discovery, role of service discovery in a SOA, service discovery mechanisms, UDDI – UDDI Registries, uses of UDDI Registry, Programming with UDDI, UDDI data structures, Publishing API, Publishing, searching and deleting information in a UDDI Registry, limitations of UDDI.

### UNIT IV
**Web Services Interoperability –** Means of ensuring Interoperability, Overview of .NET, Creating a .NET client for an Axis Web Service, Challenges in Web Services Interoperability.
**Web Services Security** – XML security frame work, Goals of Cryptography, Hash Cipher, Symmetric Cipher, Asymmetric Cipher, XML encryption, Digital signature, Digital Certificate, XML Encryption, SAML, structure.

### UNIT V
**Overview of Service Oriented Architecture** – SOA concepts, Key Service Characteristics, Technical Benefits of a SOA.
**SOA and Web Services** – Web Services Platform, Service-Level Data Models, Discovery, Security and Interaction Patterns, Atomic and Composite services, Service-level communication and alternative transports.

### TEXT BOOKS:
1. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India,   rp – 2008.
2. Understanding SOA with Web Services, Eric Newcomer and  Greg Lomow, Pearson Edition – 2009.
3. Java Web Service Architecture, James McGovern, Sameer Tyagi et al., Elsevier – 2009.

**REFERENCE BOOKS:**

1. Building Web Services with Java, 2$^{nd}$ Edition, S. Graham and others, Pearson Edn., 2008.
2. Java Web Services, D.A. Chappell & T. Jewell, O'Reilly, SPD.
3. McGovern, et al., "Java Web Services Architecture", Morgan Kaufmann Publishers,2005.
4. J2EE Web Services, Richard Monson-Haefel, Pearson Education.
5. Web Services, G. Alonso, F. Casati and others, Springer, 2005.
6. Developing Enterprise Web Services, S. Chatterjee, J. Webber, Pearson Education, 2008.
7. XML, Web Services, and the Data Revolution, F.P.Coyle, Pearson Education.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**

### DISTRIBUTED DATABASES
### (Core Elective-III)

**Objectives:**

- To introduce the student to the theory, algorithms, and methods that underlie distributed database management systems.
- The student should also acquire insight into distributed DBMS architecture, query decomposition and data localization, transaction management, and distributed concurrency control.

**UNIT I**

Features of Distributed versus Centralized Databases, Principles of Distributed Databases, Levels Of Distribution Transparency, Reference Architecture for Distributed Databases, Types of Data Fragmentation, Integrity Constraints in Distributed Databases, Distributed Database Design.

**UNIT II**

Translation of Global Queries to Fragment Queries, Equivalence transformations for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parametric Queries.
Optimization of Access Strategies, A Framework for Query Optimization, Join Queries, General Queries.

**UNIT III**

The Management of Distributed Transactions, A Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions.
Concurrency Control, Foundation of Distributed Concurrency Control, Distributed Deadlocks, Concurrency Control based on Timestamps, Optimistic Methods for Distributed Concurrency Control.

**UNIT IV**

Reliability, Basic Concepts, Nonblocking Commitment Protocols, Reliability and concurrency Control, Determining a Consistent View of the Network, Detection and Resolution of Inconsistency, Checkpoints and Cold Restart, Distributed Database Administration, Catalog Management in Distributed Databases, Authorization and Protection.

**UNIT V**

Architectural Issues, Alternative Client/Server Architectures, Cache Consistency, Object Management, Object Identifier Management, Pointer Swizzling, Object Migration, Distributed Object Storage, Object Query Processing, Object Query Processor Architectures, Query Processing Issues, Query Execution, Transaction Management, Transaction Management in Object DBMSs, Transactions as Objects.
Database Integration, Scheme Translation, Scheme Integration, Query Processing Query Processing Layers in Distributed Multi-DBMSs, Query Optimization Issues Transaction Management Transaction and Computation Model, Multidatabase Concurrency Control, Multidatabase Recovery, Object Orientation and Interoperability, Object Management Architecture CORBA and Database interoperability, Distributed Component Object Model, COM/OLE and Database Interoperability, PUSH-Based Technologies.

