

ACADEMIC REGULATIONS AND COURSE STRUCTURE

For

MASTER OF COMPUTER APPLICATIONS
(Applicable for the batches admitted from 2013-14)



MALLA REDDY ENGINEERING COLLEGE
(Autonomous)
Affiliated to
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

MALLA REDDY ENGINEERING COLLEGE

(Autonomous)

Hyderabad - 500 100, A. P.

August/September 2013

Academic Regulations 2013 for M.C.A. (Regular)

(Effective for the students admitted into first year from the academic year 2013-2014)

The MCA Degree of the Jawaharlal Nehru Technological University, Hyderabad, offered by Malla Reddy Engineering College (Autonomous), Hyderabad, shall be conferred on the candidates who are admitted to the program and fulfill all the requirements for the award of the Degree:

1.0 ELIGIBILITY FOR ADMISSIONS

Admission to the above program shall be made subject to the eligibility, qualifications and specializations prescribed by the University from time to time.

Admissions shall be made on the basis of merit rank obtained by the qualifying candidate at ICET examination or at an Entrance Test conducted by the university subject to reservations prescribed by the university from time to time.

2.0 AWARD OF MCA DEGREE

- 2.1 A Student shall be declared eligible for the award of MCA degree, if he pursues a course of study and completes it successfully for not less than three academic years and not more than six academic years.
- 2.2 If a student, who fails to fulfill all the academic requirements for the award of the degree within SIX academic years from the year of their admission, shall forfeit his/her seat in MCA course.
- 2.3 After six academic years of course work, the candidate is permitted to write the examinations for two more years
- 2.4 The minimum instruction days for each semester/year shall be 90 days.
- 2.5 The student shall register for 119 credits and secure all the 119 credits

3.0 ATTENDANCE

- 3.1 A candidate shall be deemed to have eligibility to write end semester examinations if he has put in a minimum of 75% of attendance in aggregate of all the subjects.
- 3.2 Condonation of shortage of attendance up to 10%, i.e., 65% and above and below 75% may be given by the College Academic Committee consisting of Principal, Head of the Department and a senior faculty member.
- 3.3 Condonation of shortage attendance shall be granted only on genuine and valid reasons on representation by the candidate with supporting evidence.
- 3.4 Shortage of attendance below 65% shall in no case be condoned.
- 3.5 A candidate shall not be promoted to the next semester unless he fulfills the attendance requirements of the previous semester

- 3.6 Students whose shortage of attendance is not condoned in any semester are not eligible to write their end semester examinations of that class and their registration shall stand cancelled.
- 3.7 A prescribed fee shall be payable towards condonation of shortage of attendance.
- 3.8 A student shall not be promoted to the next semester unless he satisfies the attendance requirements of the previous semester.

4.0 EVALUATION AND ACADEMIC REQUIREMENTS:

The performance of the candidate in each semester shall be evaluated subject-wise, with a maximum of 100 marks for each practical, on the basis of the Internal Evaluation and End Examination

- 4.1 While 75 marks shall be awarded for theory subjects based on the performance in the External Examination, 25 marks shall be awarded based on the Internal Evaluation. Internal Evaluation shall be made based on the average of the marks secured in the two Mid-Term Examinations conducted, one in the middle of the Semester and the other immediately after the completion of instruction. Each mid term examination shall be conducted for a total duration of 120 minutes with Part A as one compulsory question for 10 marks and Part B with 3 questions to be answered out of 5 questions each question for 5 marks. If any candidate is absent for any subject of a mid -term examination, an additional exam will be conducted in the deserving cases based on the recommendations of the College Academic Committee. End semester examination is conducted for 75 marks with Part A as a compulsory question 25 marks and Part B for 50 marks with 5 questions to be answered out of 7 Questions
- 4.2 In practical subjects, 75 marks shall be awarded based on the performance in the End Semester Examinations and 25 marks shall be awarded based on the day-to-day performance as Internal Marks.
- 4.3 There shall be a seminar presentation in the VI Semester. For Seminar there shall be only internal evaluation for 50 marks. A student has to secure a minimum of 50% of marks to be declared successful. The assessment will be made by a board consisting of Head of the Department and two other internal staff members of the department.
- 4.4 A student shall be deemed to have secured the minimum academic requirement in a subject if he secures a minimum of 40% of marks in the End semester Examination and a minimum aggregate of 50% of the total marks in the End Semester Examination and Internal Evaluation taken together
- 4.5 A student shall be promoted from II year to III year, only if he earns 40 credits up to II year I semester from all the examinations, whether or not the candidate takes the examinations
- 4.6 In case the candidate does not secure the minimum academic requirement in any subject (as specified in 4.4) he has to write the End semester Examination again in that subject. A candidate shall be given one chance to re-register for each subject provided the internal marks secured by him are less than 50%, and so failed in the end examination. In such a case the candidate must re-register for the subject(s) and secure the minimum required attendance. The attendance in re-registered subject(s) shall be calculated separately to decide the eligibility for writing the end examination in those subject(s). In the event of taking another chance, the internal marks and end examination marks already obtained in the previous attempt will stand cancelled
- 4.7 A student shall be allowed to submit the project report only after fulfilling the attendance requirements of all the semesters. The Viva-Voce examination shall be

conducted at the end of the course work and after the completion of the End Semester examination of the final semester.

- 4.8 The Laboratory examination for MCA course must be conducted with two Examiners. One of them is the Laboratory Class Teacher, and the second examiner shall be other than the Class Teacher.
- 4.9 i) A students shall register and put up a minimum attendance for all 119 credits and earn all the 119 credits. Marks obtained for all 103 credits (excluding PG Project/Dissertation) shall be considered for the calculation of percentage of marks.
ii) Students who fail to earn 119 credits as indicated in the course structure within six academic years from the year of their admission shall forfeit their seat in MCA course and their admission shall stand cancelled.

Conditions to avail the benefit of Improvement of Internal Marks

- i. The candidate should have completed the course work in the College.
- ii. The candidate should pass all the subjects in which the internal marks secured are more than 50%, before availing the benefit of Improvement of Internal Marks.
- iii. The candidate will be given only one chance for each subject for improvement of internal marks for a maximum of three subjects, provided the internal marks secured are less than 50% in each subject and so the candidate has failed in the end examination.
- iv. In the event of availing the chance of improvement of internal marks, the internal marks and the end semester examination marks secured in the previous attempt stand cancelled.
- v. For each subject, the candidate has to pay a fee equivalent to one-third of the semester fee, and the amount is to be paid in the form of a DD in favour of Principal, Malla Reddy Engineering College(Autonomous), payable at Hyderabad.

5.0 EVALUATION OF PROJECT WORK:

- 5.1 A Project Review Committee (PRC) of the college is to be constituted with Principal/Director as chairperson, heads of the departments, which are offering PG courses, and two other senior faculty members of the department offering MCA.
- 5.2 Registration of Project work: A Candidate is permitted to register for the project work after satisfying the attendance requirement of all the courses (theory and practical courses) up to V semester.
- 5.3 Every candidate shall work on projects approved by the PRC of the College.
- 5.4 The duration the Project is for one semester.
- 5.5 Three copies of Thesis, certified by the supervisor shall be submitted to the College.
- 5.6 The project report shall be adjudicated by one examiner selected by the University out of 5-member panel submitted by the College who are eminent in that field of study.
- 5.7 The viva-voce examination of the project report shall be conducted by a board consisting of the External examiner (see section 5.6), the Head of the Department and Supervisor. The Board shall jointly report candidates work as:
- A. Excellent
 - B. Good
 - C. Satisfactory
 - D. Unsatisfactory

Head of the Department shall coordinate and make arrangements for the conduct of viva-voce examination.

- 5.8 If the report of the viva-voce is not satisfactory, the candidate will retake the viva-voce examination after three months. If he fails to get a satisfactory report at the

second viva-voce examinations, he shall have to re-register for a fresh Project after getting the approval of the PRC at the beginning of the following semester.

- 5.9 In case of the project work of any such student as mentioned in section 5.8, is summarily rejected again at the end of the following semester, he will not be eligible for the award of the degree.

6.0 AWARD OF DEGREE & CLASS:

After the student has satisfied the requirements prescribed for the completion of the program and is eligible for the award of MCA Degree, he shall be placed in one of the following four classes.

(The marks in internal evaluation and end examination shall be shown separately in the marks memorandum).

Class Awarded	% of marks to be secured
FIRST CLASS WITH DISTINCTION	70% and above
FIRST CLASS	Below 70% but not less than 60%
SECOND CLASS	Below 60% but not less than 50%
FAIL	Below 50%

7.0 WITHHOLDING OF RESULTS:

If the candidate has not paid any dues to the university or if any case of indiscipline is pending against him, the result of the candidate will be withheld and he will not be allowed into the next semester. The issue of degree is liable to be withheld in such cases.

8.0 TRANSITORY REGULATIONS:

Candidates who have discontinued or have been detained for want of attendance or who have failed after having undergone the course are eligible for admission to the same or equivalent subjects as and when subjects are offered, subject to a maximum of two subjects at a time.

9.0 GENERAL:

- 9.1 The academic regulations should be read as a whole for purpose of any interpretation.
- 9.2 In case of any doubt of ambiguity in the interpretation of the above rules, the decision of the Principal is final.
- 9.3 The College may change or amend the academic regulations and syllabus at any time and the changes and amendments made shall be applicable to all the students with effect from the date notified by the College.
- 9.4 Wherever the word he, him or his occur, it will also include she, her and hers.
- 9.5 There shall be no transfers to any college of Jawaharlal Nehru Technological University, Hyderabad.

