MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Course Objectives
1. To provide the foundations on basic computer related concepts for a coherent development to the students for the courses like Fundamentals of Computer Organization, Data Structures, Design and Analysis of Algorithms, Computer Graphics and others.
2. To Comprehend different Properties of Binary Relations on a Set theory, Reflexivity, Symmetry, Transitivity, Graphical representation of symmetric relations, transitive relations, Hasse diagram and their applications apart from that they also learn topics like Monoid, semi groups, Groups, Semi group, Homomorphism and Isomorphism systems.
3. To Develop skills in understanding and applying basic concepts on Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial & Multinomial theorems, the principles of Inclusion – Exclusion along with their applications.
4. To Develop an appreciation for the use of Sequential functions and Calculating Coefficient of generating function, Characteristics roots, Solution of homogeneous Recurrence Relation.
5. To Design and Development of DFS, BFS and Spanning Trees, planar Graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers & their applications.

Course Outcomes
1. Able to demonstrate knowledge on the foundations of many mathematical computer related concepts.
2. Attained exposure to different Properties of Binary Relations subsequent to the course.
3. Ability to think logically and mathematically on topics like Basis of counting Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients etc.
4. Instill the belief that Mathematical logic is important for scientific research in Calculating Coefficient. Design and Development of Trees and Graphs & their applications.

Pre-requisites
1. Learner must have knowledge on Mathematics.
2. Should have well familiar with the concepts and terminology of Mathematical Concepts.
3. Learner should have a good understanding of Differential Equations.

UNIT-I

UNIT-II
Algebraic structures: Algebraic systems, Examples and general properties, Semi-groups and monoids, groups, and sub groups, homomorphism, Isomorphism.
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
Graph Theory: Representation of Graphs, DFS, BFS, Spanning Trees, Planar Graphs. Graph Theory and Applications, Basic Concepts, Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. Discrete Mathematical Structures with applications to computer science Trembly J.P. & Manohar . P, TMH

REFERENCE BOOKS:
3. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby, Sharon Cutler Ross, PHI/ Pearson Education.
6. Logic and Discrete Mathematics, Grass Man and Tremblay, Pearson Education
I Year I Semester

COMPUTER ORGANIZATION

Subject Code: L/T/P/C
3/1/0/4
Int :40, Ext:60, Total:100

Course Objectives

- A clear understanding of different set of functional units interconnected within the system with operations, interactions and communication.
- An in-depth knowledge of different number systems represented at machine level.
- Knowledge of various addressing modes to address the CPU.
- Understand about the I/O organization of various peripheral devices and the importance of DMA
- Explain the functioning and programming of the INTEL-8086 microprocessor using assembly language programming.
- Significance of memory hierarchy and details of cache memory.

Course Outcomes

After this course completion students should be able to know

- Details of the parts of a system and how data is stored and how it is manipulated using various circuits.
- I/O interaction in the system.
- Memory is organized.
- To write programs using assembly language.

Pre-requisites

- Fundamentals of computer system and digital electronics.

UNIT–I

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.

UNIT-IV
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-V
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.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
Gokaraju Rangaraju Institute Of Engineering And Technology(Autonomous)

I MCA I Semester
COMPUTER PROGRAMMING
Subject Code: L/T/P/C
3/1/0/4
Int :40, Ext:60, Total:100

Course Objectives

- To understand the various steps in Program development.
- To understand the basic concepts in C and C++ Programming Languages.
- Understand the concepts of functions, arrays, pointers, strings structures in C environment to solve structured problems
- Understand the concepts of classes, objects, constructers, inheritance, polymor
phism in C++ to solve object oriented problems
- Develop C, C++ application programs using sound POP and OOP practices

Course Outcomes

- Demonstrate the basic knowledge of computer hardware and software
- Ability to apply solving and logical skills to programming in C and C++ languages
- Should have the ability to extend his knowledge of C/C++ further on his/her own.

Pre-requisites

Student should have right aptitude, Logical Reasoning and Problem Solving ability.

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, Statements- Selection Statements ) – 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.

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

UNIT- III
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.
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- IV
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.
Different paradigms for problem solving need for OOP paradigm, classes and instances, fundamental characteristics of OOP, differences between OOP and Procedure Oriented Programming.
Introduction to C++ Basics, Structure of a C++ program, C++ Functions- Scope of variables, Parameter passing methods, Default arguments, inline functions, Recursive functions, C++ program examples.

UNIT-V
C++ Classes And Data Abstraction: Class definition, Objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction, ADT and information hiding. Dynamic memory allocation and de-allocation operators- new and delete operators.
Inheritance : 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, C++ program examples.
Polymorphism: Function overloading, Function Overriding. Virtual Functions, Base and Derived class virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, C++ program examples.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
I MCA

PROBABILITY AND STATISTICS

Subject Code: L/T/P/C
3/1/0/4
Int :40, Ext:60, Total:100

Course Objectives At the end of the course the student is expected to
1. Know the fundamentals of Probability and Statistics.
2. Understand and apply the Tests of Hypothesis, Correlation & Regression.
3. Understand simple Queuing models.

Course Outcomes Students will learn
1. Strong basics of Probability.
2. Applications of Probability distributions.
3. Application of regression analysis to analyse a problem.

Prerequisites
Fundamentals in Basic Mathematics.

UNIT I
Probability: Basic concepts in Probability - Conditional probability – Addition and Multiplication theorems for two events (Concepts and problem solving) – Baye’s theorem.
Random Variables: Definition of a random variable, discrete and continuous random variables – Distribution function and statements of its properties, probability mass function, probability density function with illustrations -Expectation and variance of a r.v and statements of their properties.

UNIT II
Distributions: Binomial, Poisson, Uniform, Normal and Exponential distributions (definition, real life examples, Statements of their Mean, Mode and Variance and problems). Fitting of Binomial and Poisson distributions. Sampling distribution: Definition of Population and sample, Overview of types of sampling(Purposive, Random, SRS with and without replacement cases, Stratified and Systematic random samplings) - Sampling distribution, standard error, statements of sampling distribution of mean(s) (Population variance(s) known and unknown) and proportion(s) (Population proportion(s) known and unknown) with examples.

UNIT III
Estimation & Testing of Hypothesis:Definitions of Point and Interval estimation. Confidence intervals for single mean, difference of two means, single proportion and difference of two proportions. Concepts of Null and Alternative hypotheses, Critical region, Type I and Type II errors, one tail and two-tail tests, Level of significance.
Large Samples Tests: Tests of hypothesis for mean(s) (single and difference between means), Tests of hypothesis for proportion(s) (single and difference between proportions), Chi-square test for testing goodness of fit, independence of attributes and single population variance.

UNIT IV
Small samples: Student’s t-test for testing the significance of single mean, difference of means( independent samples and paired samples), F-test for equality of variances (Concepts and problem solving).

UNIT V
Correlation & Regression: Fitting of straight line, second degree parabola and exponential curves by least squares method- Product momentcorrelation coefficient, test for its significance and statements
of properties, Spearman’s rank correlation coefficient and statement of its properties – Simple linear regression, Lines of Regression, Regression coefficients and statement of their properties, Multiple regression for three variables only.

**Teaching Methodologies**
1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**
2. Probability and Statistics, Dr.T.K.V.Iyengar,Dr.B.Krishna Gandhi et.al, S.Chand.

**REFERENCE BOOKS:**
Gokaraju Rangaraju Institute Of Engineering And Technology(Autonomous)

I MCA
ACCOUNTING AND FINANCIAL MANAGEMENT

I Semester

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
To provide necessary basic inputs and tools on maintaining books of accounts, preparation of financial statements, analysis and interpretation of financial statements. It is to provide the basic inputs to manage the finance functions.

Course Outcomes
After the completion of the course the students should be able to understand and prepare financial statements. Students should be able to understand the management of finances of the business organization.

Pre-requisites
Student should have Logical Reasoning and Problem Solving ability.

UNIT-I

UNIT-II

Unit-III

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.

UNIT-V
Capital Management : Components of working capital, gross vs. net working capital. Determinants of working capital needs, the operating cycle planning of WC. 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)

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
1. Prasanna Chandra, Financial Management, TMH, 2009
Course Objectives

Computer Organization Lab
- To provide a clear understanding of the instructions used in writing the program (8086 processor) for solving problems.
- To teach how to save, compile, execute a program.

IT Workshop-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.

Course Outcomes

Computer Organization
- Learner is aware of different commands usage and how the data is stored in different registers.
- Able to write programs using assembly language.

IT Workshop
- After this course completion students understand in depth of
  - The various internal parts of a computer and how the connection are given to these parts.
  - Able to perform assembling and dissembling of various parts of the system.
  - Able to install different OS softwares.
  - Able to explore on internet.
  - Able to make use of MS-Office is an efficient manner.

Pre-requisites

Computer Organization Lab
- Basic knowledge of the instructions of a microprocessor.
- Familiarity with the assembly language programming language.

IT Workshop
- Familiarity of computer hardware and software.
- Learner should have the basic knowledge of working on the system with MS-Office and Internet.

COMPUTER ORGANIZATION LAB

List of Sample Problems:
Write assembly language programs for the following using MASAM.

1. Write assembly language programs to evaluate the expressions:
   a) \( a = b + c - d * e \)
   b) \( z = x * y + w - v + u / k \)
   a. Considering 8-bit, 16 bit and 32 bit binary numbers as \( b, c, d, e \).
   b. Considering 2 digit, 4 digit 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, 4 digit 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 string 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.
10. A computer uses RAM chips of 1024 X 1 capacity.
    i. How many chips are needed, and how should their address lines be connected to provide an memory capacity of 1024 bytes?
    ii. 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 associativememory.
    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

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI.

IT WORKSHOP
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 effectively 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
Week 1 – 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.

Week 2 – 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.

Week 3 – 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.
Week 4 – 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.

Week 5 – 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.

Week 6 – 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
Week 7 - 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.

Week 8 - 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.

Week 9 - 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.

Week 10 - 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.

Week 11- 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
Week 12 – Word Orientation: The mentor needs to give an overview of LaTeX and Microsoft (MS) office2007/ 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.
Week 13 - 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.

Week 14 - 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
Week 15 - 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

Week 16 - 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

Week 17 - 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).

Week 18- 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

Week 19 - 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
7. PC Hardware and A+Handbook – Kate J. Chase PHI (Microsoft)
Course Objectives

- To learn basic algorithms and flowcharts both numeric and string concepts.
- To write simple C and C++ programs that incorporate different data types, different types of variables, expressions, selection, and iteration.
- To write structured programs using arrays, functions, strings, pointers, structures and unions.
- To write object oriented programs using classes, objects, constructors, functions using C++ syntax, inheritance and polymorphism concepts.

Course Outcomes

- Ability to write programs for different kinds of problems in C and C++.
- Work confidently in compilers like C and C++ compilers in various operating systems.
- Understand and find the output of simple C programs that incorporate different types of variables, expressions, selection, and iteration.
- Understand and find the output of more complex C programs containing arrays and invoking (calling) functions having input and output arguments using pointers.
- Understand and find the output of more complex C++ programs containing functions, classes and objects, constructors and destructors, inheritance and polymorphism.
- Understand different ways of memory allocations and deallocation mechanisms using C and C++.
- Understand and find the output of different file handling mechanisms using C and C++.

Pre-requisites

- Fundamentals of editing, compiling, executing using editors.