**TEXT BOOKS:**

1. Distributed Databases Principles & Systems, Stefano Ceri, Giuseppe Pelagatti, TMH.
2. Principles of Distributed Database Systems, M. Tamer Ozsu, Patrick Valduriez, Pearson Education, 2nd Edition.

**REFERENCE BOOKS:**

1. Distributed Database Systems, Chanda Ray, Pearson.
2. Distributed Database Management Systems, S.K.Rahimi and Frank.S.Haug, Wiley.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**

**INFORMATION RETRIEVAL SYSTEMS**
**(Core Elective-III)**

**Objectives:**
On completion of this course you should have gained a good understanding of the foundation concepts of information retrieval techniques and be able to apply these concepts into practice. Specifically, you should be able to:

- To use different information retrieval techniques in various application areas.
- To apply IR principles to locate relevant information from large collections of data.
- To analyze performance of retrieval systems when dealing with unmanaged data sources.
- To implement retrieval systems for web search tasks.

**UNIT I**
Boolean retrieval. The term vocabulary and postings lists. Dictionaries and tolerant retrieval. Index construction. Index compression.

**UNIT II**
Scoring, term weighting and the vector space model. Computing scores in a complete search system. Evaluation in information retrieval. Relevance feedback and query expansion.

**UNIT III**
XML retrieval. Probabilistic information retrieval. Language models for information retrieval. Text classification. Vector space classification.

**UNIT IV**
Support vector machines and machine learning on documents. Flat clustering. Hierarchical clustering. Matrix decompositions and latent semantic indexing.

**UNIT V**
Web search basics. Web crawling and indexes. Link analysis.

**TEXT BOOK:**
1. Introduction to Information Retrieval , Christopher D. Manning and Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2008.

**REFERENCE BOOKS:**
1. Information Storage and Retrieval Systems: Theory and Implementation, Kowalski, Gerald, Mark T Maybury, Springer.
2. Modern Information Retrival , Ricardo Baeza-Yates, Pearson Education, 2007.
3. Information Retrieval: Algorithms and Heuristics, David A Grossman and Ophir Frieder, 2nd Edition, Springer, 2004.
4. Information Retrieval Data Structures and Algorithms, William B Frakes, Ricardo Baeza-Yates, Pearson Education, 1992.
5. Information Storage & Retieval , Robert Korfhage , John Wiley & Sons.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**

### ETHICAL HACKING
### (Core Elective-IV)

**Objectives:**

- To learn about how systems vulnerabilities manifest themselves and why hackers continue to enjoy success breaking into systems, despite increasing attention paid to cyber defense.
- To gain experience with a systematic hacking methodology.
- To learn about and experiment with hacking tools that can be applied at different stages of the hacking process.

**UNIT I**

**Introduction to Ethical Hacking, Ethics, and Legality:**

Ethical Hacking Terminology, Different Types of Hacking Technologies, Different Phases Involved in Ethical Hacking and Stages of Ethical Hacking: Passive and Active Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks, Hacktivism, Types of Hacker Classes, Skills Required to Become an Ethical Hacker, Vulnerability Research, Ways to Conduct Ethical Hacking, Creating a Security Evaluation Plan, Types of Ethical Hacks, Testing Types, Ethical Hacking Report.

**Footprinting and Social Engineering:**

Footprinting , Information Gathering Methodology, Competitive Intelligence, DNS Enumeration   Who is and ARIN Lookups, Types of DNS Records, Traceroute, E-Mail Tracking, Web Spiders, Social Engineering, Common Types Of Attacks, Insider Attacks, Identity Theft, Phishing Attacks, Online Scams, URL Obfuscation, Social-Engineering Countermeasures.