MALPRACTICES RULES

	Nature of Malpractices / Improper conduct	Punishment
	<i>If the Candidate:</i>	
1(a)	Possesses or keeps accessible in the examination hall, any paper, note book, program-mable calculator, Cell phones, pagers, palm computers or any other form of material concerned with or related to the subject of the examination (theory or practical) in which he is appearing but has not made use of,(material shall include any marks on the body of the candidate which can be used as an aid in the subject of the examination).	Expulsion from the examination hall and cancellation of the performance in that subject only.
(b)	Gives assistance or guidance or receives it from any other candidate orally or by any other body language methods or communicates through cell phone with any candidate or person in or outside the exam hall in respect of any matter.	Expulsion from the examination hall and cancellation of the performance in that subject only of all the candidates involved. In case of an outsider, he will be handed over to the police and a case is registered against him.
2	Has copied in the examination hall from any paper, book, programmable calculators, palm computers or any other form of material concerned with or related to the subject of the examination (theory or practical) in which the candidate is appearing	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester/ year.
3	Impersonates any other candidate in connection with the examination.	The candidate who has impersonated shall be expelled from examination hall. The candidate is also debarred and forfeits the seat. The performance of the original candidate, who has been impersonated, shall be cancelled in all the subjects of the examination (including practical and project work) already appeared and shall not be allowed to appear for examination of the remaining subjects of that semester/year. The candidate is also debarred for consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat. If the imposter is an outsider, he will be handed over to the police and a case is registered against him.

4	Smuggles in the Answer book or additional sheet or takes out or arranges to send out the question paper during the examination or answer book or additional sheet, during or after the examination	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester /year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat.
5	Uses objection-able, abusive or offensive language in the answer paper or in letters to the examiners or writes to the examiner requesting him to award pass marks.	Cancellation of the performance in that subject.
6	Refuses to obey the orders of the Chief Super-intended/Assistant – Superintend/ any officer on duty or misbehaves or creates disturbances of any kind in and around the examination hall or organizes a walk out or instigates others to walk out, or threatens the officer-in charge or any person on duty in or outside the examination hall of any injury to this person or to any of his relations whether by words, either spoken or written or by signs or by visible representation, assaults the officer-in-charge, or any person on duty of his relations, or indulges in any other act of misconduct or mischief which results in damage to or destruction of property in the examination hall or any other part of college campus or engages in any other act which in the opinion of the officer on duty amounts to use of unfair means or misconduct or has the tendency to disrupt the orderly conduct of the examination.	In case of the students of the college, they shall be expelled from examination halls and cancellation of their performance in that subject and all other subjects the candidate(s) has (have) already appeared and shall not be permitted to appear for the remaining examination of the subjects of that semester/year. The candidates also are debarred and forfeit their seats. In case outsiders, they will be handed over to the police and a police case is registered against them.
7	Leaves the exam hall taking away answer script or intentionally tears of the script or any part thereof inside or out-side the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject only of all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic

		regulations in connection with forfeiture of seat
8	Possess any lethal weapon or firearm in the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has al-ready appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeiture of seat.
9	If student of the college, who is not the candidate for the particular examination or any person not connected with the college indulges in any malpractice or improper conduct mentioned in clauses 6 to 8.	Expulsion from the examination hall and cancellation of the performance in all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester /year. The candidate is also debarred and forfeiture of seat. Person(s) who does not belong to the college will be handed over to the police and a police case is registered against them.
10	Comes in a drunken condition to the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has al-ready appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year.
11	Coping detected on the basis of internal evidence, such as, during valuation or during special scrutiny.	Cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work of that semester /year examinations.
12	If any mal-practice is detected which is not covered in the above clauses 1 to 11 shall be reported to the University for further action to award suitable punishment.	

DISCIPLINARY ACTION FOR IMPROPER CONDUCT IN EXAMINATIONS

- Malpractices identified by squad or special invigilators:** Punishment to the candidates as per the above guidelines

MALLA REDDY ENGINEERING COLLEGE
(AUTONOMOUS)
Affiliated to JNTUH, HYDERABAD
I Year Scheme

MCA 1st Year (I-Semester)

Subject Code	Subject Title	L	P
MR13P0M8	Mathematical Foundations of Computer Science	4	-
MR13PA01	Computer Organization	4	-
MR13PA02	Computer programming	4	-
MR13P0M9	Probability and Statistics	4	
MR13PB48	Accounting and Financial Management	4	
MR13PA03	IT Workshop	-	3
MR13PA04	Computer programming Lab	-	6
MR13PA05	Computer Organization Lab	-	6

MCA 1st Year (II-Semester)

Subject Code	Subject Title	L	P
MR13PA06	Data Structures and Algorithms	4	-
MR13PA07	Operating Systems	4	-
MR13PB49	Organization Structure and Personnel Management	4	-
MR13POMA	Operations Research	4	
MR13P0E6	English Language Communication Skills	4	
MR13PA08	Data Structures Lab through C	-	6
MR13PA09	Operating Systems Lab	-	6
MR13P0E7	English Language Communication Skills Lab	-	3

w.e.f. 2013-14 Academic Year

**MALLA REDDY ENGINEERING COLLEGE
(AUTONOMOUS)**

Affiliated to JNTUH, HYDERABAD

II Year Scheme

MCA 2nd Year (I-Semester)

Subject Code	Subject Title	L	P
MR13PA10	Database Management Systems	4	-
MR13PA11	Computer Networks	4	-
MR13PA12	Linux Programming	4	-
MR13PA13	Software Engineering	4	-
MR13PA14	Java Programming	4	-
MR13PA15	Database Management Systems Lab	-	6
MR13PA16	Java and Linux Programming Lab	-	6

MCA 2nd Year (II-Semester)

Subject Code	Subject Title	L	P
MR13PA17	Data Warehousing and Data Mining	4	-
MR13PA18	Web Technologies	4	-
MR13PA19	Information Security	4	-
MR13PB50 MR13PA20 MR13PA21 MR13PA22	Elective –I Management Information Systems Scripting Languages Computer Graphics E-Commerce	4	-
MR13PA23 MR13PA24 MR13PA25 MR13PA26	Elective –II Distributed Systems Software Project Management Soft Computing Compiler Design	4	-
MR13PA27	Web Technologies Lab	-	6
MR13PA28	Data Warehousing & Data Mining and Information Security Lab	-	6

**MALLA REDDY ENGINEERING COLLEGE
(AUTONOMOUS)**

Affiliated to JNTUH, HYDERABAD

**III Year Scheme
MCA 3rd Year (I-Semester)**

Subject Code	Subject Title	L	P
MR13PA29	Mobile Application development	4	-
MR13PA30	Software Testing Methodologies	4	-
MR13PA31	Object-Oriented Analysis and Design (using UML)	4	-
MR13PA32 MR13PA33 MR13PA34 MR13PA35	Elective-III Ethical Hacking Multimedia and Rich Internet Applications Information Retrieval Systems Semantic Web and Social Networks	4	-
MR13PA36 MR13PA37 MR13PA38 MR13PA39	Elective-IV Web Services Adhoc and Sensor Networks Design Patterns Distributed Databases	4	-
MR13PA40	Mobile Application Development Lab	-	6
MR13PA41	Software Testing Methodologies and UML Lab	-	6

MCA 3rd Year (II-Semester)

Code	Subject	C
	Seminar	4
	Dissertation/Thesis Excellent/Good/Satisfactory/Not-Satisfactory	16

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE**Objectives:**

To develop problem-solving techniques and explore topics in a variety of areas of discrete mathematics, including but not limited to logic, graph theory, set theory, recursions, combinatorics, and algorithms.

Students will learn to express statements in the language of formal logic and draw conclusions, model situations in terms of graph and set theory, find and interpret recursive definitions for mathematical

sequences, use combinatorial methods to approach counting problems.

Outcomes:

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

UNIT I

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

UNIT II

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

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

UNIT III

Elementary Combinatorics: Basics of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial and Multinomial theorems, the principles of

Inclusion – Exclusion, Pigeon hole principles and its application.

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

COMPUTER ORGANIZATION

Objectives:

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

Outcomes:

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

PIPELINE AND VECTOR PROCESSING: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

MULTI PROCESSORS: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor Arbitration, InterProcessor Communication and Synchronization Cache Coherence, Shared Memory Multiprocessors.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

COMPUTER PROGRAMMING

Objectives:

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

Outcomes:

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

UNIT I

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

Introduction to the C Language – Background, C Programs, Identifiers, Types, Variables, Constants, Input / Output, Operators(Arithmetic, relational, logical, bitwise etc.), Expressions, Precedence and Associativity, Expression Evaluation, Type conversions, Statements- Selection Statements(making decisions) – if and switch statements, Repetition statements (loops)-while, for, do-while statements, Loop examples, other statements related to looping – break, continue, goto, Simple C Program examples.

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.

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

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

UNIT IV

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

arguments, 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.

UNIT V

Searching and Sorting – Sorting- selection sort, bubble sort, Searching-linear and binary search methods.

Lists- Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Push and Pop Operations, Queues- Enqueue and Dequeue operations.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

PROBABILITY AND STATISTICS

Objectives:

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Text book of Probability and Statistics Dr.Shahnaz Bathul, V.G.S.Publishers 2003.
2. Probability and Statistics in Engineering, 4th Edition, William W.Hines, Douglas C.Montgomery, David M.Goldsman, Connie M.Borrow, Wiley Student Edition.
3. Probability, Statistics and Queuing Theory, 2nd Edition, Trivedi, John Wiley and Sons
4. Introduction to Probability and Statistics, J.S.Milton, Jesse C.Arnold, 4th edition, TMH.
5. Probability, Statistics and Random Processes, Dr.K.Murugesan, P.Guruswamy, Anuradha Agencies, Deepti Publications.
6. Probability and Statistics for Engineers, G.S.S.Bhisma Rao, Sitech, 2nd edition,2005.

7. Probability and Statistics for Engineers and Scientists, R.E.Walpole, S.L.Myers,K.Ye, Pearson.
8. Probability, Statistics and Reliability for Engineers and Scientists, B.M.Ayyub,R.H.McCuen, 2nd edition, Chapman&Hall/CRC, Special Indian Edition.
9. Probability and Statistics for Engineers and Scientists, Sheldon M Ross, 4th ed., Elsevier, 2011.