Programs using C language

1. Write a C Program to Print a Sentence
2. Write a C Program to Add Two Integers
3. Write a C Program to Find Quotient and Remainder of Two Integers Entered by User
4. Write a C Program to Swap Two Numbers
5. Write a C Program to Check Whether a Number is Even or Odd
6. Write a C Program to Check Vowel or Consonant
7. Write a C Program to Find the Largest Number Among Three Numbers
8. Write a C program to Find all Roots of a Quadratic equation
9. Write a C Program to Check Leap Year
10. Write a C Program to Check Whether a Number is Positive or Negative or Zero.
11. Write a C Program to Check Whether a Character is an Alphabet or not
12. Write a C Program to Calculate Sum of Natural Numbers
13. Write a C Program to Find Factorial of a Number
14. Write a C program to Generate Multiplication Table
15. Write a C Program to Display Fibonacci Series
16. Write a C Program to Find HCF of two Numbers
17. Write a C Program to Find LCM of two Numbers
18. Write a C Program to Count Number of Digits of an Integer
19. Write a C Program to Reverse a Number
20. Write a C program to Calculate the Power of a Number
21. Write a C Program to Check Whether a Number is Palindrome or Not
22. Write a C Program to Check Whether a Number is Prime or Not
23. Write a C Program to Display Prime Numbers Between Two Intervals
24. Write a C program to Check Armstrong Number
25. Write a C Program to Display Armstrong Number Between Two Intervals
26. Write a C program to Display Factors of a Number
27. Write a C program to Make a Simple Calculator to Add, Subtract, Multiply or Divide Using switch...case
28. Write a C Program to Display Prime Numbers Between Intervals by Making Function
29. Write a C Program to Check Prime and Armstrong Number by Making Function
30. Write a C program to Check Whether a Number can be Express as Sum of Two Prime Numbers
31. Write a C program to Find Sum of Natural Numbers using Recursion.
32. Write a C program to Calculate Factorial of a Number Using Recursion
33. Write a C Program to Find H.C.F Using Recursion
34. Write a C program to Reverse a Sentence Using Recursion
35. Write a C program to Calculate the Power of a Number Using Recursion
36. Write a C Program to Convert Binary Number to Decimal and Decimal to Binary
37. Write a C Program to Calculate Average Using Arrays
38. Write a C Program to Find Largest Element of an Array
39. Write a C Program to Add Two Matrix Using Multi-dimensional Arrays
40. Write a C Program to Multiply to Matrix Using Multi-dimensional Arrays
41. Write a C Program to Find Transpose of a Matrix
42. Write a C Program to Multiply two Matrices by Passing Matrix to Function
43. Write a C Program to Sort Elements of an Array
44. Write a C Program to Access Elements of an Array Using Pointer
45. Write a C Program to Find Largest Number Using Dynamic Memory Allocation
46. Write a C Program to Find the Frequency of Characters in a String
47. Write a C Program to Find the Number of Vowels, Consonants, Digits and White space in a String
48. Write a C Program to Remove all Characters in a String Except Alphabet
49. Write a C Program to Reverse a String by Passing it to Function
50. Write a C Program to Find the Length of a String
51. Write a C program that counts the characters, lines and words in the text file.
52. Write a C program to Concatenate Two Strings
53. Write a C Program to Copy a String
54. Write a C Program to Sort Elements in Lexicographical Order (Dictionary Order)
55. Write a C Program to Store Information (name, roll and marks) of a Student Using Structure
56. Write a C Program to Add Two Distances (in inch-feet) System Using Structures
57. Write a C Program to Add Two Complex Numbers by Passing Structure to a Function
58. Write a C Program to Calculate Difference Between Two Time Period
58. Write a C Program to Store Information of 10 Students Using Structure
59. Write a C Program to Store Information Using Structures for n Elements Dynamically
60. Write a C++ program to display the contents of a text file.
61. Write a C Program to Copy the contents from one file to another.
62. Write a C program that counts the characters, lines and words in the text file.

Programs using C++ language
1. Write a C++ Program to implement a Class STUDENT having following members:
   Data members as Student number, name, marks in three subjects.
   Member functions as computeTotal and computeAverage and display the data.
2. Write a C++ Program to Design a Class for a Bank having following members:
   Data members as acno, acname, balance. Member functions as getdata(), showdata(), deposit(), withdraw(). Use constructors & destructors whenever necessary.
3. Write a C++ program to implement copy constructor mechanism. Use default and parameterized constructors to initialize data members of a class.
4. Write a C++ Program to demonstrate the Function Overloading.
5. Write a C++ Program to demonstrate Friend Function and Friend Class.
6. Write a C++ Program to demonstrate the use of Static data members and Static member functions.
7. Write a C++ Program to demonstrate the concepts of callby value, call by reference mechanisms.
8. Write C++ programs that illustrate how the following forms of inheritance are supported:
   a) Single inheritance   b) Multiple inheritance   c) Multi level inheritance
9. Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions.
10. Write a C++ program that illustrates the role of virtual base class in building class hierarchy.

Teaching methodologies:
- Lab experiments with GCC compiler on Linux/Turbo C & C++ on Dev C++ on Windows O.S.
- Discussion on white board and observation books.

Text Books:
  a.  Computer Programming in C, V. Rajaraman, PHI
  b.  C Programming, E.Balagurusamy, 3rd edition, TMH.
  d.  Object Oriented programming with C++, E.Balagurusamy, 4th edition, TMH.
I MCA I Semester
English Language and Communication Skills

Subject Code: L/T/P/C
3/1/0/4
Int :40, Ext:60, Total:100

Course Objectives
The course covers verbal, non-verbal, written and other forms of communication. The objectives of the course are to make the students familiar with the premises and principles of different ways and forms of communication in a business context and sharpen their communication skills.

Pre-requisites
- Should be familiar with spoken English and written English
- Should have heard lectures in English previously

UNIT- I
Features of Indian English:
Correction of sentences – Informal conversation Vs Formal expression – Verbal and non-verbal communication, barriers to effective communication – kinesics – Types of communication – Listening, Speaking, Reading and Writing, Telephone etiquette.

UNIT- II
Non-Verbal and Intercultural Communication
Importance of non-verbal communication - personal appearance - facial expressions- movement-posture – gestures - eye contact –voice - beliefs and customs- worldview and attitude.

UNIT -III
Oral Communication
Listening - types and barriers to listening - speaking - planning and audience awareness - persuasion-goals - motivation and hierarchy of needs - attending and conducting interviews participating in discussions, debates - and conferences - presentation skills- paralinguistic features -fluency development strategies

UNIT - IV
Official Correspondence

UNIT- V
Project Proposal and Reports

REFERENCE BOOKS
Gokaraju Rangaraju Institute Of Engineering And Technology(Autonomous)

I MCA II Semester

DATA STRUCTURES AND ALGORITHM ANALYSIS

Subject Code: L/T/P/C
3/1/0/4
Int :40, Ext:60, Total:100

Course Objectives

- To understand the basic concepts of Linear and Non Linear Data structures.
- To understand the notations and computation of Performance analysis of computer algorithms.
- To understand the representation and applications of data structures such as stacks, queues, trees, and Graphs.
- To understand and analyze various searching and sorting algorithms.

Course Outcomes

- Capable of analysing the time and space complexity for a given algorithm. And find ways to improve the efficiency of algorithms.
- Ability to identify appropriate data structure for solving given problem
- Able to apply suitable design strategies (greedy, divide and conquer etc.) for a given problem.
- Facilitates learner to comprehend these concepts in various courses of this programme.

Pre-requisites

1. Learner must have completed a course on Computer Organization and Understand the terminology of computer.
2. Learner should have a good understanding of Computer programming and its concepts preferably “C” language.

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


Stack ADT: definitions, operations, array and linked implementations, applications infix to postfix conversion, recursion implementation

Queue ADT: definitions and operations, array and linked Implementations, Dequeue.

UNIT-II


UNIT-III

AVL Trees: Definition, Operations – Insertion and Searching.

B-Trees: Definition, B-Tree of order m, operations- insertion and deletion, Disjoint Sets- Union and Find algorithms

Graphs: Introduction, Basic Terminology, Graph Representations- Adjacency matrix, Adjacency lists, Adjacency multi-lists, Graph traversals- DFS and BFS.

UNIT-IV

Searching- Linear Search, Binary Search, Hashing-Introduction, hash tables, hash functions, collision resolution methods, Comparison of Searching methods.

UNIT-V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS :

REFERENCE BOOKS:
3. Data Structures with Java, J.R.Hubbard and A.Huray, Pearson Education/PHI.
I MCA  

JAVA PROGRAMMING  

Subject Code: L/T/P/C  
3/1/0/4  
Int :40, Ext:60, Total:100

Course Objectives
- Understand the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, and encapsulation.
- 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.
- Develop programs using Multithreading and Networking for operating system based and distributed based applications.

Course Outcomes
- Understanding of OOP concepts and basics of java programming (Console and GUI based)
- Describe the use of packages and interfaces
- Demonstrate understanding of exception handling
- Demonstrate understanding of Java’s byte and character streams for input and output
- Demonstrate understanding of multithreaded programming
- Demonstrate understanding of applets, events and keywords.
- Should have the ability to extend his knowledge of Java programming further on his/her own.

Pre-requisite
- Students should have basic computer literacy including the use of a text editor, be familiar with the Internet, with the World Wide Web.
- Experience with directory structures, saving and retrieving files, and web browsers.
- Should have basic knowledge of C & C++

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 concepts, benefits of inheritance ,Super classes and Subclasses, 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: Uses of inner classes, local inner classes, anonymous inner-classes, static inner classes, examples.

UNIT-III

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

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.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
5. Starting out with Java, T. Gaddis, Dreamtech India Pvt. Ltd.
10. Maurach’s Beginning Java2, D. Lowe, J. Murach, A. Steelman, SPD.
11. Programming with Java, M. P. Bhave, S. A. Patekar, Pearson Education
I MCA II Semester
ORGANIZATION STRUCTURE AND PERSONNEL MANAGEMENT

Subject Code: L/T/P/C
        4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
• To understand basic management concepts and use of management principles in the organization
• To understand function of management and organization structure.
• To acquire the knowledge of quality management practices.
• This course also covers Personnel Management and Communication.
• To understand the organizational behaviour.

Course Outcomes:
Upon completion of the course, the students should be able to:
• 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.
• 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.

Pre-requisites
• Familiarity with Organizational hierarchy involved in some organization.
• Familiarity with organization functions.
• Familiar with some principles involved in interpersonal relationships in day to day life.

UNIT-I

UNIT-II
Personnel Management-I: Evolution, objectives, Personnel policies, Personnel Management vs HRM, Position of the personnel department in the organization. Role of Personnel Manager as Line manager and Staff Manager. Manpower planning, recruitment and selection, interviewing techniques, transfers, promotion and its policies.

UNIT-III
Personnel Management-II: Training and development: Objectives and policies planning, organizing the training department, training manager and his job, Methods of on- and off- the- job training, career planning, objectives and methods of performance appraisal.

UNIT-IV
UNIT-V
Contemporary Strategic Issues: Total Quality Management (TQM), Six sigma and People Capability Maturity Model (PCMM) Levels, Performance Management, Business Process outsourcing (BPO), Business Process Re-engineering and Bench Marking, Balanced Score Card.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
I MCA

OPERATIONS RESEARCH

II Semester

Subject Code: L/T/P/C

4/0/0/4

Int :40, Ext:60, Total:100

Course Objectives  At the end of the course the student is expected to

- Formulate mathematical models to real world problems.
- Understand and apply routing problems.
- Understand simple Queuing models.

Course Outcomes  Students will learn

- Strong basics of optimization techniques.
- Basics of LPP models.
- Applications of replacement models.
- Applications of queuing models.

Prerequisites

Fundamentals of mathematics

Unit I

Introduction to Operations Research: Definition, scope, objectives, phases, models and limitations of Operations Research.


Unit II


Sequencing models: Solution of sequencing problem – Johnson’s Algorithm for processing n jobs through two machines, processing n jobs through 3 machines-processing 2 jobs through m machines, processing n jobs through m machines.