**UNIT II**

**Scanning and Enumeration**

Scanning,  types of Scanning , CEH Scanning Methodology ,Ping Sweep Techniques, Nmap Command Switches, SYN, Stealth, XMAS, NULL, IDLE, and FIN Scans, TCP Communication Flag Types, War-Dialing Techniques, Banner Grabbing and OS Fingerprinting Techniques, Proxy Servers, Anonymizers , HTTP Tunneling Techniques, IP Spoofing Techniques, Enumeration, Null Sessions, SNMP Enumeration, Windows 2000 DNS Zone Transfer, Steps Involved in Performing Enumeration.

**System Hacking**

Understanding Password-Cracking Techniques, Understanding the LAN Manager Hash Cracking Windows 2000  Passwords, Redirecting the SMB Logon to the Attacker SMB Redirection, SMB Relay MITM Attacks and    Countermeasures NetBIOS DoS Attacks, Password-Cracking Countermeasures, Understanding Different Types of Passwords Passive Online Attacks, Active Online Attacks, Offline Attacks Nonelectronic Attacks, Understanding Keyloggers and Other Spyware Technologies Understand Escalating Privileges, Executing Applications, Buffer Overflows, Understanding Rootkits Planting Rootkits on Windows 2000 and XP Machines, Rootkit Embedded TCP/IP Stack  Rootkit Countermeasures, Understanding How to Hide Files, NTFS File Streaming  NTFS Stream Countermeasures, Understanding Steganography Technologies, Understanding How to Cover Your Tracks and Erase Evidence, Disabling Auditing, Clearing the Event Log.

**UNIT III**

**Trojans, Backdoors, Viruses, and Worms**

Trojans and Backdoors, Overt and Covert Channels, Types of Trojans, Reverse-Connecting Trojans, Netcat Trojan, Indications of a Trojan Attack, Wrapping, Trojan Construction Kit and Trojan Makers, Countermeasure Techniques in  preventing Trojans, Trojan-Evading Techniques, System File Verification Sub-objective to Trojan Countermeasures  Viruses and Worms, Difference between a Virus and a Worm ,Types of Viruses, Understand Antivirus Evasion Techniques, Understand Virus Detection Methods.

**Sniffers**

Protocols Susceptible to Sniffing, Active and Passive Sniffing, ARP Poisoning, Ethereal Capture and Display Filters, MAC Flooding, DNS Spoofing Techniques, Sniffing Countermeasures.

**Denial of Service and Session Hijacking**

Denial of Service, Types of DoS Attacks, DDoS Attacks, BOTs/BOTNETs, "Smurf" Attack, "SYN" Flooding, DoS/DDoS Countermeasures, Session Hijacking, Spoofing vs. Hijacking, Types of Session Hijacking, Sequence Prediction, Steps in Performing Session Hijacking, Prevention of Session Hijacking.

**UNIT IV**
**Hacking Web Servers, Web Application Vulnerabilities, and Web-Based Password Cracking Techniques**
Hacking Web Servers, Types of Web Server Vulnerabilities, Attacks against Web Servers, IIS Unicode Exploits, Patch Management Techniques, Web Server Hardening Methods Web Application Vulnerabilities, Objectives of Web Application Hacking, Anatomy of an Attack, Web Application Threats, Google Hacking, Web Application Countermeasures Web-Based Password Cracking Techniques, Authentication Types, Password Cracker, Password Attacks: Classification, Password-Cracking Countermeasures.
**SQL Injection and Buffer Overflows**
SQL Injection, Steps to Conduct SQL Injection, SQL Server Vulnerabilities, SQL Injection Countermeasures Buffer Overflows, Types of Buffer Overflows and Methods of Detection, Stack-Based Buffer Overflows, Buffer Overflow Mutation Techniques.

**UNIT V**
**Linux Hacking**
Linux Basics, Compile a Linux Kernel, GCC Compilation Commands, Install Linux Kernel Modules, Linux Hardening Methods.
**Penetration Testing Methodologies**
Security Assessments, Penetration Testing Methodologies, Penetration Testing Steps, Pen-Test Legal Framework, Automated Penetration Testing Tools, Pen-Test Deliverables.