ACCOUNTING AND FINANCIAL MANAGEMENT

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

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

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

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

Objectives

- Explain financial accounting terminology and the recording process.
- Discuss and apply the basic principles of accounting, the accounting model, and the accounting cycle.
- Demonstrate and show how items such as assets, liabilities, share capital, etc., are reported in the financial statements.
- Carry out journal entries and adjustments.
- Prepare and analyze a trial balance and financial statements.
- Analyze a cash flow statement.
- Analyze financial statements using ratio analysis.
- Understand the elements, uses, and limitations of each financial statement and the relationships among the statements
- Understand how financial information, primarily that provided by the financial statements, can be used to analyze business operations and make economic decisions

- Identify the basic principles used in safeguarding assets and insuring the accuracy of accounting records
- Appreciate the role accounting software applications play in gathering, recording, reporting and interpreting financial accounting information
- Interpret and create standard financial statements.
- Perform financial statement analysis for the purposes of evaluating and forecasting in financial management.
- Evaluate a firm's working capital position.
- Estimate the components of cost of capital by applying time value of money principles.
- Perform net present value analysis for capital budgeting purposes.
- Evaluate risk in the capital budgeting process.
- Evaluate the cost of debt, preferred stock, and common stock as sources of capital.
- The Most important learning objective is how to do accounts in Computers.

Prerequisites

Some basic accounting knowledge and mathematics level is recommended.

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

Ability to analyze market competency in the global village

To become a Master in accounting and financial management.

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

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

UNIT V

Capital Budgeting: Capital and its significance, Types of Capital, Estimation of Fixed and Working capital requirements, Methods and sources of raising capital. Capital Budgeting:

features of capital budgeting proposals, Methods of Capital Budgeting: Payback Method, Accounting Rate of Return (ARR) and Net Present Value Method (simple problems).

TEXT BOOKS:

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

REFERENCE BOOKS:

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

IT WORKSHOP

Objectives:

The IT Workshop for engineers is a training lab course spread over 54 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel and Power Point. **PC Hardware** introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. **The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible.**

Internet & World Wide Web module introduces the different ways of hooking the PC on to the internet from home and workplace and 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) office 2007/ equivalent (FOSS) tool word: Importance of LaTeX and MS office 2007/ equivalent (FOSS) tool Word as word Processors, Details of the three tasks and features that would be covered in each, using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter.

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

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
6. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. PC Hardware and A+Handbook – Kate J. Chase PHI (Microsoft)

COMPUTER PROGRAMMING LAB

Objectives:

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

Outcomes:

Ability to write programs for different kinds of problems in C.

Work confidently in compilers like C and others.

Ability to identify appropriate data type or data structure to given problem.

Recommended Systems/Software Requirements:

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

Week 1

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

Week 2

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

Week 3

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

Week 4

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

Week 5

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

Week 6

- a) Write a C program that uses functions to perform the following operations:
 - i) To insert a sub-string in to a given main string from a given position.
 - ii) To delete n Characters from a given position in a given string.

b) Write a C program to determine if the given string is a palindrome or not

Week 7

a) Write a C program that displays the position or index in the string S where the string T begins, or - 1 if S doesn't contain T.

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

Week 8

a) Write a C program to generate Pascal's triangle.

b) Write a C program to construct a pyramid of numbers.

Week 9

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

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

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

Print x, n, the sum

Perform error checking. For example, the formula does not make sense for negative exponents - if n is less than 0. Have your program print an error message if n<0, then go back and read in the next pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too.

Week 10

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

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

Week 11

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

- i) Reading a complex number
- ii) Writing a complex number
- iii) Addition of two complex numbers
- iv) Multiplication of two complex numbers

(Note: represent complex number using a structure.)

Week 12

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

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

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

Week 13

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

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

Week 14

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

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

Week 15

a) Write a C program that implements the Selection sort method to sort a given array of integers in ascending order.

b) Write a C program that implements the Bubble sort method to sort a given list of names in ascending order.

Week 16

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

- i) Create a singly linked list of integer elements.
- ii) Traverse the above list and display the elements.

Week 17

Write a C program that implements stack (its operations) using a singly linked list to display a given list of integers in reverse order. Ex. input: 10 23 4 6 output: 6 4 23 10

Week 18

Write a C program that implements Queue (its operations) using a singly linked list to display a given list of integers in the same order. Ex. input: 10 23 4 6 output: 10 23 4 6

Week 19

Write a C program to implement the linear regression algorithm.

Week 20

Write a C program to implement the polynomial regression algorithm.

Week 21

Write a C program to implement the Lagrange interpolation.

Week 22

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

Week 23

Write a C program to implement Trapezoidal method.

Week 24

Write a C program to implement Simpson method.

TEXT BOOKS:

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

COMPUTER ORGANIZATION LAB

Objectives:

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

List of Sample Problems:

Write assembly language programs for the following using MASAM.

1. Write assembly language programs to evaluate the expressions:

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

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

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

b. Considering 2 digit, 4digit and 8 digit BCD numbers.

Take the input in consecutive memory locations and results also.

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

2. Write an ALP of 8086 to add two exponential numbers which are in IEEE 754 notation.

Display the results by using “int xx” of 8086. Validate program for the boundary Conditions.

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

a) Arrange in ascending and descending order.

b) Find max and minimum

c) Find average

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

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

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

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

c) Find whether given string substring or not.

d) Reverse a string e) Concatenate by taking another sting

Display the results by using “int xx” of 8086.

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

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

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

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

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

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

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

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

ex. Input: Advanced Programming in UNIX

Out put:

Vowel count

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

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

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

b) How many chips are needed to provide a memory capacity of 16K bytes?

11. A computer employs RAM chips of 256X8 and ROM chips of 1024 X 8. The computer needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, 10 for interface registers.

a. How many RAM and ROM chips are needed?

b. Draw a memory-address map for the system.

c. Give the address range in hexadecimal for RAM, ROM and interface.

12. Obtain the complement function for the match logic of one word in an associative memory. Draw the logic diagram for it and compare with the actual match logic diagram.

13. A two-way set associative cache memory uses blocks of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128K X 32.

a. Formulate all pertinent information required to construct the cache memory.

b. What is the size of the cache memory?

14. A digital computer has a memory unit of 64K X 16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.

a. How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.

b. How many bits are there in the tag, index, block, and word fields of the address format?

c. How many blocks can the cache accommodate?

15. An address space is specified by 24 bits and the corresponding memory space by 16 bits.

a. How many words are there in the address space?

b. How many words are there in the memory space?

c. If a page consists of 2K words, how many pages and blocks are there in the system.

16. A virtual memory has a page size of 1K words. There are eight pages and four blocks. The associative memory page table contains the following entries. Make a list of all virtual addresses (in decimal) that will cause a page fault.

Page Block

0 3

1 1

4 2

6 0

TEXT BOOKS:

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

2. Introduction to Assembly Language Programming, Sivarama P.Dandamudi, Springer Int. Edition, 2003.

3. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Application, 4th edition, W.A.Triebel, A.Singh, N.K.Srinath, Pearson Education.

DATA STRUCTURES AND ALGORITHMS

Objectives:

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

Outcomes:

Learn how to use data structure concepts for realistic problems.

Ability to identify appropriate data structure for solving computing problems in respective language.

Ability to solve problems independently and think critically

- Be able to analyze algorithms and improve the efficiency of algorithms.
- Apply different designing methods for development of algorithms to realistic problems, such as divide and conquer, greedy and etc.

Ability to understand and estimate the performance of algorithm

UNIT I

Basic concepts-Data types, Abstract Data Types, Data structures, Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Analysis-Big O, Omega and Theta notations.

Linear data structures- Linear Lists, Sequential and Linked allocation ,The list ADT, array and linked Implementations, Singly Linked Lists-Operations-Insertion, Deletion, Doubly Linked Lists- Operations- Insertion, Deletion, Stack ADT, definitions, operations, array and linked implementations, applications-infix to postfix conversion, recursion implementation, Queue ADT, definitions and operations ,array and linked Implementations

UNIT II

Non Linear data structures- Trees – Basic Terminology, Binary tree ADT, array and linked representations, traversals, threaded binary trees, Disjoint Sets, Union and Find algorithms, Priority Queues-Definition, ADT, Realizing a Priority Queue using Heap.

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

UNIT III

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

Sorting- Bubble Sort, Insertion Sort, Selection Sort, Heap Sort, Radix Sort. Algorithm Design method- Divide and Conquer method-applications- Quick sort, Merge sort, Comparison of Sorting methods, Model for External Sorting.

UNIT IV

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

UNIT V

Algorithm Design methods-Greedy method-applications-Kruskal's Algorithm for Minimum cost Spanning trees, Job Sequencing with deadlines, Single Source Shortest path problem, Dynamic Programming method-applications-Ordering matrix multiplications, Optimal Binary Search Trees, APSP problem.

Text Processing - Pattern matching algorithm- The Knuth-Morris-Pratt algorithm, Tries-Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

OPERATING SYSTEMS

Objectives:

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

Outcomes:

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

UNIT I

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

UNIT II

Process and CPU Scheduling - Process concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies: Linux, Windows.

Process Coordination – Process Synchronization, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT III

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

UNIT IV

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System

Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance, Case Studies: Linux, Windows.
Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management

UNIT V

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

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

ORGANIZATION STRUCTURE AND PERSONNEL MANAGEMENT

Objectives:

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

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

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

Outcomes:

The course will help each student to better:

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

UNIT I

Classical Theories of organization : Functional approach, classical theories of organization, division of labour, levels of authority, span of control, authority & responsibility, efficiency of management.

Behavioral theories of organization, limitations of formal organization, human relation, group behavior, committee and group making, motivation and morale.

UNIT II

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

UNIT III

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

UNIT IV

Strategic management: Objectives, importance policies, concept of core competence capability of organizational learning, strategic levels and planning, business level strategy and functional level,

PHASES OF PLANNING,SWOT, develop strategies and prepare strategic plan.

UNIT V

Communication: Importance of communication, inter personnel communication barriers of communication, communication in organizations, using communication skills to manage conflicts.