Unit III

Replacement Models: Replacement of items that deteriorate whose maintenance cost increase with time with and without change in the money value. Replacement of items that fail suddenly: individual replacement, group replacement.

Dynamic Programming: Characteristics of Dynamic programming, Bellman’s principle of optimality. Dynamic programming approach for stage coach and shortest path.

Game theory: competitive games, Two-person zero sum game, rectangular game, saddle point, min-max (max-min) method of optimal strategies, value of the game. Solution of games with saddle point, Dominance principle. Rectangular games without saddle point – mixed strategy for 2X2 games. Solution of 2Xn, mX2 games by Graphical method.

Unit IV

Necessity of maintaining inventory, inventory costs and inventory control problem, selective inventory management techniques- ABC analysis, VED analysis, EOQ, Inventory models: 1. Classical EOQ model (Demand rate uniform, replenishment rate infinite.)

2. Demand rate uniform, production rate finite, shortages allowed.

3. (a) instantaneous models with probabilistic models.
4. Instantaneous demand, no setup cost, stock levels, continuous, lead time is zero

**Unit V**

**Queuing theory:** Queue description, characteristics of a queuing model, Poisson process, concept of Birth and death process, steady state solutions of \((M/M/1: \infty/FIFO)\) and \((M/M/1: N/FIFO)\).

**Teaching Methodologies:**
1. Chalk & Talk
2. Ppt's

**TEXT BOOKS:**
2. N.D. Vohra “Quantitative Techniques in Management “TATA MCGRAW HILL publications.

**REFERENCE BOOKS:**
3. P. Shankaralyer “Operations Research” TATA MCGRAW HILL
Course Objectives

- Clear understanding of the concepts of OS, its components, types and their working.
- Clear insight of the scheduling policies of OS.
- Understand process concurrency and synchronization.
- A thorough understanding of various memory management techniques.
- Clear insight of secondary storage devices and file management.
- To understand the operations performed by OS as a resource manager.
- Understand the various protection techniques provided by OS.
- To study different OS and compare their features.

Course Outcomes

The course will help each student to better:

- Understand how the OS manages all the processes and handles the different IO operations.
- Understand process coordination.
- Learn to minimize turnaround time, waiting time and response time and also maximization of throughput keeping CPU as busy as possible.
- Appreciate the nuances of different allocation techniques.
- The importance of memory management using different techniques.
- Applying optimization techniques for the improvement of system performance.
- Ability to understand the synchronous and asynchronous communication mechanisms in their respective OS.
- Understand how the OS is keeping a track of the resources among all processes without the occurrence of deadlock with different algorithms.
- Understand how protection is provided by OS.
- Ability to compare different Operating Systems.

Pre-requisites

- Basic knowledge of computer hardware and software.
- Knowledge of some of the data structure concepts.
- Familiarity of at least one operating system.

UNIT-I


UNIT-II

UNIT-III

UNIT-IV

UNIT-V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
Course Objectives

- To write and execute programs using arrays and Linked List to implement data structures linked.
- To write and execute write programs in C to implement various sorting and searching methods.

Course Outcomes

- Able to understand and identify the appropriate data structure for given problem.
- Ability to design and analyze the time and space complexity of algorithm or program.
- Ability to effectively use compilers includes library functions, debuggers and trouble shooting.

Pre-requisites

1. Learner must understand the terminology of computer.
2. Should be able to code simpler programs using a programming Language preferably “C” language.

List of Sample Problems/Experiments:

1. Write a program to perform the following operations:
   a) Create a SLL of integers.
   b) Delete an integer from SLL.
2. Write a program to perform the following operations:
   a) Create a DLL of integers.
   b) Delete an integer from DLL.
3. Write programs to implement the following using an array.
   a) Stack ADT
   b) Queue ADT
4. Write programs to implement the following using a singly linked list.
   a) Stack ADT
   b) Queue ADT
5. Write a program to convert a given infix expression into postfix form using stack.
6. Write programs to implement the deque (double ended queue) ADT using DLL and an array.
7. Write programs that use recursive functions to traverse the given binary tree in
   a) Preorder
   b) inorder and
   c) postorder
8. Write programs that use non-recursive functions to traverse the given binary tree in
   a) Preorder
   b) inorder and
   c) postorder
9. Write a program to perform the following operations:
   Insert an element into a BST
   Delete an element from a BST
   Search for a key element in a BST
10. Write programs to Implement BFS and DFS of a given graph.
11. Write programs to Implement Linear and Binary Search methods.
12. Write programs to implement the following Sorting methods
    a) Bubble Sort
    b) Selection Sort
    c) Insertion Sort
    d) Merge Sort
    e) Quick Sort
    f) Heap Sort
13. Write a program to insert and search an element from an AVL-tree.
14. Write a program to insert and delete an element from B-tree
15. Write a program to implement all the functions of a dictionary (ADT) using hashing.
16. Write a program for generating Minimum cost spanning tree using Kruskal’s algorithm

Teaching Methodologies

1. Program Demonstrations using Power Point Presentations
2. Whiteboard and Marker Pen

**TEXT BOOKS :**
3. Data Structures with Java, J.R.Hubbard and A.Huray, Pearson Education/ PHI.
I MCA

JAVA PROGRAMMING LAB

Course Objectives

- To learn to write, compiling & execute basic java program.
- To learn the use of data types & variables, decision control structures: if, nested if etc.
- To learn the use loop control structures: do, while, for etc.
- To create classes and objects and use them in their program.
- To will learn the use oop concept i.e data abstraction & data hiding, encapsulation, inheritance, polymorphism.
- To create and use threads, handle exceptions and write applets, awt, swing.
- To learn the use of interfaces and inner classes, wrapper classes, generics.

Course 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.
- Implementation of logic using java lab
- Able to write reusable function/utilities.
- The student will be able to write a Java program that demonstrates a class hierarchy, inheritance, and a simple degree of polymorphism.
- The student will be able to create files and read from computer files using Java.
- The student will be able to create exception handling in Java programs
- The student will be able to write a program that demonstrates the use of static variables and interfaces in object-oriented programming

Pre-requisites

- Basic OOP concept.
- Basic knowledge of C/C++ language.

List of Sample Problems/Experiments

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 first 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 \( n \)th 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 display 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.
   b. Printing a matrix.
c. Addition of matrices.

d. Subtraction of matrices.

8. Write a Java Program to solve Tower’s 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 complexADT 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
    b) Writing a complex number.
    c) Addition of Complex numbers
    d) Subtraction of Complex numbers
    e) Multiplication of complex numbers.
    f) Division 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.

1. 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, 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. 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.

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

18. Write a Java program for handling mouse events.

19. 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.

20. 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. when the Divide button is clicked. If Num1 or Num2 were not an 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.

21. 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)
22. 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.

23. 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, all of which extend the class Shape. Each one of the classes contains the only method numberOfSides() which display the number of sides in the given geometrical figures.
   b) A table named Table.txt stored in a text file has the first line as header and the remaining lines correspond to rows in table. The elements are separated by comma. Write a Java program to display the table using JTable component.

24. Write a Java program that illustrates the following
   a. Creation of simple package.  
   b. Accessing a package.  
   c. Implementing interfaces.
25. Write Java programs that illustrates the following
   a) Handling predefined exceptions
   b) Handling user defined exceptions
26. Write a Java program for handling Key events
I MCA                    SOFT SKILLS PRACTICE                    II Semester

Subject Code:             L/T/P/C                          Int :40, Ext:60, Total:100
                         3/1/0/4

Course Objectives
To enable students to
1. Enhance personality accomplished in behaviour and attitude.
2. Expertise in cooperation and leadership when working with a team.
3. Enhance skills in appearing for any type of interview.
4. Realize the EGO states in interactions and know to approach to think laterally.
5. Learn to face public with assertive approach and use aids to effectively enhance in such presentations.

Pre-requisites
1. Should have gone through a course on English language communication.
2. Should be eager to improve one’s personality for facing panels

UNIT- I
Personal Growth and Group Development
Awareness of self, Emotional Intelligence, Body language, Mannerism, Telephone Etiquette, Goal setting, grooming, Role playing.

UNIT- II
Team Building Conflict Resolution
Stages of team development, Teamwork for innovation and change, leadership Acclimatization of Organizational climate, Adapting with new Organizational change, Continuous learning

UNIT -III
Interview Skills
Technical interview sample questions with real-time experience, HR and Managerial Interview, Stress Interview, Interview etiquettes

UNIT- IV
Development of Critical and Reflective Thinking
Understanding thinking process using NLP, Out of box thinking, Ego states and transactions.

UNIT-V
Presentation Skill Development
Public speaking, Audience awareness, Pre-preparation of presentation, Developing a presentation, Usage of audio-visual aids

REFERENCES:
• Wallance, Masters, Personality Development, Cengage Learning, 2009
• Abrams Business Plan in a Day, PHI Learning, 2009
• Mohan, Developing Communication Skills, Macmillan, 2010
• Hurlock, Personality Development, Tata McGraw Hill, 2010
• Bhatti, The Dynamics of Personality, Pearson Education, 2010
Course Objectives
- To familiarize students with Linux Environment
- To learn fundamentals of Shell programming
- To develop a deeper understanding of operating system functions.
- To familiarize students with implementation details of Operating Systems
- To simulate various basic concepts of Operating System Functions using system calls and library functions.

Course Outcomes
- Should be able to work confidently in Linux/Unix Environment.
- Should be able to write shell scripts to automate various tasks.
- Should be able to use system calls for interacting with File System.
- Should be able to use System calls to Create Process, Handle Signals.
- Should be able to perform Inter Process Communication and Network Programming.

Pre-requisites
- Learner must have completed a course on Computer Organization and Operating Systems.
- Should be well familiar with the concepts and terminology of Operating Systems.
- Learner should have a good understanding of a programming language preferably “C” language.

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.

UNIT - II
Working with the Bourne again shell(bash): Introduction, shell responsibilities, pipes and input Redirection, output redirection, 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.

UNIT - III
Linux Files: File Concept, File System Structure, Inodes, File types, The standard I/O (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputs, putc, putchar, fgets, gets etc.), formatted I/O, stream errors, kernel support for files, System calls, library functions, file descriptors, low level file access - usage of open, creat, read, write, close, lseek, stat family, umask, dup, dup2, fcntl, file and record locking. File and directory management - chmod, chown, links(soft links & hard links - unlink, link, symlink), mkdir, rmdir, chdir, getcwd, Scanning Directories-opendir, readdir, closedir, rewinddir, seekdir, telldir functions.

UNIT-IV
UNIT-V
Interprocess Process Communication: Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes, FIFOs, Introduction to three types of IPC (Linux)-message queues, semaphores and shared memory. Message Queues- Kernel support for messages, Linux APIs for messages, client/server example. Semaphores-Kernel support for semaphores, Linux APIs for semaphores, file locking with semaphores. Shared Memory- Kernel support for shared memory, Linux APIs for shared memory, semaphore and shared memory example.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. Unix System Programming using C++, T.Chan, PHI.

REFERENCE BOOKS:
2. Unix Network Programming, W.R.Stevens, PHI.
II MCA I Semester

DATA COMMUNICATION AND COMPUTER NETWORKS

Subject Code: L/T/P/C
0/0/6/2
Int:40, Ext:60, Total:100

Course Objectives
- To introduce the fundamental various types of computer networks and their applications.
- To understand the principles involved in different layers with details of their involvement in packet transfer.
- To understand the relationship between the underlying transmission network and the specifications in the packet.
- To introduce UDP and TCP Models with the constraints in its implementation.

Course Outcomes
- After this course students should get a clear idea of the details of the various Protocols and the frames involved.
- After this course he/she will be able to appreciate the different aspects involved in transfer of data in a computer network by which if he/she uses tools, he/she can easily assimilate internal principles involved.