**TEXT BOOK:**
1.  CEH official Certified Ethical Hacking Review Guide, Wiley India Edition.

**REFERENCE BOOKS:**
1.  Hacking Exponsed Web 2.0, by Rich Annings, Himanshu Dwivedi, Zane Lackey, Tata Mc Graw hill Edition.
2.  Ethical Hacking & Network Defense, Michael T. Simpson edition.
3.  Hacking Exposed Windows, Joel Scambray, cissp, Stuart Mcclure, Cissp, Third Edition, Tata Mc Graw hill edition.
4.  Hacking Exposed Window server 2003, Joel Scambray Stuart Mcclure, Tata Mc Graw hill edition.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**
## MULTIMEDIA AND RICH INTERNET APPLICATIONS
### (Core Elective-IV)

**Objectives:**
This course aims to further develop students' competency in producing dynamic and creative graphic solutions for multimedia productions. It provides students with the basic concepts and techniques of interactive authoring. It also introduces students with the advanced scripting skills necessary for implementing highly interactive, rich internet applications using multimedia technologies and authoring tools. Students will develop aesthetic value and competencies in multimedia authoring. Artistic visual style and layout design are stressed, as well as the editing and integration of graphic images, animation, video and audio files. The course allows students to master industry-wide software and technologies to create highly interactive, rich internet applications.

**UNIT  I**
Fundamental concepts in Text and Image: Multimedia and hypermedia, World Wide Web, overview of multimedia software tools. Graphics and image data representation graphics/image data types, file formats, Color in image and video: color science, color models in images, color models in video.

**UNIT  II**
Fundamental concepts in video and digital audio: Types of video signals, analog video, digital video, digitization of sound, MIDI, quantization and transmission of audio.
Multimedia Data Compression: Lossless compression algorithms, Lossy compression algorithms, Image compression standards.

**UNIT  III**
Basic Video compression techniques, Case study: MPEG Video Coding I, Basic Audio compression techniques, Case study: MPEG Audio compression.
**Web 2.0**
What is web 2.0, Search, Content Networks, User Generated Content, Blogging, Social Networking, Social Media, Tagging, Social Marking, Rich Internet Applications, Web Services, Mashups, Location Based Services, XML, RSS, Atom, JSON, and VoIP, Web 2.0 Monetization and Business Models, Future of the Web.

**UNIT  IV**
**Rich Internet Applications(RIAs)  with Adobe Flash** : Adobe Flash- Introduction, Flash Movie Development, Learning Flash with Hands-on Examples, Publish your flash movie, Creating special effects with Flash, Creating a website splash screen, action script, web sources.
**Rich Internet Applications (RIAs) with Flex 3** - Introduction, Developing with Flex 3, Working with Components, Advanced Component Development, Visual Effects and Multimedia,

**UNIT  V**
 **Ajax- Enabled Rich Internet Application :** Introduction, Traditional Web Applications vs Ajax Applications, Rich Internet Application with Ajax, History of Ajax, Raw Ajax example using xmlhttprequest object, Using XML, Creating a full scale Ajax Enabled application, Dojo ToolKit.

**TEXT BOOKS:**
1.      Fundamentals of Multimedia by Ze-Nian Li and Mark S. Drew PHI Learning, 2004
2.      Professional Adobe Flex 3, Joseph Balderson, Peter Ent, et al, Wrox Publications, Wiley India, 2009.
3.      AJAX, Rich Internet Applications, and Web Development for Programmers, Paul J Deitel and Harvey M Deitel, Deitel Developer Series, Pearson Education.