Impact of informational technology and fostering effective communication

TEXT BOOKS:

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

REFERENCE BOOKS:

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

OPERATIONS RESEARCH

Objectives:

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

Prerequisites

- Probability and Statistics

Outcomes:

Ability to understand the optimization methods and also learn where to use in computer programming.

Ability to apply the OR methods in different domains like operating system, algorithms and etc.

UNIT I

Introduction to Operations Research: Basics definition, scope, objectives, phases, models and limitations of Operations Research. Linear Programming Problem – Formulation of LPP, Graphical solution of LPP.

Simplex Method, Artificial variables, big-M method, two-phase method, degeneracy and unbound solutions.

UNIT II

Transportation Problem: Formulation, solution, unbalanced Transportation problem. Finding basic feasible solutions – Northwest corner rule, least cost method and Vogel's approximation method.

Optimality test: the stepping stone method and MODI method.

Assignment model: Formulation. Hungarian method for optimal solution. Solving unbalanced problem. Traveling salesman problem as assignment problem.

UNIT III

Sequencing models: Solution of Sequencing Problem – Processing n Jobs through 2 Machines – Processing n Jobs through 3 Machines – Processing 2 Jobs through m machines – Processing n Jobs through m Machines.

Replacement Models: Replacement of Items that Deteriorate whose maintenance costs increase with time without change in the money value. Replacement of items that fail suddenly: individual replacement policy, group replacement policy.

UNIT IV

Dynamic programming: Characteristics of dynamic programming. Dynamic programming approach for Priority Management employment smoothening, Stage Coach/Shortest Path and Reliability problems.

Games Theory: Competitive games, rectangular game, saddle point, minimax (maximin) method of optimal strategies, value of the game. Solution of games with saddle points, dominance principle.

Rectangular games without saddle point – mixed strategy for 2 X 2 games.

UNIT V

Inventory models: Inventory costs. Models with deterministic demand – model (a) demand rate uniform and production rate infinite, model (b) demand rate non-uniform and production rate infinite, model (c) demand rate uniform and production rate finite.

Queuing Theory: Essential Features of a queuing system. Performance measures of a queuing system.

Model 1: $\{(M/M/1) : (\infty/FCFS)\}$ Single server, Unlimited Queue model. Model 2: $\{(M/M/1) : (\infty/SIRO)\}$

Single server, Unlimited Queue model. Model III: $\{(M/M/1) : (N/FCFS)\}$ Single server, Finite Queue model.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

2013-14

MR13P0E6

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - I Year - II SEM

L T/P C
4 -/- 3

ENGLISH LANGUAGE COMMUNICATION SKILLS

Objectives:

- To teach students the four language skills - Listening, Speaking, Reading and Writing.
- To enable students comprehend the concept of communication.
- To train students convert the conceptual understanding of communication into every day practice.
- To help students cultivate the habit of Reading and develop their critical reading skills.
- To prepare students communicate their ideas relevantly and coherently in professional writing.
- To prepare students for the recruitment process.

UNIT I

INTRODUCTION

Basics of Communication - Principles of Communication - Types of Communication – Stages of Communication – Verbal and Non-verbal Communication – Channels of Communication – Barriers to Effective Communication – Formal and Informal Expressions in Various Situations.

UNIT II

READING & STUDY SKILLS

Reading Comprehension – Reading Strategies - Skimming and Scanning- Intensive and Extensive Reading– Unknown Passage for Comprehension - Critical Reading of Short Stories – Study Skills – Note Making – Summarizing.

UNIT III

LANGUAGE SUPPORT FOR WRITING

Formation of a Sentence – Types of sentences- Common errors in Writing - Writing coherent sentences using connectives and conjunctions– Tenses – Concord – Question Tags. Active and passive voice,- Practice Exercises –synonyms, antonyms, One Word Substitutes – Words Often Confused and Misspelt, Idioms and Phrases

UNIT IV

PROFESSIONAL WRITING

Difference between Spoken and Written Communication- Features of Effective Writing - Letter writing – Types, Parts and Styles of Formal Letters – Language to be used in Formal Letters – Letters of Enquiry, Complaint, and Apology with Replies – Letter of Application - Resume – E-mail , social networking.

UNIT V

REPORT WRITING

Types of Reports – Formats of Reports – Memo Format – Letter Format and Manuscript Format- Parts of Technical Report – Informational, Analytical and Project Reports – Idioms and Phrases.

REFERENCE BOOKS:

1. Meenakshi Raman & Sangeetha Sharma. 2012. *Technical Communication*. New Delhi: OUP
2. Suresh Kumar, E. *et al.* 2011. *Essential English*. Hyderabad: Orient Blackswan
3. Yadava Raju, B. and Muralikrishna, C. 2009. *Advantage English*. Hyderabad: Orient Blackswan
4. Rizvi, M. A. 2005. *Effective Technical Communication*. New Delhi: Tata McGraw Hill
5. Er. A. K. Jain, Dr. Pravin S. R. Bhatia & Dr. A. M. Sheikh. 2013. *Professional Communication Skills*. S. Chand Publishers. New Delhi.
6. Farhathullah, T.M. 2009. *English for Business Communication*. Bangalore: Prism Publishers
7. Bikram K Das. 2011. *Functional Grammar and Spoken and Written Communication in English*. Kolkata: Orient Blackswan
8. Board of Editors. 2010. *Delight and Wisdom- An Anthology of Short Stories*. Mumbai : Orient Blackswan
9. Kiranmai Dutt, P *et al.* 2011. *A Course in Communication Skills*. New Delhi: CUP India
10. Krishnaswamy, N. 2000. *Modern English – A Book of Grammar, Vocabulary and Usage*. Macmillan India Pvt. Ltd
11. Mukherjee, Meenakshi. 2009. *Let's Go Home and Other Stories*. Mumbai : Orient Blackswan
12. Ramachandran, K K. *et al.* 2007. *Business Communication*. New Delhi: Macmillan
13. Taylor, Ken. 2011. *50 ways to improve your Business English*. Hyderabad: Orient Blackswan

DATA STRUCTURES THROUGH C LAB

Objectives:

- To write and execute programs in C to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.
- To write and execute write programs in C to implement various sorting and searching methods.

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.

Recommended Systems/Software Requirements:

- Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space
- C compiler.

Week1:

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

- a) Create a singly linked list of integers.
- b) Delete a given integer from the above linked list.
- c) Display the contents of the above list after deletion.

Week2:

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

- a) Create a doubly linked list of integers.
- b) Delete a given integer from the above doubly linked list.
- c) Display the contents of the above list after deletion.

Week3:

Write a C program that uses stack operations to convert a given infix expression into its postfix Equivalent, Implement the stack using an array.

Week 4:

Write C programs to implement a double ended queue ADT using i) array and ii) doubly linked list respectively.

Week 5:

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

- a) Create a binary search tree of characters.
- b) Traverse the above Binary search tree recursively in Postorder.

Week 6 :

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

- a) Create a binary search tree of integers.
- b) Traverse the above Binary search tree non recursively in inorder.

Week 7:

Write C programs for implementing the following sorting methods to arrange a list of integers in Ascending order:

- a) Insertion sort b) Merge sort

Week 8:

Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order:

- a) Quick sort b) Selection sort

Week 9:

i) Write a C program to perform the following operation:

a) Insertion into a B-tree.

ii) Write a C program for implementing Heap sort algorithm for sorting a given list of integers in ascending order.

Week 10:

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

Week 11:

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

Week 12:

Write C programs for implementing the following graph traversal algorithms:

- a) Depth first traversal b) Breadth first traversal

TEXT BOOKS:

1. C and Data Structures, Third Edition, P.Padmanabham, BS Publications.
2. C and Data Structures, Prof. P.S.Deshpande and Prof. O.G. Kakde, dreamtech Press.
3. Data structures using C, A.K.Sharma, 2nd edition, Pearson.
4. Data Structures using C, R.Thareja, Oxford University Press.
5. C and Data Structures, N.B.Venkateswarlu and E.V.Prasad, S.Chand.
6. C Programming and Data Structures, P.Radha Krishna, Hi-Tech Publishers.

OPERATING SYSTEMS LAB

Objectives:

- At the end of the course, the students should be able to:
- To use Linux operating system for study of operating system concepts.
- To write the code to implement and modify various concepts in operating systems using Linux.

Outcomes:

- The course objectives ensure the development of students applied skills in operating systems related areas.
- Students will gain knowledge in writing software routines, modules for implementing various concepts of operating systems.

List of Programs:

1. Simulate the following CPU scheduling algorithms
 - a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate all file allocation strategies
 - a) Sequential b) Indexed c) Linked
3. Simulate MVT and MFT
4. Simulate all File Organization Techniques
 - a) Single level directory b) Two level c) Hierarchical d) DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention
7. Simulate all page replacement algorithms
 - a) FIFO b) LRU c) LFU etc.
8. Simulate Paging Technique of memory management.

2013-14

MR13POE7

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)

MCA - I Year - II SEM

L T/P C

- -/3 2

ENGLISH LANGUAGE COMMUNICATION SKILLS LAB

Objective:

The aim of the course is to enrich Interpretation Skills, Problem Solving Skills, Interpersonal Skills, Analytical Skills, and Leadership Skills of the students. Keeping in view the most essential requirement of communication Skills for the management students, this course lays emphasis on the language integrated Skills in simple and comprehensive manner.

Unit I:

Presentations- Characteristics of a good presentation-Types of presentations based on content and methods of –delivery, Strategies and stages of a presentation.

Unit II:

Group Discussions- Types of Group Discussions, Participation and Assessment and Evaluation of a Group discussion. Debate- Types of Debates, participation and Assessment and evaluation of a Debate.

Unit III:

Interview Skills- Pre interview Preparation, mock interviews, assessment and evaluation, performance appraisal.

Unit IV:

Business Etiquette and Human Values: Human values and Professional ethics, Appearance, Attire, Attitude, Dining etiquette, Telephone, email etiquette , Visiting card etiquette ,meeting etiquette, etc....