Pre-requisites
- Should have learnt concepts of operating systems.
- Have some real-time exposure to usage of Computer Network based application or should have been introduced to laboratory with Network connections

UNIT-I

UNIT-II
Data link layer: Introduction, Block coding, cyclic codes, checksum, framing, flow and error control, Noiseless channels, noisy channels, HDLC, point to point protocols Medium Access sub layer: Random access, controlled access, channelization, IEEE standards, Ethernet, Fast Ethernet, Giga-Bit Ethernet, wireless LANs.

UNIT-III
Connecting LANs, backbone networks and virtual LANs, Wireless WANs, SONET, frame relay and ATM. Network Layer: Logical addressing, internetworking, tunneling, address mapping, ICMP, IGMP, forwarding, uni-cast routing protocols, multicast routing protocols.

UNIT-IV
Transport Layer: Process to process delivery, UDP and TCP protocols, SCTP, data traffic, congestion, congestion control, QoS, integrated services, differentiated services, QoS in switched networks.

UNIT-V
Application Layer: Domain name space, DNS in internet, electronic mail, FTP, WWW, HTTP, SNMP, TELNET, SSH,C

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
3. Computer and Communication Networks Nader F. Mir, Person Education
II MCA  MANAGEMENT INFORMATION SYSTEMS  I Semester

Subject Code:  L/T/P/C  4/0/0/4  Int :40, Ext:60, Total:100

Course 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 organization within the company, Managers view of Information systems,Contemporary Approaches to Information Systems,How Information Systems Impact Organizations and Business Firms.

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. Organising data and information: Datawarehouses, Datamart and datamining

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 personal, computer related acquisition, develop forms for data collection and information dissemination, develop the files, test the system, cut over, 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. Introduction to E-Commerce and E-commerce challenges. A overview of
ERP, Applications of information systems to business. Security and ethical issues of information systems.

**Teaching Methodologies**

1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**


**REFERENCE BOOKS:**

Gokaraju Rangaraju Institute Of Engineering And Technology(Autonomous)

II MCA

SOFTWARE ENGINEERING

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
To Know what is a Software and Software Engineering and how it is playing a vital role in various applications. Software Engineering is the core for today’s Software implementation. Also to know the fields which are using Software Engineering applications.

Course Outcomes
After reading Software Engineering, a student is able to know the difference between Manual and Software oriented applications as per the working conditions. Student is also able to understand the implementation of the Software Engineering Process.

Pre-requisites
Student should have the knowledge of basic engineering principles which specifies the systematic manufacturing of a product/service.
He/she should also know the Re-Engineering Process.

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

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


Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
3. Fundamentals of Software Engineering,Rajib Mall,PHI, 2005
Course Objectives
The objective of the course is to learn the foundation of Database and Database Management Systems. In this course, students will learn to design a database, removing the redundancies using Functional Dependencies and Normalization. Students also learn Querying, Improve the database performance by Indexing, Hashing and finally optimize the queries, Transaction Management and Memory management.

Course Outcomes
Demonstrate the basic elements of a data models such as ER and relational data models. 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. Ability to understand Transaction management and memory Management with relevant problems.

Pre-requisites
- Experience with programming in some common programming language.
- A good understanding of data structures and algorithms.

UNIT-I

UNIT-II

UNIT-III

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 Dead Locks, 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**

**Storing data - Disks and Files:** -The Memory Hierarchy – Redundant Arrays of Independent Disks.  
**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. **Tree Structured Indexing:** Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM) B+ Trees: A Dynamic Index Structure, Search, Insert, Delete. **Hash Based Indexing:** Static Hashing, Extendable hashing, Linear Hashing, Extendible vs. Linear Hashing.

**Teaching Methodologies**

1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**


**REFERENCE BOOKS:**

1. Introduction to Database Systems,C.J.Date, Pearson Education.  
2. Database Management System Oracle SQL and PL/SQL,P.K.Das Gupta,PHI.  
I MCA II Semester

ADVANCED UNIX PROGRAMMING LAB

Subject Code: L/T/P/C
0/0/6/4
Int :40, Ext:60, Total:100

Course Objectives
1. To familiarize students with Linux Environment
2. To learn fundamentals of Shell programming
3. To develop a deeper understanding of operating system functions.
4. To familiarize students with implementation details of Operating Systems

Course Outcomes
1. Should be able to work confidently in Linux/Unix Environment.
2. Should be able to write shell scripts to automate various tasks.
3. Should be able to use system calls for interacting with File System.
4. Should be able to use System calls to Create Process, Handle Signals.

Pre-requisites
1. Learner must have completed a course on Computer Organization and Operating Systems.
2. Should be well familiar with the concepts and terminology of Operating Systems.
3. Learner should have a good understanding of a programming language preferably “C” language.

List of Sample Problems/Experiments
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 to find factorial of a given integer.
6. Write a c program that makes a copy of a file using standard I/O and system calls.
7. Implement in C the following Unix commands using System calls
   A. cat          B. ls         C. mv
8. Write a program that takes one or more file/directory names as command line input and reports the following information on the file.
   A. File type.
   B. Number of links.
   C. Time of Last Access
   D. Read Write and execute permissions
9. Write a C program to list for every file in a directory, its inode number and file name.
10. Write a C program to emulate the Unix ls –l command.
11. Write a C program that demonstrates redirection of standard output to a file.Ex: ls > f1.
12. Write a C program that illustrates the uses of the mkdir, opendir, readdir, closedir and rmdir API’s.
13. 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.
14. Write a C program to create a Zombie process.
15. Write a C program that illustrates how an orphan is created.
16. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- `ls -l | sort`

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

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

19. Simulate the following CPU Scheduling algorithms
   a. Round Robin
   b. SJF
   c. FCFS
   d. Priority

20. Simulate the following page replacement algorithms
   a. FIFO
   b. LRU
   c. LFU

Teaching Methodologies
   1. White Board and Marker
   2. Power Point Presentations
   3. Demonstration using LCD

TEXT BOOKS:
II MCA  I Semester

DATA BASE MANAGEMENT SYSTEMS LAB

Subject Code: L/T/P/C
0/0/6/2
Int :40, Ext:60, Total:100

Course Objectives
- This lab enables the students to practice the concepts learnt in the subject Database management systems.

Course Outcomes
- Ability to design and implement a database schema for given problem.
- Be capable to Design and build a GUI application.
- Apply the normalization techniques for development of application software to realistic problems.
- Ability to formulate queries using SQL DML/DDL/DCL commands.

Pre-requisites
- Experience with programming in some common programming language.
- A good understanding of data structures and algorithms.

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
- Ticketing
- Cancellations

Reservations:
Reservations are directly handled by booking office. Reservations can be made 60 days in advance in either cash or credit. In case the ticket is not available, a wait listed ticket is issued to the customer. This ticket is confirmed against the cancellation.

Cancellation and Modifications:
Cancellations are also directly handed at the booking office. Cancellation charges will be charged. 
Wait listed tickets that do not get confirmed are fully refunded.

Week1: 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

PRIMARY KEY ATTRIBUTES:
1. Ticket ID (Ticket Entity)
2. Passport ID (Passenger Entity)

Apart from the above mentioned entities you can identify more. The above mentioned are few.

Week2: 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

Week 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, Multivalued, 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Passport ID</th>
</tr>
</thead>
</table>

Week 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.

Week 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 table:
```
CREATE TABLE Passenger (Passport id INTEGER PRIMARY KEY, Name CHAR (50) NULL,Age Integer,Sex Char);
```

Week 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,'hyderabad','Banglore');

**Inserting values into Bus table:**
Insert into Passenger values (1,45,'ramesh', 45,'M','abc123');
Insert into Passenger values (2,78,'geetha', 36,'F','abc124');

**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;

**Week7: 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. Display the source and destination having journey time more than 10 hours.
5. Find the ticket numbers of the passengers whose name start with ‘A’ and ends with ‘H’.
6. Find the names of passengers whose age is between 30 and 45.
7. Display all the passengers names beginning with ‘A’
8. Display the sorted list of passengers names
9. Display the Bus numbers that travel on Sunday and Wednesday
10. Display the details of passengers who are traveling either in AC or NON_AC (Using only IN operator)

**Week 8 and week9: 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.
• Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
• Write a Query to display different travelling options available in British Airways.
• Display the number of days in a week on which the 9W01 bus is available.
• Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
• Find the distinct PNR numbers that are present.
• Find the number of tickets booked in each class where the number of seats is greater than 1. **Hint:** Use GROUP BY, WHERE and HAVING CLAUSES.
• Find the total number of cancelled seats.
• Write a Query to count the number of tickets for the buses, which travelled after the date '14/3/2009'. **Hint:** Use HAVING CLAUSES.