**REFERENCE BOOKS**:
1.	Multimedia Communications: Applications, Networks, Protocols and Standards, Fred Halsall, Pearson Education, 2001, rp 2005.
2.	 Multimedia Making it work, Tay Vaughan, 7th edition, TMH, 2008.
3.	Introduction to multimedia communications and Applications, Middleware, Networks, K.R.Rao, Zoran, Dragored, Wiley India, 2006, rp. 2009.
4.	Multimedia Computing, Communications & Applications, Ralf Steinmetz and Klara Nahrstedt, Pearson Education, 2004
5.	 Principles of Multimedia, Ranjan Parekh, TMH, 2006.
6.	Multimedia in Action, James E.Shuman, Cengage Learning, 198, rp 2008.
7.	Multimedia Systems design, Prabhat K. Andleigh, Kiran Thakrar, PHI, 1986.
8.	Multimedia and Communications Technology, Steve Heath, Elsevier, 1999, rp 2003.
9.	Adobe Flash CS3 Professional, Adobe press, Pearson Education, 2007.
10.	Flash CS3 Professional Advanced, Russel Chun, Pearson Education, 2007.
11.	 Flash CS5, Chris Grover, O'Reilly, SPD, 2010.
12.	SAMS Teach yourself Adobe flash CS3, Pearson Education, 2007.
13.	Flex 4 Cookbook, Joshua Noble, et.al, O'Reilly, SPD 2010.
14.	Flex3 – A beginner's guide, Michele E.Davis, Jon A. Phillips, TMH, 2008.
15.	Mastering Dojo, R.Gill, C.Riecke and A.Russell, SPD.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**

## SEMANTIC WEB AND SOCIAL NETWORKS
### (Core Elective-IV)

**Objectives:**

- To learn Web Intelligence.
- To learn Knowledge Representation for the Semantic Web.
- To learn Ontology Engineering.
- To learn Semantic Web Applications, Services and Technology.
- To learn Social Network Analysis and semantic web.

**UNIT I**
Thinking and Intelligent Web Applications, The Information Age, The World Wide Web, Limitations of Todays Web, The Next Generation Web
Machine Intelligence, Artificial Intelligence, Ontology, Inference engines, Software Agents, Berners-Lee www, Semantic Road Map, Logic on the semantic Web.

**UNIT II**
Ontologies and their role in the semantic web, Ontologies Languages for the Semantic Web –Resource Description Framework(RDF) / RDF Schema, Ontology Web Language(OWL),UML,XML/XML Schema.
Ontology Engineering, Constructing Ontology, Ontology Development Tools, Ontology Methods, Ontology Sharing and Merging, Ontology Libraries and Ontology Mapping.

**UNIT III**
Logic, Rule and Inference Engines. Semantic Web applications and services, Semantic Search, e-learning, Semantic Bioinformatics, Knowledge Base.

**UNIT IV**
XML Based Web Services, Creating an OWL-S Ontology for Web Services, Semantic Search Technology, Web Search Agents and Semantic Methods.
What is social Networks analysis, development of the social networks analysis, Electronic Sources for Network Analysis – Electronic Discussion networks.

**UNIT V**
Blogs and Online Communities, Web Based Networks. Building Semantic Web Applications with social network features.

**TEXT BOOKS:**
1. Thinking on the Web - Berners Lee, Godel and Turing, Wiley inter science, 2008.
2. Social Networks and the Semantic Web , Peter Mika, Springer, 2007.
**REFERENCE BOOKS:**

1. Semantic Web Technologies, Trends and Research in Ontology Based Systems, J.Davies, Rudi Studer, Paul Warren, John Wiley & Sons.
2. Semantic Web and Semantic Web Services -Liyang Lu Chapman and Hall/CRC Publishers,(Taylor & Francis Group)
3. Information sharing on the semantic Web - Heiner Stuckenschmidt; Frank Van Harmelen, Springer Publications.
4. Programming the Semantic Web, T.Segaran, C.Evans, J.Taylor, O'Reilly, SPD.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

**MCA - III Year - I Sem.**

## ANDROID MOBILE APPLICATION DEVELOPMENT LAB

The student is expected to be able to do the following problems, though not limited.