REFERENCES

1. LAB: Globarena E-Mentoring System, by Globarena Technologies Pvt. Ltd, Hyderabad. It can be provided in LAN and also can be accessed by registering in <http://gems.globarena.net>.
OR ANY OTHER SOFTWARE.
2. Sherfeild:Developing Soft skills-Pearson Education-2013.
3. Murphy: Effective Business communication, 7/e, TMH, 2013.
4. Colin BEARD: Experiential Learning, Kogan Page, 2013.
5. Anandamurugan, Placement Interview Skills for Success, TMH, 1/e, 2012.

DATABASE MANAGEMENT SYSTEMS

Objectives:

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

Outcomes:

Demonstrate the basic elements of a relational database management system.

Ability to identify the data models for relevant problems.

Ability to design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respective data.

Apply normalization and professional attitudes for the development of application software's.

UNIT I

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

UNIT II

Relational Algebra and Calculus: Relational Algebra – Selection and Projection, Set operations, Renaming, Joins, Division, Examples of Algebra Queries, Relational calculus – Tuple relational Calculus– Domain relational calculus – Expressive Power of Algebra and calculus.

Form of Basic SQL Query – Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set – Comparison Operators, Aggregate Operators, NULL values – Comparison using Null values – Logical connectives – AND, OR and NOT – Impact on SQL Constructs, Outer Joins, Disallowing NULL values, Complex Integrity Constraints in SQL Triggers and Active Data bases.

UNIT III

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

UNIT IV

Overview of Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control, Deadlocks – Performance of Locking – Transaction Support in SQL. Concurrency Control: Serializability, and recoverability – Introduction to Lock Management – Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques – Concurrency Control without Locking.

Crash recovery: Introduction to Crash recovery, Introduction to ARIES, the Log, Other Recovery related Structures, the Write-Ahead Log Protocol, Check pointing, recovering from a System Crash, Media recovery

UNIT V

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

Storing data: Disks and Files: -The Memory Hierarchy – Redundant Arrays of Independent Disks. Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM)

B+ Trees: A Dynamic Index Structure, Search, Insert, and Delete. Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendable Vs. Linear Hashing.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

COMPUTER NETWORKS

Objectives:

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

Outcomes:

- After this course students could understand and explore the basics of Computer Networks and Various Protocols. She/he will be in a position to understand the World Wide Web concepts.
- After this course he/she will be in a position to administrate a network and flow of information further he/she can understand easily the concepts of network security, Mobile and ad hoc networks.

UNIT I

Overview of the Internet: Protocol, Layering Scenario, TCP/IP Protocol Suite: The OSI Model, Internet history standards and administration; Comparison of the OSI and TCP/IP reference model

Data Link Layer – design issues, CRC Codes, Elementary Data link Layer protocols, sliding window protocol

UNIT II

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

UNIT III

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

UNIT IV

Internetworking: Tunneling, Internetwork Routing, Packet fragmentation, IPv4, Ipv6 Protocol, IP addresses, CIDR, ICMP, ARP, RARP, DHCP.

Transport Layer: Services provided to the upper layers elements of transport protocol-addressing connection establishment, connection release, Connection Release, Crash Recovery.

UNIT V

The Internet Transport Protocols UDP-RPC, Real Time Transport Protocols, The Internet Transport Protocols- Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Connection Management Modeling, The TCP Sliding Window, The TCP Congestion Control, The future of TCP.Application Layer-Introduction, providing services, Applications layer paradigms,

Client server model, Standard client-server application-HTTP, FTP, electronic mail, TELNET, DNS, SSH

TEXT BOOKS:

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

REFERENCE BOOKS:

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

LINUX PROGRAMMING

Objectives:

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

Prerequisites

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

Outcomes:

Work confidently in Linux environment.

Work with shell script to automate different tasks as Linux administration

UNIT I

Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities. Sed-Scripts, Operation, Addresses, Commands, Applications, awk- Execution, Fields and Records, Scripts, Operation, Patterns, Actions, Associative Arrays, String and Mathematical functions, System commands in awk, Applications.. Shell programming with Bourne again shell(bash)- Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts. Review of C programming concepts-arrays, strings (library functions), pointers, function pointers, structures, unions, libraries in C.

UNIT II

Files and Directories- File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, create, read, write, close, lseek, dup2,file status information-stat family, file and record locking-lockf and fcntl functions, file permissions - chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links – symlink, link, unlink. Directories-Creating, removing and changing Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir, seekdir, telldir functions.

UNIT III

Process – Process concept, Layout of a C program image in main memory, Process environment, environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process hierarchy, process states, process control - process creation, replacing a process image, waiting for a process, process termination, zombie

process, orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, system, I/O redirection, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes.

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

UNIT IV

Interprocess Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions.

Message Queues- Kernel support for messages, APIs for message queues, client/server example. Semaphores-Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Linux System Programming, Robert Love, O'Reilly, SPD, rp-2007.
2. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education, 2003.
3. Advanced Programming in the Unix environment, 2nd Edition, W.R.Stevens, Pearson Education.
4. System Programming with C and Unix, A.Hoover, Pearson.
5. Unix System Programming, Communication, Concurrency and Threads, K.A.Robbins and S.Robbins, Pearson Education.
6. Unix shell Programming, S.G.Kochan and P.Wood, 3rd edition, Pearson Education.

SOFTWARE ENGINEERING

Objectives:

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

Outcomes:

Ability to identify the minimum requirements for the development of application.

Ability to develop, maintain, efficient, reliable and cost effective software solutions

Ability to critically thinking and evaluate assumptions and arguments.

UNIT I

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

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

UNIT II

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

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

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

UNIT III

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

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

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

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

UNIT IV

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

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

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

UNIT V

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

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

JAVA PROGRAMMING

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.

Outcomes:

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

UNIT I

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

UNIT II

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

Packages-Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

UNIT III

Files – streams- byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class, java.io.

Networking – Introduction, Manipulating URLs, Ex. Client/Server Interaction with Stream Socket Connections, Connectionless Client/Server Interaction with Datagrams, java.net.

UNIT IV

Exception handling – Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of

try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes. Guide lines for proper use of exceptions.

Multithreading - Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, interthread communication, thread groups, daemon threads.

UNIT V

GUI Programming with Java - The AWT class hierarchy, Introduction to Swing, Swing vs. AWT, MVC architecture, Hierarchy for Swing components, Containers – Top-level containers – JFrame, JApplet, JWindow, JDialog, Light weight containers – JPanel, A simple swing application, Overview of several swing components- JButton, JToggleButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JList, JComboBox, JMenu, Java's Graphics capabilities – Introduction, Graphics contexts and Graphics objects, color control, Font control, Drawing lines, rectangles and ovals, Drawing arcs, Layout management - Layout manager types – border, grid, flow, box.

Event Handling - Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Semantic and Low-level events, Examples: handling a button click, handling mouse and keyboard events, Adapter classes.

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

TEXT BOOKS:

1. Java: the complete reference, 8th edition, Herbert Schildt, TMH.
2. Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education / Java: How to Program
P.J.Deitel and H.M.Deitel, 8th edition, PHI.

REFERENCE BOOKS:

1. Java Programming, D.S.Malik, Cengage Learning.
2. Core Java, Volume 1-Fundamentals, eighth edition, Cay S.Horstmann and Gary Cornell, Pearson Education.
3. An introduction to Java programming and object oriented application development, R.A. Johnson- Cengage Learning.
4. Advanced Programming in Java2, K.Somasundaram, Jaico Publishing House.
5. Programming in Java, S.Malhotra and S.Choudhary, Oxford Univ. Press.
6. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
7. Object Oriented Programming through Java, P.Radha Krishna, Universities Press.

DATABASE MANAGEMENT SYSTEMS LAB

Objectives:

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

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.

List of Sample Problems:

1) Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

2) Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.

Example:- Select the roll number and name of the student who secured fourth rank in the class.

3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

4) Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

5) i) Creation of simple PL/SQL program which includes declaration section, executable section and exception –handling section (Ex. Student marks can be selected from the table and printed for

those who secured first class and an exception can be raised if no records were found)

ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.

8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.

9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.

10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.

11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.

12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

Example Problems:

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

CLIENT_MASTER :(CLENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE)

PRODUCT_MASTER : (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNITMEASURE, QTY_ON_HAND, REORDERLVL, SELLPRICE, COSTPRICE)

SALESMAN_MASTER: (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, SLAMT, TGTTOGET, YTDSALES, RESALES)

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

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

i) Insert data into their respective table.

ii) Exercise on retrieving records from table.

a. Find out the names of all clients.

b. Retrieve the entire contents of the client_master table.

c. Retrieve the list of names, city and the state of all clients.

d. List the various products available from the Product_master table.

e. List all the clients who are located in Mumbai.

f. Find the names of salesmen who have a salary equal to Rs.3000.

iii. Exercise on updating records in a table

a. Change the city of client No 'C00005' to 'Bangalore'.

b. Change the BalDue of Client No 'C00001' to Rs.1000.

c. Change the cost price of 'Trousers' to Rs.950.00.

d. Change the city of the salesman to Pune.

iv. Exercise on deleting records in a table

a. Delete all sales men from the salesman_master whose salaries are equal to Rs.35000

b. Delete all products from product_master where the quantity on hand is equal to 100.

c. Delete from Client_Master where the column state holds the value 'Tamilnadu'.

v. Exercise on altering the table

a. Add a column called 'Telephone' of data type 'number' and size='10' to the Client Master table.

b. Change the size of Sell Price column in Product_Master to 10, 2.

vi. Exercise on deleting the table structure along with the data

a. Destroy the table Client_Master along with the data.

vii. Exercise on renaming the table

a. Change the name of the Salesman_Master table to Sman_mast.

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

Client_Master Product_Master Salesman_Master Sales_Order
Sales_Order_Details

i) Perform the following computations on table data:

a. List the names of all clients having 'a' as the second letter in their names.

b. List the clients who stay in a city whose first letter is 'M'.

c. List all clients who stay in 'Bangalore' or 'Mangalore'.

d. List all clients whose BalDue is greater than value 10000.

e. List all information from the Sales_Order table for orders placed in the month of June.

f. List the order information for Client No 'C00001' and 'C00002'.