**Week10: 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
```


Week11: 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;

Week12: 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

<table>
<thead>
<tr>
<th>BUS</th>
<th>PASSENGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus No: Varchar: Pk</td>
<td>PNR_No : Numeric(9) : PK</td>
</tr>
<tr>
<td>Source : Varchar</td>
<td>Ticket_No: Numeric (9)</td>
</tr>
<tr>
<td>Destination : Varchar</td>
<td>Name: Varchar(15)</td>
</tr>
<tr>
<td></td>
<td>Age : int (4)</td>
</tr>
<tr>
<td></td>
<td>Sex:Char(10) : Male / Female</td>
</tr>
<tr>
<td></td>
<td>PPNO: Varchar(15)</td>
</tr>
</tbody>
</table>

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
Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. Introduction to SQL, Rick F. Vander Lans, 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
II MCA  
WEB TECHNOLOGIES  
II Semester  

Subject Code:  
L/T/P/C  
4/0/0/4  
Int :40, Ext:60, Total:100  

Course Objectives  
1. To provide the students a detailed knowledge on understanding the role of web services in the world of web applications, various different techniques used for building an web application.  
2. To provide the basic knowledge of HTML, Beans, XML for communication of information from source to destination.  
3. To provide the students with a solid foundation in various technologies containing technical concepts required to solve web application problems and also assists the students in pursuing higher studies.  

Course Outcomes  
1. An ability to develop project management skills related to web development.  
2. Master technical skills required for Web Developers through use of W3C standards, HTML, XHTML, and Style Sheets, develop databases and supporting code in Java language, Client and Server-Side Scripting languages such as JavaScript and JSP, performing unit and integration testing.  
3. Ability to communicate effectively to a wide variety of audiences, verbally, in writing, and electronically  

Pre-requisites  
1. Learner must have completed a course on J2SE.  
2. Should have well familiar with the concepts and terminology of Object Oriented Programming.  
3. Learner should have a good understanding of a SQL.  

UNIT-I  
HTML Common tags: List, Tables, images, forms, Frames; Cascading Style sheets;Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with JavaScrip, XML: Document type definition, XML Schemas, Document Object model,Presenting XML, Using XML Processors: DOM and SAX  

UNIT-II  

UNIT-III  

UNIT-IV  
JSP Application Development: Generating Dynamic Content, Using ScriptingElements Implicit JSP Objects, Conditional Processing – Displaying Values Usingan Expression to Set an Attribute, Declaring Variables and Methods Error Handlingand Debugging Sharing Data Between JSP pages, Requests, and Users PassingControl and Date between Pages – Sharing Session and Application Data – MemoryUsage Considerations, Deploying JAVA Beans in a JSP Page.  

UNIT- V
Introduction to struts framework, Model Layer, View Layer, ControllerLayer, Validate, Programming using struts framework.

**Teaching Methodologies**
1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**
2. The complete Reference Java Seventh Edition by Herbert Schildt. TMH.
3. Java Server Pages – Hans Bergsten, SPD O’Reilly

**REFERENCE BOOKS:**
1. Programming world wide web - Sebesta, Pearson
2. Core SERVLETS AND JAVA SERVER PAGES VOLUME 1: CORE TECHNOLOGIES By Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia.
5. Web Applications Technologies Concepts-Knuckles, John Wiley
6. Java Script, D. Flanagan, O’Reilly, SPD.
9. Beginning Web Programming - Jon Duckett, WROX.
II MCA II Semester

INFORMATION SECURITY

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
- The objectives of information security
- The importance and application of each of confidentiality, integrity, authentication and availability
- To develop a various cryptographic algorithms.
- Understand public-key cryptosystem and the enhancements made to IPv4 by IPSec.
- To how to generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
- To familiarize with Intrusions and intrusion detection, Web security and Firewalls.

Course Outcomes
- Should be able to work with various cryptographic algorithms.
- Should be able to Understand Intrusions and intrusion detection, Web security and Firewalls.
- Should be able to understand difference between the various Message Authentication algorithms.

Pre-requisites
- Learner must have knowledge about computer networks.
- Should be well familiar with the concepts and terminology of computer networks.

Teaching Aids

UNIT I

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

UNIT IV
E-Mail Security: Pretty Good Privacy, S/MIME


UNIT V

Intruders, Virus and Firewalls: Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls.

Teaching Methodologies
- PPT’s are using for students friendly understanding.
- Executing the programs in the lab based on the class room theoretical algorithms

TEXT BOOKS:

REFERENCE BOOKS:
3. Information Security, Principles and Practice: Mark Stamp, Wiley India.
5. Introduction to Network Security: Neal Krawetz, CENGAGE Learning
II MCA     DATA WAREHOUSING AND DATA MINING
II Semester

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives

• To understand data mining principles and techniques: Introduce DM as a cutting edge business intelligence method and acquaint the students with the DM techniques for building competitive advantage through proactive analysis, predictive modeling, and identifying new trends and behaviors.
• To learn how to gather and analyze large sets of data to gain useful business understanding.
• To learn how to produce a quantitative analysis report/memo with the necessary information to make decisions. And Learning how to gather and analyze large sets of data to gain useful business understanding.
• To describing and demonstrating basic data mining algorithms, methods, and tools
• To identifying business applications of data mining
• To develop and apply critical thinking, problem-solving, and decision-making skills.
• Overview of the developing areas - web mining, text mining, and ethical aspects of data mining.

Course Outcomes

• Understand the principles behind DWDM technologies and the characteristics of DWDM components
• Understand why the data warehouse in addition to traditional database systems.
• Ability to perform the preprocessing of data and apply mining techniques on it.
• Ability to identify the association rules, classification and clusters in large data sets.
• Ability to solve real world problems in business and scientific information using data mining.

Pre-requisites

• Fundamentals of SQL and Database.
• Experience with programming in some common programming language.
• A good understanding of data structures and algorithms.

UNIT- I
Introduction : Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Integration of a Data Mining System with a Database or a Data Warehouse System, Issues in Data Mining. Data Preprocessing: Need for Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

UNIT- II
Data Warehouse and OLAP Technology for Data Mining:Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Usage of Data Warehousing Online Analytical Processing and Mining.Data Cube Computation: Efficient Methods for simple Data Cube Computation (Full Cube, Iceberg Cube, Closed Cube and Shell Cube), Discovery Driven exploration of data cubes, Attribute-Oriented Induction for data characterization and its implementation

UNIT -III
Mining Frequent Patterns, Associations and Correlations: Basic Concepts, The Apriori algorithm for finding frequent itemsets using candidate generation, Generating association rules from frequent itemsets, Mining frequent itemsets without candidate generation, Mining various kinds of Association Rules, Correlation Analysis

UNIT- IV
Classification and Prediction: Description and comparison of classification and prediction, preparing data for Classification and Prediction. Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Back propagation Prediction, linear and non-linear regression, evaluating accuracy of a Classifier or a Predictor

UNIT - V
Cluster Analysis: Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, k-means and k-mediods methods, CLARANS, Agglomerative and divisive hierarchical clustering, chameleon dynamic modeling, clustering based on density distribution function, wavelet transformation based clustering, conceptual Clustering, Constraint-Based Cluster Analysis, Outlier Analysis.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. Data Mining –Concepts and Techniques - Jiawei Han, Micheline Kamber, and Jian Pei, Morgan Kaufmann Publishers, 3rd Edition, ELSEVIER.
2. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.

REFERENCE BOOKS:
4. The Data Warehouse Life cycle Tool kit – Ralph Kimball Wiley student edition
6. Data Mining Introductory and advanced topics – Margaret H Dunham, Pearson education
II MCA

SCRIPTING LANGUAGES

(ELECTIVE-I)

Subject Code: L/T/P/C

4/0/0/4

Int :40, Ext:60, Total:100

Course Objectives

- The course demonstrates an in-depth understanding of the tools and the scripting languages necessary for design and development of applications.
- To provide a foundation to use scripts for use on the internet.
- To provide the basic knowledge to use python with object orientation and database programming.

Course Outcomes

- Ability to develop scripts in PERL and use PHP for web pages.
- Develop skills to use python with objects and to access databases.

Pre-requisites

- Experience with programming in some common programming language.
- Knowledge of Basics of computer networks and databases.

UNIT-I

Introduction to PERL and Scripting: Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of 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, advance perl - finer points of looping, pack and unpack, filesystem, eval, datastructures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT-II


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 – Translating Websites- Updating Web sites Scripts, Creating the Localization Repository, Translating Files, text, Generate Binary Files, Set the desired language within your scripts, Localizing Dates, Numbers and Times.

UNIT-IV


UNIT-V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. The World of Scripting Languages, David Barron, Wiley India.

REFERENCE BOOKS:
1. Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and
2. PHP, J. Lee and B. Ware (Addison Wesley) Pearson Education.
3. Programming Python, M. Lutz, SPD.
5. PHP 5.1, I. Bayross and S. Shah, The X Team, SPD.
6. Core Python Programming, Chun, Pearson Education.
8. Perl by Example, E. Quigley, Pearson Education.
10. PHP and MySQL by Example, E. Quigley, Prentice Hall (Pearson).
12. PHP Programming solutions, V. Vaswani, TMH.
II MCA

E-COMMERCE
(ELECTIVE-I)

Subject Code: L/T/P/C 4/0/0/4
Int :40, Ext:60, Total:100

Course Objective:
To Know the Electronic Commerce basics and how the Consumer applications are emerging as Online. E-Commerce is the core for today’s Online Transactional Processing. Also to know the fields which are using E-Commerce applications.

Course Outcomes
After reading E-Commerce, a student is able to know the difference between Manual Transactions and Online Transactions and also the power of the Electronic Commercial applications. Student is also able to understand the implementation of the Intra-Organizational and Inter-Organizational commercial applications.

Pre-requisites
Student should have knowledge of manual business transactions and should know the difference between local market and global market.
He/she has to think what is a transaction and how it is implemented automatically.

UNIT-I

UNIT-II
Electronic payment systems Digital Token-Based, Smart Cards, Credit Cards, Risks in Electronic Payment systems. Inter Organizational Commerce: EDI, EDI Implementation, Value added networks.

UNIT-III

UNIT-IV

UNIT - V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOK :

REFERENCES :
1. E-Commerce fundamentals and applications Hendry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang, John Wiley.
II MCA II Semester
GRID AND CLOUD COMPUTING (ELECTIVE-I)

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
- To Familiarise Basics, techniques and tools for Grid & Cloud Computing
- To understand any kind of heterogeneous resources over a network using open standards
- To get insight of the Service models
- To introduce the broad perceptive of cloud architecture and model
- To understand the concept of Virtualization and design of cloud Services
- To understand the features of cloud simulator
- To learn to design the trusted cloud Computing system

Course Outcomes
- Student will explain the lead players in cloud.
- Student differentiates cluster, grid and cloud computing applications.

Pre-requisites
Students should have completed a course on Computer Organization and Computer Networks.

UNIT-I
System models for advanced computing –clusters of cooperative computing, grid computing and cloud computing; software systems for advanced computing-service oriented software and parallel and distributed programming models with introductory details, Features of grid and cloud platform.

UNIT-II
Cloud Computing services models and features in Saas, Paas and Iaas. Service oriented architecture and web services; Features of cloud computing architectures and simple case studies.

UNIT-III

UNIT-IV
Cloud programming Environmental- Map Reduce Hadoop Library from Apache, Open Source Cloud Software Systems –Eucalyptus.

UNIT-V
Grid Architecture and Service modeling, Grid resource management, Grid Application trends.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
II MCA II Semester

COMPUTER GRAPHICS (ELECTIVE-I)

Subject Code: L/T/P/C
4/0/0/4
Int:40, Ext:60, Total:100

Course Objectives
• To provide students a detailed understanding of primitives to transform 2D and 3D images
• To provide foundation to know ways of generation of curves and surfaces.

Course Outcomes
• Ability to implement methods of 2D and 3D transformations
• To implement methods for generating curves & surfaces with the constraints involved

Pre-requisites
• Knowledge of basic geometry and Mathematics
• Awareness of data structures, 3D images, curves and surfaces

UNIT-I
Geometry and line Generation: Points, Lines, Planes, Pixels and frame buffers, types of display devices, DDA and Brasenham’s Line Algorithms, Brasenham’s algorithms for circle generation, algorithm for ellipse generation, character generation, Aliasing and Antialiasing.

UNIT-II
Graphics Primitives: Display Files, Display processors, Algorithms for polygon generation, Polygon filling algorithms, NDC (Normalized device co-ordinates), pattern filling. 2D Transformations: Scaling, Rotation, translation, homogenous for ordinates, rotation about arbitrary point Reflections, Zooming.

UNIT-III
Segment tables, operations on segments, data structures for segments and display files. Windowing and clipping: Window, viewport, viewing transformation clipping, line and polygon clipping, generalized clipping, multiple windowing.

UNIT-IV

UNIT-V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
RICH INTERNET APPLICATIONS
(ELECTIVE-II)

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course 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.