1. (a) Create an Android application that shows Hello + name of the user and run it on an emulator.
(b) Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.

2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.

3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.

4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.

5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.

6. Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message.

7. Create a user registration application that stores the user details in a database table.

8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.

10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.

11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.

12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

13. Create an application that shows the given URL (from a text field) in a browser.

14. Develop an application that shows the current location's latitude and longitude continuously as the device is moving (tracking).

15. Create an application that shows the current location on Google maps.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**MCA - III Year - I Sem.**

**SOFTWARE TESTING METHODOLOGIES AND UML LAB**

**i) Software Testing :**
**Objectives:**
1. To learn to use the following (or Similar) automated testing tools to automate testing:
a)         Win Runner/QTP for functional testing.
b)         LoadRunner for Load/Stress testing.
c)         Test Director for test management.
d)          JUnit, HTML Unit, CPP Unit.

**Sample problems on testing:**
1.         Write programs in 'C' Language to demonstrate the working of the following constructs:
i) do...while ii) while….do iii) if…else iv) switch v) for
2.         "A program written in 'C' language for Matrix Multiplication fails" Introspect the causes for its failure and write down the possible reasons for its failure.
3.         Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
4.         Write the test cases for any known application (e.g. Banking application)
5.         Create a test plan document for any application (e.g. Library Management System)
6.         Study of any testing tool (e.g. Win runner)
7.         Study of any web testing tool (e.g. Selenium)
8.         Study of any bug tracking tool (e.g. Bugzilla, bugbit)
9.         Study of any test management tool (e.g. Test Director)
10.         Study of any open source-testing tool (e.g. Test Link)
11.         Take a mini project (e.g. University admission, Placement Portal) and execute it.  During the Life cycle of the mini project create the various testing documents* and final test report document.

**Additional problems on testing:**
    1. Test the following using JUnit and CPPUnit:
i) Sorting problems  ii)Searching problems  iii)Finding gcd of two integers iv)Finding factorial of a number.

    2. Test web based forms using HTML Unit.

    3. Test database stored procedures using SQLUnit.
      (Use sufficient number of test cases in solving above Problems)

*Note: To create the various testing related documents refer to the text "Effective Software Testing Methodologies by William E. Perry".

  **ii) UML:**
**Objectives:**
- 	Understand how UML supports the entire OOAD process.
- 	Become familiar with all phases of OOAD.

**List of Sample Problems:**
1.  The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.

2.  Student has to take up another case study of his/her own interest and do the same whatever mentioned in first problem. Some of the ideas regarding case studies are given in reference books which were mentioned in theory syllabus can be referred for some idea.

**Additional Problems on UML:**
*Unified Modeling Language Lab:*

Students are divided into batches of 5 each and each batch has to draw the following diagrams using UML for an ATM system whose description is given below.

UML diagrams to be developed are:

1. Use Case Diagram.
2. Class Diagram.
3. Sequence Diagram.
4. Collaboration Diagram.
5. State Diagram
6. Activity Diagram.
7. Component Diagram
8. Deployment Diagram.
9. Test Design.

## Description for an ATM System

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

The ATM must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.

2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.

3. A customer must be able to make a transfer of money between any two accounts linked to the card.

4. A customer must be able to make a balance inquiry of any account linked to the card.

5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.

The ATM will provide the customer with a printed receipt for each successful transaction

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

**REFERENCE BOOKS:**

1. Software Testing Concepts and Tools, P.Nageswara Rao, dreamtech press.
2. Software Testing Tools, Dr.K.V.K.K.Prasad, dreamtech Press.
3. Software Testing with Visual Studio Team System 2008, S.Subashini, N.Satheesh kumar, SPD.
4. Learning UML 2.0, Russ Miles and Kim Hamilton, O'Reilly, SPD.
5. Mastering UML with Rational Rose, W.Boggs & M.Boggs, Wiley India.