- g. List products whose selling price is greater than 500 and less than or equal to 750.
- h. List products whose selling price is more than 500. Calculate a new selling price as, original selling price * .15. Rename the new column in the output of the above query as new_price.
- i. List the names, city and state of clients who are not in the state of 'Maharashtra'.
- j. Count the total no of orders.
- k. Calculate the average price of all the products.
- l. Determine the maximum and minimum products prices. Rename the output as max_price and min_price respectively.
- m. Count the no of products having price less than or equal to 500.
- n. List all the products whose Qty On Hand is less than reorder level.
- ii) Exercise on Date Manipulation
 - a. List the order number and day on which clients on placed their order.
 - b. List the months (in alphabets) and date when the orders must be delivered.
 - c. List the Order Date in the format 'DD-Month-YY'. E.g.12-February-02.
 - d. List the date, 15 days after today's date.
- iii). Exercises on using Having and Group by Clauses:
 - a. Print the description and total qty sold for each product.
 - b. Find the value of each product sold
 - c. Calculate the average qty sold for each client that has a maximum order value of 15000.00.
 - d. Find out the total of all the billed orders for the month of June.
- iv). Exercises on Joins and Correlation:
 - a) Find out the products, which have been sold to 'Ivan Bay Ross'.
 - b) Find out the products and their quantities that will have to be delivered in the current month.
 - c) List the product no and description of constantly sold products (i.e. rapidly moving products).
 - d) Find the names of clients who have purchased 'Trousers'.
 - e) List the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.
 - f) Find the products and their quantities for the orders placed by 'Ivan Bay Ross' and 'Mamta Muzumdar'.
 - g) Find the products and their quantities for the orders placed by Client No 'C00001' and 'C00002'.
- v). Exercise on Sub-queries:
 - a. Find the Product No and Description of non_moving products i.e. Products not being sold.
 - b. List the customer Name, Address1, Address2, City and Pin Code for the client who has placed order no 'O19001'.
 - c. List the client names that have placed orders before the month of May'02.
 - d. List if the product 'Lycra Top' has been ordered by any client and print the Client_no, Name to whom it was sold.
 - e. List the names of clients who have placed orders worth Rs.10, 000 or more.

3) Creating Views

4) Writing Assertions

5) Writing Triggers

6) Implementing Operations on relations (tables) using PL/SQL

Ex: Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named Areas(radius, area).

- 7) Creating FORMS.
- 8) Generating REPORTS.

TEXT BOOKS:

1. Introduction to SQL, Mastering the relational DB languages, IV th edition, Rick F. Vander Lans, Pearson ed., 2007.
2. Oracle PL/SQL, 3rd edition, Benjamin Rosenzweig, Elena Silvestrova, Pearson ed., 2004.
3. Oracle, Forms Developer's Hand Book. Albert Lulushi, Pearson ed.,
4. Oracle Database 11g PL/SQL Programming, M. McLaughlin, TMH.
5. Oracle Pl/Sql. Programming, IV Edition, Steven Feuerstein. SPD,rp-2007.
6. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.
7. Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL, Shah, PHI.

Additional Problems:

i)Databases :

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

Roadway Travels

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

The company wants to *computerize its operations* in the following areas:

- Reservations and Ticketing
- Cancellations

Reservations & Cancellation:

Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family). Cancellations are also directly handed at the booking office.

In the process of *computerization* of **Roadway Travels** you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You

should also develop query's using SQL to retrieve the data from the database.

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

Experiment 1: E-R Model

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example: **Entities:**

1. BUS
2. Ticket
3. Passenger

Relationships:

1. Reservation

2. Cancellation

PRIMARY KEY ATTRIBUTES:

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

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

Note: *The student is required to submit a document by writing the Entities and Keys to the lab teacher.*

Experiment 2: Concept design with E-R Model

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

Example: E-R diagram for bus

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

Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion.

There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes

(Composite, Multi-valued, and Derived) have different way of representation.

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

Passenger

Name Age Sex Address

Ticket_id

Passport ID

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

BUS

Bus NO Source

Destination

Age

Reservation

Ticket

Sex

Ticket NO

Departure Time

Date of Journey

Departure Time

Source

Destination

Experiment 4: Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will

not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

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

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

Passenger

Name Age Sex Address Passport ID

Passport ID Ticket_id

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

Experiment 5: Installation of Mysql and practicing DDL commands

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

Example for creation of a normalized "Passenger" table.

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

Similarly create all other tables.

Note: Detailed creation of tables is given at the end.

Experiment 6: Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

Inserting values into "Bus" table:

```
Insert into Bus values (1234,'hyderabad', 'tirupathi');
```

```
Insert into Bus values (2345,'hyderabad', 'Banglore');
```

```
Insert into Bus values (23,'hyderabad', 'Kolkata');
```

```
Insert into Bus values (45,'Tirupathi', 'Banglore');
```

```
Insert into Bus values (34,'hyderabad', 'Chennai');
```

Inserting values into "Passenger" table:

```
Insert into Passenger values (1, 45,'ramesh', 45,'M', 'abc123');
```

```
Insert into Passenger values (2, 78,'geetha', 36,'F', 'abc124');
```

```
Insert into Passenger values (45, 90,'ram', 30,'M', 'abc12');
```

```
Insert into Passenger values (67, 89,'ravi', 50,'M', 'abc14');
```

```
Insert into Passenger values (56, 22,'seetha', 32,'F', 'abc55');
```

Few more Examples of DML commands:

```
Select * from Bus; (selects all the attributes and display)
```

```
UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;
```

Experiment 7: Querying

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

Practice the following Queries:

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

Experiment 8 and Experiment 9: Querying (continued...)

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

1. Write a Query to display the Information present in the Passenger and cancellation tables.

Hint: Use UNION Operator.

2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1.

Hint: Use GROUP BY, WHERE and HAVING CLAUSES.

6. Find the total number of cancelled seats.

Experiment 10: Triggers

In this week you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger.

Practice triggers using the above database.

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger FOR EACH ROW**

BEGIN

IF NEW.TicketNO > 60 THEN

SET New.Ticket no = Ticket no;

ELSE

SET New.Ticketno = 0;

END IF;

END;

Experiment 11: Procedures

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

Eg: **CREATE PROCEDURE myProc()**

BEGIN

SELECT COUNT(Tickets) FROM Ticket WHERE age>=40;

End;

Experiment 12: Cursors

In this week you need to do the following: Declare a cursor that defines a result set.

Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done

CREATE PROCEDURE myProc(in_customer_id INT)

BEGIN

DECLARE v_id INT;

DECLARE v_name VARCHAR (30);

DECLARE c1 CURSOR FOR SELECT stdId,stdFirstname FROM students WHERE

stdId=in_customer_id;

OPEN c1;

FETCH c1 into v_id, v_name;

Close c1;

END;

Tables

BUS

Bus No: Varchar: PK (public key)

Source : Varchar

Destination : Varchar

Passenger

PPNO: Varchar(15) : PK

Name: Varchar(15)

Age: int (4)

Sex:Char(10) : Male / Female

Address: VarChar(20)

Passenger_Tickets

PPNO: Varchar(15) : PK

Ticket_No: Numeric (9)

Reservation

PNR_No: Numeric (9): FK

Journey_date : datetime (8)

No_of_seats : int (8)

Address : Varchar (50)

Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer

Status: Char (2): Yes / No

Cancellation

PNR_No: Numeric (9) : FK

Journey_date : datetime(8)

No_of_seats : int (8)

Address : Varchar (50)

Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer

Status: Char (2): Yes / No

Ticket

Ticket_No: Numeric (9): PK

Journey_date : datetime(8)

Age : int (4)

Sex:Char(10) : Male / Female

Source: Varchar

Destination: Varchar

Dep_time : Varchar

TEXT BOOKS:

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

JAVA AND LINUX PROGRAMMING LAB

Objectives:

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

List of Sample Problems:

1. a) Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.
 - b) The Fibonacci sequence is defined by the following rule:
The 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 nth value in the Fibonacci sequence.
 2. a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
 - b) Write a Java program to multiply two given matrices.
 - c) Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)
 3. Write a Java program to find both the largest and smallest number in a list of integers.
 4. Write a Java program to sort a list of names in ascending order.
 5. Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:
 - a) Reading a matrix. b) Addition of matrices.
 - c) Printing a matrix d) Multiplication of matrices.
 6. Write a Java Program to solve Towers of Hanoi problem.
 7. Implement the complex number ADT in Java using a class. The complex ADT is used to represent complex numbers of the form $c=a+ib$, where a and b are real numbers. The operations supported by this ADT are:
 - a) Reading a complex number.
 - b) Multiplication of complex numbers.
 - c) Writing a complex number.
 - d) Addition of Complex numbers.
 8. 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.
 9. a) Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
 - b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
 - c) Write a Java program that displays the number of characters, lines and words in a text file.
 - d) Write a Java program to change a specific character in a file.
- Note: Filename , number of the byte in the file to be changed and the new character are specified on the command line.

10. 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.
11. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not 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.
12. Write a Java program for handling Key events.

LINUX PROGRAMMING LAB

List of sample problems:

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 accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
5. Write a shell script to find factorial of a given integer.
6. Write an awk script to count the number of lines in a file that do not contain vowels.
7. Write a C program that makes a copy of a file using standard I/O and system calls.
8. 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.
9. Write C programs that illustrate communication between two unrelated processes using named pipe.
10. Write a C program in which a parent writes a message to a pipe and the child reads the message.
11. Write a C programs to transfer a large amount of data between processes, using
 - a) a pipe b) a FIFO c) a message queue.

TEXT BOOKS:

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

DATA WAREHOUSING AND DATA MINING

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 build basic terminology.
- 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.
- 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.

Outcomes:

- 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.

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, kmeans and k-medoids 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.