Course Outcomes
Know multimedia applied on Internet.
Learn to use Adobe Flash, Flex 3 and Ajax with Dojo

Pre-requisites
- The course is for those interested in developing dynamic Web applications.
- One should have programming experience.

UNIT- I

UNIT-II

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
Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
1. Fundamentals of Multimedia by Ze-Nian Li and Mark S. Drew PHI Learning, 2004

REFERENCE BOOKS:
II MCA II Semester

DISTRIBUTED SYSTEMS (ELECTIVE-II)

Subject Code: L/T/P/C
4/0/0/4
Int.: 40, Ext.: 60, Total: 100

Course 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.

Course Outcomes
- The student distinguishes various architectures used to design distributed systems, such as client-server and peer-to-peer.
- The student will analyze and explain various interprocess communication techniques, such as Remote Procedure Calling, Design issues, Implementation, Asynchronous RPC.
- The student will build distributed systems using various techniques for tolerating partial failures, such as leasing, replication and interprocess coordination techniques, such as distributed mutual exclusion, distributed monitors, tuple spaces.

Pre-requisites
- To know concepts of operating systems
- To have knowledge of computer Networks

UNIT-I

UNIT-II
Distributed OS, Its kernel, Processes and Threads, Naming and Protection, Communication and Invocation, Virtual Memory, File Service components, Design issues, Interfaces, implementation techniques, SUN network file systems

UNIT-III
SNS – a name service model, its design issues, Synchronizing physical clocks, Logical time and logical clocks, Distributed coordination. Replication and its architectural model, Consistency and request ordering, Conversation between a client and a server, Transactions, Nested Transactions - Concurrency control, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control.

UNIT-IV
Distributed Transactions and Nested Transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed Deadlocks, Transactions with replicated data, Transaction recovery, Fault tolerance, Hierarchical and group masking of faults - Cryptography, Authentication and key distribution, Logics of Authentication, Digital signatures.

UNIT-V
Distributed shared memory, Design and Implementation issues, Sequential consistency and ivy, Release consistency and Munin, Overview of Distributed Operating systems Mach, Chorus.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**

**REFERENCE BOOKS:**
1. Distributed Operating Systems, Pradeep K. Sinha, PHI.
II MCA SOFTWARE PROJECT MANAGEMENT (ELECTIVE-II)

II Semester

Subject Code: L/T/P/C 4/0/0/4
Int:40, Ext:60, Total:100

Course Objectives

• Understand the specific roles in the organizations with their responsibilities
• Clear understanding of life cycle phases and its milestones
• Understanding role of project management including planning, scheduling, risk management etc
• A clear understanding of principles, techniques, methods and tools for model based management of software projects
• Understanding of basic infrastructure competencies

Course Outcomes
At the end of the course, the student shall be able to:

• Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
• Compare and differentiate organization structures and project structures.
• Implement a project to manage project schedule, expenses and resources with the application of suitable project management tools.

Pre-requisites

• Familiarity with software engineering principles
• Understanding of the importance of management in software development

UNIT-I


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


UNIT-IV


UNIT-V

**Teaching Methodologies**

1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**

**REFERENCE BOOKS:**
2. Head First PMP, Jennifer Greene & Andrew Stellman, O’Reilly, 2007
5. The art of Project management, Scott Berkun, O’Reilly, 2005.
II MCA II Semester

SOFT COMPUTING
(ELECTIVE-II)

Subject Code: L/T/P/C 4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
- To make the students understand the concepts of soft computing
- Gives a clear idea of Artificial Intelligence and Neural Networks.
- Student gets introduced to Fuzzy Sets and Fuzzy Logic and Genetic Algorithms.

Course Outcomes
- Learn to apply Predicate Logic and Rules.
- Techniques of training Neural Networks.
- Able to use classical relations and fuzzy relations.

Pre-requisites
- Must have an aptitude for logical thinking, problem solving.
- Students should have completed a course on Algorithm Analysis and Computer Programming.

UNIT-I
Artificial Intelligence 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

UNIT-III

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-IV

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
Course Objectives
1. Design and develop a simple interactive web application
2. Enable students to write simple server-side scripts for web interactivity
3. Students should be able to develop simple back-end database to support a website

Course Outcomes
1. An ability to design and implement web services.
2. Master the techniques needed for developing web applications using consistent databases.

Pre-requisites
1. Learner must have completed a course on J2SE.
2. Should have well familiar with the concepts and terminology of Object Oriented Programming.
3. Learner should have a good understanding of a SQL.

Hardware and Software requirement
1. A working computer system with either Windows or Linux
2. A web browser either IE or firefox
3. Tomcat web server and Apache web server
4. XML editor like Altova Xml-spy [www.Altovala.com/XMLSpy – free], Stylusstudio, etc.,
5. A database either Mysql or Oracle
6. JVM(Java virtual machine) must be installed on your system

Week-1:
Design the following static web pages required for an online book store web site. 1) HOME PAGE: The static home page must contain three frames.
Top frame : Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).
Left frame : At least four links for navigation, which will display the catalogue of respective links.
For e.g.: When you click the link “CSE” the catalogue for CSE Books should be displayed in the Right frame.
Right frame: The pages to the links in the left frame must be loaded here. Initially this page contains description of the web site.

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
<tr>
<td></td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td>Catalogue</td>
</tr>
<tr>
<td></td>
<td>Cart</td>
</tr>
</tbody>
</table>

CSE
ECE
EEE
CIVIL

Description of the Web Site
2) LOGIN PAGE:
This page looks like below:

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
<tr>
<td>CSE</td>
<td></td>
</tr>
<tr>
<td>ECE</td>
<td>Login :</td>
</tr>
<tr>
<td>EEE</td>
<td>Password:</td>
</tr>
<tr>
<td>CIVIL</td>
<td></td>
</tr>
</tbody>
</table>

3) CATALOGUE PAGE:
The catalogue page should contain the details of all the books available in
the web site in a table.
The details should contain the following:

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
</tbody>
</table>
| CSE  | Book : XML bible
Author : Winston
Publication : Wiely $40.5
Add to cart button |
| ECE  | Book : AI
Author : S.Russel
Publication : Princeton hall $63
Add to cart button |
| EEE  | Book : Java2
Author : Watson $35.5
Add to cart button |
| CIVIL| Book : HTML 4
Add to cart button |
Note: Week 2 contains the remaining pages and their description.

**Week-2:**

**4) CART PAGE:**
The cart page contains the details about the books which are added to the cart.
The cart page should look like this:

<table>
<thead>
<tr>
<th>Logo</th>
<th>Web Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Login</td>
</tr>
<tr>
<td></td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td>Catalogue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CSE</th>
<th>ECE</th>
<th>EEE</th>
<th>CIVIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book name</td>
<td>Price</td>
<td>Quantity</td>
<td>Amount</td>
</tr>
<tr>
<td>Java 2</td>
<td>$35.5</td>
<td>2</td>
<td>$70</td>
</tr>
<tr>
<td>XML bible</td>
<td>$40.5</td>
<td>1</td>
<td>$40.5</td>
</tr>
</tbody>
</table>

**Total amount**  $130.5

**5) REGISTRATION PAGE:**
Create a “registration form” with the following fields

1) Name (Text field)
2) Password (password field)
3) E-mail id (text field)
4) Phone number (text field)
5) Sex (radio button)
6) Date of birth (3 select boxes)
7) Languages known (check boxes – English, Telugu, Hindi, Tamil)
8) Address (text area)

**Week 3:**

**VALIDATION:**
Write JavaScript to validate the following fields of the above registration page.

1. Name (Name should contain alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern i. name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

**Note:** You can also validate the login page with these parameters.

**Week-4:**
Design a web page using CSS (Cascading Style Sheets) which includes the following:

1) Use different font, styles:
   In the style definition you define how each selector should work (font, color etc.). Then, in the body of your pages, you refer to these selectors to activate the styles.
   For example:
2) Set a background image for both the page and single elements on the page. You can define the background image for the page like this:

BODY {background-image:url(myimage.gif);}

3) Control the repetition of the image with the background-repeat property. As background repeat: repeatTiles the image until the entire page is filled, just like an ordinary background image in plain HTML.

4) Define styles for links as

A:link
A:visited
A:active
A:hover

Example:

<style type="text/css"> A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline; color: red; } </style>

5) Work with layers: For example:

AYER 1 ON TOP:

AYER 2 ON TOP:
Week-5:
Write an XML file which will display the Book information which includes the following:
1) Title of the book
2) Author Name
3) ISBN number
4) Publisher name
5) Edition
6) Price
Write a Document Type Definition (DTD) to validate the above XML file. Display the XML file as follows.
The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose.
Note: Give at least for 4 books. It should be valid syntactically.
Hint: You can use some xml editors like XML-spy

Week-6:
JAVA BEANS:
Create a simple visual bean with a area filled with a color. The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false. The color of the area should be changed dynamically for every mouse click. The color should also be changed if we change the color in the “property window”.

Week-7:
1) Install TOMCAT web server and APACHE. While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.
2) Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root. Access the pages by using the urls:
   http://localhost:4040/rama/books.html (for tomcat)
   http://localhost:8080/books.html (for Apache)

Week-8:
User Authentication :
Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servlet for doing the following.

1. Create a Cookie and add these four user id’s and passwords to this cookie.
2. Read the user id and passwords entered in the Login form (week1). The values (user id and passwords) available in the cookies. If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display “ You are not an authenticated user “. Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInitParameters() method.

**Week-9:**
Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form). Practice ‘JDBC’ connectivity. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

**Week-10:**
Write a JSP which does the following job : Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (similar to week8 instead of cookies).

**Week-11:**
Create tables in the database which contain the details of items (books in our case like Book name, Price, Quantity, Amount) of each category. Modify your catalogue page (week2) in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using JDBC.

**Week-12:**
HTTP is a stateless protocol. Session is required to maintain the state. The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time (i.e., from different systems in the LAN using the ip-address instead of localhost). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method session.invalidate()). Modify your catalogue and cart JSP pages to achieve the above mentioned functionality using sessions.

**Teaching Methodologies**
1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**
2. JDBC, Servlets, and JSP, Black Book, K. Santosh Kumar, dreamtech.
Course Objectives

• Learn to perform data mining tasks using a data mining toolkit (such as open source WEKA), Understand the data sets and data preprocessing. Demonstrate the working of algorithms for data mining tasks such as association rule mining, classification, clustering and regression, Exercise the data mining techniques with varied input values for different parameters.
• To obtain practical experience using data mining techniques on real world data sets.
• Emphasize hands-on experience working with all real data sets.
• Understand various cryptographic algorithms by implementing and executing them.
• Implementing public-key cryptosystem.
• Implementing the fundamental ideas of public-key and private-key cryptography.

Pre-requisites

• Fundamentals of SQL and Database.
• Experience with programming in some common programming language.
• A good understanding of data structures and algorithms.

Data Warehousing and Data Mining:
Task 1: Credit Risk Assessment
Description:
The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the bank's profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient. To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
3. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

The German Credit Data:
Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. credit dataset (original) Excel spreadsheet version of the German credit data. In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer !) A few notes on the German dataset

• DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
• owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
• foreign_worker. There are millions of these in Germany (many from Turkey). It is very hard to get
German citizenship if you were not born of German parents.

- There are 20 attributes used in judging a loan applicant. The goal is to classify the applicant into one of two categories, good or bad.

**Subtasks:**

1. List all the categorical (or nominal) attributes and the real-valued attributes separately.
2. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.
3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.
4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy?
5. Is testing on the training set as you did above a good idea? Why or why not?
6. One approach for solving the problem encountered in the previous question is using cross-validation. Describe what is cross-validation briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease? Why?
7. Check to see if the data shows a bias against "foreign workers" (attribute 20), or "personal-status" (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka's GUI Explorer. Did removing these attributes have any significant effect? Discuss.
8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)?
10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees? How does the complexity of a Decision Tree relate to the bias of the model?