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:

1. Data Warehousing in the Real World – Sam Aanhory & Dennis Murray Pearson Edn Asia.
2. Insight into Data Mining, K.P.Soman, S.Diwakar, V.Ajay, PHI, 2008.
3. Data Warehousing Fundamentals – Paulraj Ponnaiah Wiley student Edition
4. The Data Warehouse Life cycle Tool kit – Ralph Kimball Wiley student edition
5. Building the Data Warehouse By William H Inmon, John Wiley & Sons Inc, 2005.
6. Data Mining Introductory and advanced topics –Margaret H Dunham, Pearson education
7. Data Mining Techniques – Arun K Pujari, 2nd edition, Universities Press.
8. Data Mining, V.Pudi and P.Radha Krishna, Oxford University Press.
9. Data Mining: Methods and Techniques, A.B.M Shawkat Ali and S.A.Wasimi, Cengage Learning.
10. Data Warehouse 2.0, The Architecture for the next generation of Data Warehousing, W.H.Inmon, D.Strauss, G.Neushloss, Elsevier, Distributed by SPD.

WEB TECHNOLOGIES

Objectives:

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

Outcomes

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

UNIT I

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads, Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies

File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories

UNIT II

XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, XHTML

Parsing XML Data - DOM and SAX Parsers in java.

UNIT III

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

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

INFORMATION SECURITY

Objectives:

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

UNIT I

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

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

Web Security: Web security considerations, Secure Socket Layer and Transport Layer Security, Secure electronic transaction **Intruders, Virus and Firewalls:** Intruders, Intrusion

detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls

Case Studies on Cryptography and security: Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability, Virtual Elections

TEXT BOOKS:

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

REFERENCE BOOKS:

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

2013-14

MR13PB50

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - II Year - II SEM

L T/P C
4 -/- 3

MANAGEMENT INFORMATION SYSTEMS (ELECTIVE-I)

Objectives:

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

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

Prerequisites

Organization working which the student will pursue in earlier classes.

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

Implementation, evaluation and maintenance of the MIS: Plan the implementation, acquire floor space and plan space layouts, organize for implementation, develop procedures for

implementation, train the operating personnel, computer related acquisitions, develop forms for data collection and information dissemination, develop the files, test the system, cutover, document the system, evaluate the MIS, control and maintain the system.

UNIT V

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

TEXT BOOKS:

1. Information systems for modern management, 3rd Edition by R.G Murdick, J.E Ross and J. R clagget, PHI-1994.
2. Management Information Systems, Managing the Digital Firm Edition by Kenneth C. Laudon, Jane P. Laudon, Pearson Education, 10th Edition.

REFERENCE BOOKS:

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

2013-14

MR13PA20

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - II Year - II SEM

L T/P C
4 -/- 3

SCRIPTING LANGUAGES (ELECTIVE-I)

Objectives:

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

UNIT I

Introduction to PERL and Scripting

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

UNIT II

Advanced perl

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

UNIT III

Advanced PHP Programming

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

UNIT IV

TCL

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

UNIT V

Python

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and PHP, J.Lee and B.Ware(Addison Wesley) Pearson Education.
2. Programming Python,M.Lutz,SPD.
3. PHP 6 Fast and Easy Web Development, Julie Meloni and Matt Telles, Cengage Learning Publications.
4. PHP 5.1, I.Bayross and S.Shah, The X Team, SPD.
5. Core Python Programming, Chun, Pearson Education.
6. Guide to Programming with Python, M.Dawson, Cengage Learning.
7. Perl by Example, E.Quigley, Pearson Education.
8. Programming Perl, Larry Wall, T.Christiansen and J.Orwant, O'Reilly, SPD.
9. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
- 10.PHP and MySQL by Example, E.Quigley, Prentice Hall (Pearson).
- 11.Pperl Power, J.P.Flynt, Cengage Learning.
12. PHP Programming solutions, V.Vaswani, TMH.

2013-14

MR13PA21

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - II Year - II SEM

L T/P C
4 -/- 3

COMPUTER GRAPHICS (ELECTIVE-I)

Objectives:

- To provide students with an understanding of the algorithms and theories that forms the basis of computer graphics and modeling.
- To give students skills necessary in the production of 2D & 3D models.

UNIT I

Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, Raster-scan systems, random scan systems, graphics monitors and work stations and inputDevices Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

UNIT II

2-D Geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems. 2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland – Hodgeman polygon clipping algorithm.

UNIT III

3-D Object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-spline curves, Bezier and B-spline surfaces. Basic illumination models, polygon rendering methods.

UNIT IV

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

UNIT V

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications

TEXT BOOKS:

1. “Computer Graphics C version”, Donald Hearn and M. Pauline Baker, Pearson education.
2. “Computer Graphics Principles & practice”, second edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.

REFERENCE BOOKS:

1. "Computer Graphics Second edition", Zhigand xiang, Roy Plastock, Schaum's outlines, Tata Mc Graw Hill edition.
2. "Procedural elements for Computer Graphics", David F Rogers, Tata Mc Graw hill, 2nd edition.
3. "Principles of Interactive Computer Graphics", Neuman and Sproul, TMH.
4. "Principles of Computer Graphics", Shalini, Govil-Pai, Springer.
5. "Computer Graphics", Steven Harrington, TMH.
6. Computer Graphics, F.S.Hill, S.M.Kelley, PHI.
7. Computer Graphics, P.Shirley, Steve Marschner & Others, Cengage Learning.
8. Computer Graphics & Animation, M.C.Trivedi, Jaico Publishing House.
9. An Integrated Introduction to Computer Graphics and Geometric Modelling, R.Goldman, CRC Press, Taylor & Francis Group.
10. Computer Graphics, Rajesh K.Maurya, Wiley India.

E-COMMERCE (ELECTIVE-I)

Objectives:

- Identify the major categories and trends of e-commerce applications.
- Identify the essential processes of an e-commerce system.
- Identify several factors and web store requirements needed to succeed in e-commerce.
- Discuss the benefits and trade-offs of various e-commerce clicks and bricks alternatives.
- Understand the main technologies behind e-commerce systems and how these technologies interact.
- Discuss the various marketing strategies for an online business.
- Define various electronic payment types and associated security risks and the ways to protect against them.

UNIT I

Electronic Commerce-Frame work, anatomy of E-Commerce applications, E-Commerce Consumer applications, E-Commerce organization applications. Consumer Oriented Electronic commerce - Mercantile Process models.

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

Intra Organizational Commerce - work Flow, Automation Customization and internal Commerce, Supply chain Management.

UNIT IV

Corporate Digital Library - Document Library, digital Document types, corporate Data Warehouses. Advertising and Marketing - Information based marketing, Advertising on Internet, on-line marketing process, market research.

UNIT V

Consumer Search and Resource Discovery - Information search and Retrieval, Commerce Catalogues, Information Filtering.

Multimedia - key multimedia concepts, Digital Video and electronic Commerce, Desktop video processing, Desktop video conferencing.

TEXT BOOK:

1. Frontiers of electronic commerce – Kalakata, Whinston, Pearson.

REFERENCE BOOKS:

1. E-Commerce fundamentals and applications Hendry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang, John Wiley.
2. E-Commerce, S.Jaiswal – Galgotia.
3. E-Commerce, Efrain Turbon, Jae Lee, David King, H.Michael Chang.
4. Electronic Commerce – Gary P.Schneider – Thomson.

DISTRIBUTED SYSTEMS (ELECTIVE II)

Objectives:

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

UNIT I

Characterization of Distributed Systems-Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models-Introduction, Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication, Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

UNIT II

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

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

UNIT III

Peer to Peer Systems-Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies-Squirrel, OceanStore, Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging. Coordination and Agreement-Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT IV

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

UNIT V

Security-Introduction, Overview of Security techniques, Cryptographic algorithms, Digital signatures, Case studies-Kerberos, TLS, 802.11 Wi-Fi. Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy casestudy, Release consistency and Munin case study, Other consistency models, CORBA case study- Introduction, CORBA RMI, CORBA Services.

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Distributed Computing, S.Mahajan and S.Shah, Oxford University Press.
2. Distributed Operating Systems Concepts and Design, Pradeep K.Sinha, PHI.
3. Advanced Concepts in Operating Systems, M Singhal, N G Shivarathri, TMH.
4. Reliable Distributed Systems, K.P.Birman, Springer.
5. Distributed Systems – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, Pearson Education.
6. Distributed Operating Systems and Algorithm Analysis, R.Chow, T.Johnson, Pearson.
7. Distributed Operating Systems, A.S.Tanenbaum, Pearson education.
8. Distributed Computing, Principles, Algorithms and Systems, Ajay D.Kshemakalyani and Mukesh Singhal, Cambridge, rp 2010.

SOFTWARE PROJECT MANAGEMENT (ELECTIVE-II)

The main goal of software development projects is to create a software system with a predetermined functionality and quality in a given time frame and with given costs. For achieving this goal, models are required for determining target values and for continuously controlling these values. This course focuses on principles, techniques, methods & tools for model-based management of software projects, assurance of product quality and process adherence (quality assurance), as well as experience-based creation & improvement of models (process management). The goals of the course can be characterized as follows:

1. Understanding the specific roles within a software organization as related to project and process management
2. Understanding the basic infrastructure competences (e.g., process modeling and measurement)
3. Understanding the basic steps of project planning, project management, quality assurance, and process management and their relationships

Objectives:

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

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

UNIT I

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

UNIT II

Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.

The old way and the new: The principles of conventional software engineering, principles of modern software management, transitioning to an iterative process.

UNIT III

Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.

Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts. Model based software architectures: A Management perspective and technical perspective.

UNIT IV

Work Flows of the process: Software process workflows, Inter trans workflows. Checkpoints of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments.

Iterative Process Planning:

Work breakdown structures, planning guidelines, cost and schedule estimating, Interaction planning process, Pragmatic planning.

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation: Automation Building Blocks, The Project Environment.

UNIT V

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

Future Software Project Management: Modern Project Profiles Next generation. Software economics, modern Process transitions.

Case Study : The Command Center Processing and Display System-Replacement(CCPDS-R)

TEXT BOOKS:

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

REFERENCE BOOKS:

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

2013-14

MR13PA25

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - II Year - II SEM

L T/P C
4 -/- 3

SOFT COMPUTING (ELECTIVE-II)

Objectives:

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

UNIT I

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

UNIT II

Artificial Neural Networks: Introduction, Basic models of ANN, important terminologies, Supervised Learning Networks, Perceptron Networks, Adaptive Linear Neuron, Back propagation Network.