11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning. Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain? Also, report your accuracy using the pruned model. Does your accuracy increase?

**Task Resources**

- Mentor lecture on Decision Trees
- Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
- Decision Trees (Source: Tan, MSU)
Task 2: Hospital Management System
Data Warehouse consists of Dimension Table and Fact Table. REMEMBER The following Dimension
The dimension object (Dimension): _Name_ _Attributes (Levels) , with one primary key _Hierarchies
One time dimension is must. About Levels and Hierarchies Dimension objects (dimension) consist of a set
of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation.
Hierarchies describe parent-child relationships among a set of levels.

For example, a typical calendar dimension could contain five levels. Two hierarchies can be defined on
these levels:

H1: YearL > QuarterL > MonthL > WeekL > DayL H2: YearL > WeekL > DayL

The hierarchies are described from parent to child, so that Year is the parent of Quarter, Quarter the
parent of Month, and so forth.

About Unique Key Constraints
When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for each level
of the hierarchy and a unique key constraint on the lowest level (Base Level) Design a Hospital
Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier,
Time. Where measures are ‘NO UNITS’, UNIT PRICE.

Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

PATIENT (patient_name, Age, Address, etc.,)

MEDICINE (Medicine_Brand_name, Drug_name, Supplier, no_units, Uinit_Price, etc.,)

SUPPLIER (Supplier_name, Medicine_Brand_name, Address, etc.,)

If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably.
Design the Hospital Management system data warehouse using all schemas. Give the example 4-D cube
with assumption names.

INFORMATION SECURITY LAB
List of programs:
1. Write a Java program to perform encryption and decryption using the following algorithms
   a. Caesar cipher
   b. Substitution cipher
   c. Hill Cipher
2. Write a C/Java program to implement the DES algorithm logic.
3. Write a C/Java program to implement the Blowfish algorithm logic.
4. Write a C/Java program to implement the AES algorithm logic.
5. Write the RC4 logic in Java.
6. Implement DES-2 and DES-3 using Java cryptography package.
7. Write a Java program to implement RSA algorithm.
8. Implement the Diffie-Hellman Key Exchange mechanism
9. Calculate the message digest of a text using the SHA-1 algorithm in Java.
10. Calculate the message digest of a text using the MD5 algorithm in Java.
11. Explore the Java classes related to digital certificates.
12. Write a program in Java, which performs a digital signature on a given text.
Course Objectives
- To learn the characteristics of mobile applications in J2ME and Android.
- Understand the intricacies of UI required by mobile applications.
- To study about the design aspects of mobile application.
- To understand the best practices to develop mobile application.
- To store data on mobile device.
- To retrieve data from server.
- To realize the differences between developing conventional applications and mobile applications.
- To learn programming skills in J2ME and Android.
- To learn development and programming of mobile applications.

Course Outcomes
Upon Completion of the course, the students should be able to:
- Design and implement the user interfaces of mobile applications in J2ME and Android
- Design the mobile applications that are aware of the resource constraints of the mobile devices.
- Develop advanced mobile applications that accesses network.
- Develop mobile application which uses local storage on mobile device.

Pre-requisite:
- Basic understanding of java programming.
- Need basic OOP concept.

UNIT-I
J2ME Overview

UNIT-II
J2ME Architecture and Development Environment

UNIT-III
Commands, Items, and Event Processing and RMS
J2ME User Interfaces, Display Class, The Palm OS Emulator, Command Class, Item Class, Exception Handling
High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, Text Box Class, Ticker Class Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation
Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener

UNIT- IV
Generic Connection Framework
The Connection, Hypertext Transfer Protocol, Communication Management Using HTTP Commands, Session Management, Transmit as a Background Process

UNIT-V
Introduction to Android
Overview Android, Features of Android, Android architecture, Dalvik Virtual Machine & .apk file extension
Fundamentals: Android application life cycle, Basic building blocks – Activities, Services, Broadcast Receivers & Content providers.
UI Components - Views & notifications. Components for communication - Intents & Intent Filters
Android API levels (versions & version names)

UNIT-VI
Android Application Structure
AndroidManifest.xml, Resources & R.java, Layouts
First sample application, Emulator-Android Virtual Device, Launching emulator, Editing emulator settings, Emulator shortcuts, Second App:- (switching between activities)

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
3. Beginning Android Application Development by Wei-Meng Lee Wiley India

REFERENCE BOOKS:
4. Beginning Android by Mark Murphy APress
5. Android Application Development: Programming with the Google SDK by Rick Rogers,
Subject Code: 4/0/0/4

Course Objectives

- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- To understand software test automation problems and solutions.
- To learn how to write software testing documents, and communicate with engineers in various forms.

Course Outcomes

By the end of the course, student should:

- Have an ability to apply software testing knowledge and engineering methods.
- Have an ability to design and conduct a software test process for a software testing project.
- Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
- Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.

Pre-requisites

Object Oriented Analysis and Design with UML, Software Engineering and Software metrics understanding

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, domain and interface testing, domains and testability.
Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

UNIT-IV
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).

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
1. The craft of software testing - Brian Marick, Pearson Education.
7. Software Testing, M.G.Limaye, TMH.
III MCA III Semester

OBJECT ORIENTED ANALYSIS AND DESIGN (using UML)

Subject Code: L/T/P/C 4/0/0/4

Int :40, Ext:60, Total:100

Course Objectives
- A clear understanding of all the notations used in the UML
- Understanding the key terms like class, object, state, behaviors, object etc
- A detailed understanding of different diagrams in UML
- To model any real worked application using UML diagrams
- Understanding of when to use generalization, aggregation and composition of different kinds of relationships

Course Outcomes
Upon completion of this course, the student should be able to
- Understand all the standard notation of UML
- Take up a case study and model it in different views with respect to user requirements
- Should be able to apply object-oriented concepts and modeling techniques to simulate real-life situations

Pre-requisites
- Knowledge of object oriented concepts
- Knowledge of structured programming
- Students must demonstrate familiarity with several different areas of knowledge and several different modes of inquiry

UNIT-I

UNIT-II

UNIT-III

UNIT-IV

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

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

REFERENCE BOOKS:
7. UML and C++, R.C. Lee, and W.M. Tepfenhart, PHI.
10. Learning UML 2.0, Russ Miles and Kim Hamilton, O’Reilly, SPD.
III MCA            WEB SERVICES
            (ELECTIVE-III)           I Semester

Subject Code:          L/T/P/C
            4/0/0/4

Int :40, Ext:60, Total:100

Course Objectives
• To Understand the details of the core distributed computing technologies  
• To learn the underlying web services architecture 
• To provide foundations of WSDL and interoperability means in web services  
• To explore interoperability between different frameworks.
• To learn basic concepts of SOA.

Course Outcomes
• To have knowledge of interoperability methods between different architecture in web services  
• To be aware of limitations of interoperability

Pre-requisites
• Be aware of java and web applications  
• Be familiar with details of client/server architecture

UNIT-I
Evolution and Emergence of Web Services: Evolution of distributed computing, Core distributed computing technologies – client/server, CORBA, JAVA RMI, Micro Soft DCOM, MOM, Challenges in Distributed Computing, role of J2EE and XML in distributed computing, emergence of Web Services and Service Oriented Architecture (SOA).

UNIT-II
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, basic steps of implementing web services, developing web services enabled applications.

UNIT-III
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, limitations of SOAP.

UNIT-IV
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, support for categorization in UDDI Registries, Publishing API, Publishing information to a UDDI Registry, searching information in a UDDI Registry, deleting information in a UDDI Registry, limitations of UDDI.

UNIT-IV
Web Services Interoperability: Means of ensuring Interoperability, Overview of .NET and J2EE.

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
3. XML, Web Services, and the Data Revolution, F.P. Coyle, Pearson Education.

REFERENCE BOOKS:
2. Java Web Services, D.A. Chappell & T. Jewell, O’Reilly, SPD.
6. Java Web Services Programming, R. Mogha, V. V. Preetham, Wiley India Pvt. Ltd.
III MCA  I Semester

DISTRIBUTED DATABASES
(ELECTIVE-III)

Subject Code: L/T/P/C
Int :40, Ext:60, Total:100

Course Objectives
The objective of the course is to learn the management of distributed data using distributed database management systems. The student should also acquire insight into difference between the centralized databases and distributed databases, distributed DBMS architecture, query decomposition and data localization, transaction management, distributed concurrency control, client/server architectures and distributed multi-DBMSs.

Course Outcomes
1. Key concepts and techniques for centralized databases as well as distributed databases and implementation, such as storage, indexing, query evaluation, query optimization, transaction management, concurrency control, cash recovery, deadlocks and catalog management.
2. Analyze and design distributed database systems based on the principles of distributed indexing, query evaluation, data replication, transaction management, concurrency and recovery.
3. Demonstrates the principles and techniques for database secure accessing and mining.

Pre-requisites
Fundamentals of Relational Algebra, SQL, DBMS

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

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

**Teaching Methodologies**
1. White Board and Marker
2. Power Point Presentations

**TEXT BOOKS:**
1. Distributed Databases Principles & Systems, Stefano Ceri, Giuseppe Pelagatti, TMH.

**REFERENCE BOOKS:**
III MCA  
EMBEDDED SYSTEMS  
(ELECTIVE-IV)  

Subject Code: L/T/P/C  
4/0/0/4  
Int :40, Ext:60, Total:100

Course Objectives  
• To provide detailed knowledge of the 8051 Architecture and program this microcontroller  
• To provide a foundation of a real-time operating system

Course Outcomes  
• Program an application on 8051 microcontroller  
• To have complete knowledge of a real-time operating system

Pre-requisites  
• Knowledge of the basics of Internal organization of a computer  
• Awareness of concepts involved with a microprocessor

UNIT-I  
Introduction to advanced architecture ARM and SHARC, Processor and memory organization and Instruction level parallelism;  
Networked embedded systems: Bus protocols, I2C bus and CAN bus; Internet-Enabled Systems, Design Example-Elevator Controller.

UNIT-II  

UNIT-III  
Applications: Interfacing with Keyboards, Displays, D/A and A/D Conversions, Multiple Interrupts, Serial Data Communication.

UNIT-IV  
Introduction to Real-Time Operating Systems: Tasks and Task States, Tasks and Data,Semaphores, and Shared Data; Message Queues, Mailboxes and Pipes, Timer Functions, Events,Memory Management,Interrupt Routines in an RTOS Environment.

UNIT-V  

Teaching Methodologies  
1. White Board and Marker  
2. Power Point Presentations

TEXT BOOKS :
1. Computers as Components-principles of Embedded computer system design, Wayne Wolf, Elsevier.
3. An Embedded Software Primer, David E. Simon, Pearson Education.

REFERENCE BOOKS:
1. Embedding system building blocks, Labrosse, via CMP publishers.
2. Embedded Systems, Raj Kamal, TMH.
3. Micro Controllers, Ajay V Deshmukhi, TMH.
5. Microcontrollers, Raj kamal, Pearson Education.
III MCA

ETHICAL HACKING
(ELECTIVE-III)

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
- To be aware of the Ethics and legality of a ethical hacker
- To provide students knowledge of different attacks used by hackers
- To give foundation of methods of system attacks, network based attacks on systems, servers and databases

Course Outcomes
- Ability to appreciate the techniques used in real-time attacks
- Develop skills and identify ways of prevent attacks on systems, servers and databases

Pre-requisites
- Well founded in programming techniques
- Aware of concepts of computer networks

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


UNIT-II


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 Subobjective 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


UNIT-IV


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


Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

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

REFERENCE BOOKS:
III MCA  

DESIGN PATTERNS  
(ELECTIVE-IV)

Subject Code:  

L/T/P/C  
4/0/0/4  
Int :40, Ext:60, Total:100

Course Objectives

- Analyze a software development problem and express its essence succinctly and precisely.
- Design a module structure to solve a problem, and evaluate alternatives.
- Implement a module so that it executes efficiently and correctly.
- Appreciate engineering issues in the development of software, such as the importance of addressing the user’s concerns, working with limited resources, maintainability, dependability, and division of labor.

Course Outcomes

- Identify key entities and relationships in the problem domain; write succinct textual descriptions of problems in the style of a user manual;
- Design programs consisting of a collection of modules exploiting well-known design patterns and devise appropriate module specifications and express them informally in terms of pre- and post-conditions.

Prerequisites

The student should be aware of imperative programming and details of deployment of applications.

UNIT-I

Introduction: What Is a Design Pattern?, Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

UNIT-II


Creational Patterns Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

UNIT-III

Structural Pattern Part-I: Adapter, Bridge, Composite.
Structural Pattern Part-II: Decorator, açade, Flyweight, Proxy.

UNIT-IV

Behavioral Patterns Part-I: Chain of Responsibility, Command, Interpreter, Iterator.
Behavioral Patterns Part-II: Mediator, Memento, Observer.

UNIT-V


TEXT BOOK:
1. Design Patterns By Erich Gamma, Pearson Education
REFERENCE BOOKS:
4. Head First Design Patterns By Eric Freeman-Oreilly-spd
5. Design Patterns Explained By Alan Shalloway,Pearson Education.
Gokaraju Rangaraju Institute Of Engineering And Technology(Autonomous)

III MCA I Semester

INFORMATION RETRIEVAL SYSTEMS (ELECTIVE-III)

Subject Code: L/T/P/C
4/0/0/4
Int :40, Ext:60, Total:100

Course Objectives
- To enable student understand the concepts of Information Retrieval Systems.
- To familiarize the techniques of retrievals from large collections of data.
- To analyze the retrieval systems for web search tasks.

Course Outcomes
- Should have acquired a good understanding on fundamentals of information retrieval techniques
- Should be able to apply the IR techniques on large text collections.

Pre-requisites
Logical thinking, Problem Solving techniques
Should have completed a course on Data Structures and Database Management Systems.

UNIT-I
Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital Libraries and Data Warehouses, Information Retrieval System Capabilities - Search, Browse, Miscellaneous.

UNIT-II

UNIT-III
Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters - User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, Weighted searches of Boolean systems, Searching the Internet and hypertext - Information Visualization: Introduction, Cognition and perception, Information visualization technologies.

UNIT-IV
Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems.
Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example – TREC results.

UNIT-V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:

REFERENCE BOOKS:
Gokaraju Rangaraju Institute Of Engineering And Technology(Autonomous)

III MCA I Semester

HUMAN COMPUTER INTERACTION (ELECTIVE-IV)

Subject Code: L/T/P/C

4/0/0/4

Int :40, Ext:60, Total:100

Course Objectives
1. The objectives and Importance of user Interface
2. Importance of Screen design and graphical user interface
3. Interaction between the Software tools and Interaction Devices.

Course Outcomes
1. Should be able to work with various graphical user interfaces.
2. Should be able to understand technological consideration in interface design.
3. Should be able to understand Multimedia, colors, uses problems.

Pre-requisites
1. Learner must have basic knowledge about user Interface.
2. Should be well familiar with the concepts and terminology of computer.

UNIT-I
Introduction: Importance of user Interface – definition, importance of good design.
Benefits of good design. A brief history of Screen design, The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

UNIT-II
Design process – Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, and understanding business junctions.

UNIT-III
Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

UNIT-IV

UNIT-V

Teaching Methodologies
1. White Board and Marker
2. Power Point Presentations
TEXT BOOKS:
1. The essential guide to user interface design, Wilbert O Galitz, Wiley DreamTech.
2. Designing the user interface. 3rd Edition Ben Shneidermann, Pearson Education Asia

REFERENCE BOOKS:
1. Human – Computer Interaction. Alan Dix, Janet Finlay, Gre Goryd, Abowd, Russell Bealg, Pearson Education
Course 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

Course Outcomes

- Will be able to build semantic web applications with features of social networking
- Will be able to distinguish Semantic web applications from other Web applications.

Pre-requisites

Web Search techniques

UNIT -I

UNIT -II
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.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

TEXT BOOKS:
REFERENCE BOOKS:
4. Programming the Semantic Web, T.Segaran, C.Evans, J.Taylor, O'Reilly, SPD.
III MCA  I Semester
MOBILE APPLICATION DEVELOPMENT LAB

Subject Code: L/T/P/C
0/0/6/2
Int :40, Ext:60, Total:100

Course Objectives
- To realize the differences between developing conventional applications and mobile applications.
- To learn programming skills in J2ME and Android
- To study about micro browser based applications to access the Internet using Sun Java Toolkit and Eclipse
- Create a small but realistic working mobile application for small computing devices.
- Categorize the challenges posed by developing mobile applications and be able to propose and evaluate and select appropriate solutions.

Course Outcomes
Upon Completion of the course, the students should be able to:
- Develop useful mobile applications for the current scenario in mobile computing and pervasive computing.
- Use Local Storage on device.
- Develop networked application.
- Use all the features of J2ME and Android
- Port(Deploy) application on mobile device.

Pre-requisites
- Knowledge of Java programming.
- Knowledge of inheritance, event handling and listeners.

Week-1: Installation of Java Wireless Toolkit (J2ME)
1) If the Java Development Kit (JDK) is not there or only having the Java Runtime Environment (JRE) installed, install the latest JDK from http://java.sun.com/javase/downloads/index.jsp. Current stable release of Java is JDK 6 Update 7 but check the web page in case there are newer non-beta releases available.
3) Run the installer (for example, for Windows it is: sun_java_wireless_toolkit- 2.5.2- windows.exe). The installer checks whether a compatible Java environment has been pre-installed. If not, it is necessary to uninstall old versions of Java and perform Step 1 again. Once after successful installation of Java and the tool kit compile this program and run the following program in the toolkit.

Steps to run this program in toolkit:
1. Start -> All Programs -> Sun Java Tool Kit -> Wireless Tool Kit
2. Click New Project – Enter Project Name -> Enter Class Name -> Click on Create Project.
3. Choose appropriate API Selection and Configurations.
4. Place Java Source file in WTK2.1 / WTK2.2\apps\projectname\src
5. Build the Project.
6. Run the Project.

import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
public class HelloWorld extends MIDlet{
private Form form;
private Display display;
Week-2: Working with J2ME Features

Working with J2ME Features: Say, creating a Hello World program. Experiment with the most basic features and mobile application interaction concepts (lists, text boxes, buttons, radio boxes, soft buttons, graphics, etc).

2.2 Event Handling.
Create a menu which has the following options:
* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

2.3. Input checking
Create an MIDP application which examine, that a phone number, which a user has entered is in the given format.
* Area code should be one of the following: 040, 041, 050, 0400, 044
* There should 6-8 numbers in telephone number (+ area code)
Week-3 Threads & High Level UI:
3.1 Create a slide show which has three slides, which includes only text. Program should change to the new slide after 5 seconds. After the third slide program returns to the first slide.

3.2 High-level UI
Create a MIDP application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many right answers were right and shows them to user.

3.3 Create a MIDP application, where the user can enter player name and points. The program saves the information to the record using RMS at MIDP device. Program should also print out the top 10 player list.
to the end user. You can use this class in your game if you made own class for saving and reading record sets.

Week-4: Working on Drawing and Images

4.1 Create a slide show which has three slides, which includes pictures at PNG format. Program should change to the new slide other 5 seconds.

4.2 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array.

4.3 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array. You can enter four data (integer) values to the input text field.
Week-5: Developing Networked Applications using the Wireless Toolkit
Creating a Simple Client-Server Application. Create, compile and run a basic UDP-based client-server application.

Creating the Datagram Server project:
1) Click on Wireless Toolkit 2.5.2 under the group: All Programs→Sun Java (TM) Wireless Toolkit 2.5.2.
2) Click on 'New Project...' button.
3) Enter project name as 'DatagramServer'. Enter MIDlet name as 'DatagramServer'.
   Note that the MIDlet name is the same as the name of the class in the source code, which extends the MIDlet class, otherwise the application won’t run.
4) Another window pops up where it is required to select a target platform. Select 'MIDP 1.0' from the drop down list.
5) After clicking OK, the project is created; and the Wireless Toolkit tells that the name of the folder where source code files are created. The path of the source code folder is displayed in the debug output window. Creating and Compiling the DatagramServer source files:The Wireless Toolkit does not come with an IDE by default so Use any IDE or a text editor like Notepad.
   1) Create a new text file called DatagramServer.java in the source folder of the project. The exact path of this folder is displayed in the Wireless Toolkit window.
   2) Paste contents DatagramServer.java into the source file. Running your Server application on the Phone simulator:
      1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.
      2) A graphical window depicting a phone handset will appear with the name of your application highlighted on its screen as shown below.
      3) To start the application, click on the right soft-key (marked with a dot) below the ‘Launch’ command.
      4) The phone simulator might ask if it is OK to run the network application. Select ‘Yes’ by clicking on the appropriate soft-key. The server is now up and running.
      5) Keep the server running during the creation, compilation and running of the Datagram Client application.

Creating the DatagramClient project:
1) Use the same instance of the Wireless Toolkit that is used for creating and compiling the Datagram Server project.
2) Click on 'New Project...' button.
3) A new window pops up. Enter project name as 'DatagramClient'. Enter MIDlet name as 'DatagramClient'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class.
4) Another window pops up where one has to select a target platform. Select 'MIDP 1.0' from the drop down list.
5) After clicking OK, the project is created and the Wireless Toolkit tells where to place the source code files. The path of the source code folder is displayed in the debug output window as explained before.

Creating and Compiling the DatagramClient source files:
1) Create a new text file called DatagramClient.java in the source folder of the project.
2) Paste contents DatagramClient.java into the source file.
3) Then click on the Build button in the Wireless Toolkit window. If the compilation is OK, it will say Build Complete in the window's debug output window, otherwise it will show the errors.
Note: In the source code, use the System.out.println() statement to output debug information to this window.

Running your Client application on the Phone simulator:
1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.
2) A graphical window depicting a phone handset will appear with the name of the application highlighted on its screen.
3) To start the application, click on the right soft-key (marked with a dot) below the ‘Launch’ command.
4) The phone simulator might ask if it is OK to run the network application. Select ‘Yes’ by clicking on the appropriate soft-key. The client is now up and running.
5) When the client executes on the phone simulator, one should see a text box with the caption ‘Message’. Enter any message and press the right soft-key (corresponding to Send). If the client-server application is working properly, the screen of the server phone will display the message sent by the client and the client screen will now display a message sent by the server in response. The response message from the server is the original client message in reverse.
6) Try various features of the phone simulator including the different look-and-feel options.

Week-6: Authentication with a Web Server
6.1 Write a sample program to show how to make a SOCKET Connection from j2me phone. This J2ME sample program shows how to make a SOCKET Connection from a J2ME Phone. Many times there is a need to connect backend HTTP server from the J2ME application. shows how to make a SOCKET connection from the phone to port 80.

6.2 Login to HTTP Server from a J2ME Program This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server. Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server.
Note: Use Apache Tomcat Server as Web Server and Mysql as Database Server.

Week-7 & 8: Web Application using J2ME
The following should be carried out with respect to the given set of application domains: (Assume that the Server is connected to the well-maintained database of the given domain. Mobile Client is to be connected to the Server and fetch the required data value/information)
- Students Marks Enquiry
- Town/City Movie Enquiry
- Railway/Road/Air (For example PNR) Enquiry/Status
- Sports (say, Cricket) Update
- Town/City Weather Update
- Public Exams (say Intermediate or SSC)/ Entrance (Say EAMCET) Results Enquiry
Divide Student into Batches and suggest them to design database according to their domains and render information according the requests.

Android Programs:
Write an Android application program that accepts a name from the user and displays the hello name to
the user in response as output using Eclipse.
Write an Android application program that demonstrates the following:
   (i) LinearLayout
   (ii) RelativeLayout
   (iii) TableLayout
   (iv) GridView layout
Write an Android application program that demonstrates intent in mobile application development.
Course Objectives
Software Testing Lab
To learn to use the following (or Similar) automated testing tools to automate testing:
- Win Runner/QTP for functional testing.
- LoadRunner for Load/Stress testing.
- Test Director for test management.
- JUnit, HTML Unit, CPP Unit.

UML Lab
- Understand how UML supports the entire OOAD process.
- Become familiar with all phases of OOAD.

Course Outcomes
Software Testing Lab
- By the completion of the lab, the student should be able to use automated testing tools like WinRunnerm Test Director, JUnit etc

UML Lab
- By the completion of the lab, the student should be able to take up any case and do analysis, understand problem space and model the solution using different diagrams with respect to different user requirements.
- Should be able to apply object-oriented concepts and modeling techniques to simulate real-life situations

Pre-requisites
Software Testing Lab
- Familiarity with framing test cases
- Knowledge on various software engineering principles

UML Lab
- Knowledge of object oriented concepts
- Knowledge of structured programming
- Students must demonstrate familiarity with several different areas of knowledge and several different modes of inquiry.

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.
5. State Diagram
6. Activity Diagram.
7. Component Diagram
8. Deployment Diagram.

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.

**Software Testing Lab**

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.

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