Associative Memory Networks. Training Algorithms for pattern association, BAM and Hopfield Networks.

UNIT III

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

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERECE BOOKS :

1. Artificial Intelligence and SoftComputing- Behavioural and Cognitive Modeling of the Human Brain- Amit Konar, CRC press, Taylor and Francis Group.

COMPILER DESIGN (ELECTIVE-II)

Objectives:

In order to reduce the complexity of designing and building computers, nearly all of these are made to execute relatively simple commands (but do so very quickly). A program for a computer must be built by combining these very simple commands into a program in what is called machine language. Since this is a tedious and error prone process most programming is, instead, done using a high-level programming language. This language can be very different from the machine language that the computer can execute, so some means of bridging the gap is required. This is where the compiler comes in.

A compiler translates (or compiles) a program written in a high-level programming language that is suitable for human programmers into the low-level machine language that is required by computers.

During this process, the compiler will also attempt to spot and report obvious programmer mistakes.

- To understand the process by which programs written in high-level languages are translated and executed. The main objective of this course is to gain an in depth understanding of the above process.
- Most complex software systems are not monolithic; they are programmable using special purpose languages. An understanding of the language translation process thus plays an important role in the design of real life systems.
- To understand the major phases of compilation, particularly lexical analysis, parsing, semantic analysis, and code generation.
- To understand the use of formal attributed grammars for specifying the syntax and semantics of programming languages, and their impact on compiler design.
- To describe the steps and algorithms used by language translators.
- To recognize the underlying formal models such as finite state automata, Push-down automata and their connection to language definition through regular expressions and grammars.
- To discuss the effectiveness of optimization.
- To know about the powerful compiler generation tools

Prerequisites

Students must have the prerequisite knowledge on the following topics:

- Programming Languages

UNIT I

Formal Language and Regular Expressions: Languages, Definition Languages regular expressions, Finite Automata – DFA, NFA. Conversion of regular expression to NFA, NFA to DFA. Applications of Finite Automata to Compiler Construction- lexical analysis, Construction of lexical analyze using LEX tool. Phases of Compilation and A simple One-Pass Compiler.

UNIT II

Context Free grammars and parsing: Context free grammars, derivation, parse trees, ambiguity, Application CFG in compilation-Preprocessing steps in Parsing, LL(1) parsing. Bottom up parsing handle pruning LR Grammar Parsing, LALR parsing, parsing ambiguous grammars, YACC programming specification.

UNIT III

Semantics: Syntax directed translation, S-attributed and L-attributed grammars, Intermediate code – abstract syntax tree, translation of simple statements and control flow statements. Context Sensitive features – Chomsky hierarchy of languages and recognizers. Type checking, type conversions, equivalence of type expressions, overloading of functions and operations.

UNIT IV

Run time storage: Storage organization, storage allocation strategies scope access to now local names, parameters, language facilities for dynamics storage allocation. Code optimization: Principal sources of optimization, optimization of basic blocks, peephole optimization

UNIT V

Global optimizations-flow graphs, Data flow analysis of flow graphs. Code generation: Machine dependent code generation, object code forms, generic code generation algorithm, Register allocation and assignment. Using DAG representation of Blocks.

TEXT BOOKS:

1. Compilers Principles, Techniques and Tools Aho, Ullman, Ravisethi, Pearson Education.
2. Introduction to Automata Theory, Languages and Computation - Hopcroft, Rajeev Motwani and Ullman, 3rd Edition, Pearson Education.

REFERENCE BOOKS:

1. Introduction to Theory of computation.Sipser, 2nd Edition, Cengage Learning.
2. Introduction to Formal Languages and Automata Theory and Computation – Kamala Krithivasan and Rama R, Pearson.
3. Modern Compiler Construction in C, Andrew W.Appel Cambridge University Press.
4. Compiler Construction, Louden, Cengage Learning, 1997.
5. Elements of Compiler Design, A.Meduna, Auerbach Publications, Taylor and Francis Group.
6. Principles of Compiler Design, V.Raghavan, TMH.
7. Engineering a Compiler, K.D.Cooper, L.Torczon, ELSEVIER.
8. Principles of Compiler Design, N.Prasad K.S, ELSEVIER.

WEB TECHNOLOGIES LAB

Objectives:

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

List of Sample Problems

1. Develop static pages (using Only HTML) of an online Book store. The pages should resemble: www.amazon.com. The website should consist the following pages.

Home page, Registration and user Login User Profile Page, Books catalog Shopping Cart, Payment By credit card Order Conformation

2. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

3. Create and save an XML document at the server, which contains 10 users information.

Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

*4. Bean Assignments

a. Create a JavaBean which gives the exchange value of INR(Indian Rupees) into equivalent American/Canadian/Australian Dollar value.

b. Create a simple Bean with a label - which is the count of number of clicks. Than create a BeanInfo class such that only the “count” property is visible in the Property Window.

c. Create two Beans-a)Keypad .b)DisplayPad .After that integrate the two Beans to make it work as a Calculator.

d. Create two Beans Traffic Light (Implemented as a Label with only three background colors-Red, Green, Yellow) and Automobile(Implemented as a TextBox which states its state/movement). The state of the Automobile should depend on the following Light Transition Table.

Light Transition Automobile State

Red ---> Yellow Ready

Yellow ---> Green Move

Green --> Red Stopped

5. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

6. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

7. Implement the “Hello World!” program using JSP Struts Framework.

8.Redo the problem 5 using PHP.

Additional Assignment Problems for the WT Lab.:

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

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

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

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

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

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

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

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

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

Value 1 Value 2 Result

+ =

Operator

Consider the following web application for implementation:

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

If name and password matches, serves a welcome page with user's full name. If name matches and password doesn't match, then serves "password mismatch" page If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application using:

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

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

DATA WAREHOUSING AND DATA MINING AND INFORMATION SECURITY LAB

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 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.

DATA WAREHOUSING AND DATA MINING LAB

List of Sample Problems:

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 banks 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: (Turn in your answers to the following tasks)

1. List all the categorical (or nominal) attributes and the real-valued attributes separately. (5 marks)
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. (5 marks)
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. (10 marks)
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 ? (10 marks)
5. Is testing on the training set as you did above a good idea ? Why or Why not ? (10 marks)
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? (10 marks)

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. (10 marks)
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.) (10 marks)
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 marks)
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? (10 marks)
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? (10 marks)

12.(Extra Credit): How can you convert a Decision Trees into "if-then-else rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one! Can you predict what attribute that might be in this dataset? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and oneR. (10 marks)

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)
- Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)
- Weka resources:
- Introduction to Weka (html version) (download ppt version)
- Download Weka
- Weka Tutorial
- ARFF format
- Using Weka from command line

INFORMATION SECURITY LAB

List of programs:

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
 - a. Ceaser cipher
 - b. Substitution cipher
 - c. Hill Cipher
4. Write a C program to implement the DES algorithm logic.
5. Write a JAVA program to implement the DES algorithm logic.
6. Write a Java program that contains functions, which accept a key and input text to be encrypted/decrypted. This program should use the key to encrypt/decrypt the input by using the triple Des algorithm. Make use of Java Cryptography package.
7. Write a JAVA program to implement the Blowfish algorithm logic.
8. Write a JAVA program to implement the Rijndael algorithm logic.
9. Write the RC4 logic in Java.
10. Using Java cryptography, encrypt the text "Hello world" using Blowfish. Create your own key using Java keytool.
11. Implement DES-2 and DES-3 using Java cryptography package.
12. Write a Java program to implement RSA algorithm.
13. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties(Alice) and the JavaScript application as the other party(Bob)

14. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
15. Calculate the message digest of a text using the MD5 algorithm in JAVA.
16. Explore the Java classes related to digital certificates.
17. Create a digital certificate of your own by using the Java keytool.
18. Write a Java program to encrypt users passwords before they are stored in a database table, and to retrieve them whenever they are to be brought back for verification.
19. Key generation (public and private key pair) can be performed using Java. Write a program which can do this.
20. Write a program in java, which performs a digital signature on a given text.
21. Study phishing in more detail. Find out which popular bank sites have been phished and how.

MOBILE APPLICATION DEVELOPMENT

Mobile application development is the process by which application software is developed for low-power handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones.

These applications can be pre-installed on phones during manufacture, downloaded by customers from various mobile software distribution platforms, or delivered as web applications using server-side or clientside processing (e.g. JavaScript) to provide an "application-like" experience within a Web browser.

Application software developers also have to consider a lengthy array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms.

Objectives:

On completion of this course the students should be able to:

1. Design, implement and evaluate a User Interface for a mobile application using J2ME.
2. Create a small but realistic working mobile application for small computing devices.
3. Categorize the challenges posed by developing mobile applications and be able to propose and evaluate and select appropriate solutions.

UNIT I

J2ME Overview

Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices *Small Computing Technology*: Wireless Technology, Radio Data Networks, Microwave Technology, Mobile Radio Networks, Messaging, Personal Digital Assistants

UNIT II

J2ME Architecture and Development Environment

J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2ME Style, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit

J2ME Best Practices and Patterns: The Reality of Working in a J2ME World, Best Practices

UNIT III

Commands, Items, and Event Processing

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

UNIT IV

Record Management System

Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener

JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions

JDBC and Embedded SQL: Model Programs, Tables, Indexing, Inserting Data into Tables, Selecting Data from a Table, Metadata, Updating Tables, Deleting Data from a Table, Joining Tables, Calculating Data, Grouping and Ordering Data, Subqueries, VIEWS

UNIT V

Generic Connection Framework

The Connection, Hypertext Transfer Protocol, Communication Management Using HTTP Commands, Session Management, Transmit as a Background Process

TEXT BOOKS:

1. J2ME: The Complete Reference, James Keogh, Tata Mc Graw Hill.
2. Programming for Mobile and Remote Computers, G.T.Thampi, dreamtech press.

REFERENCE BOOKS:

1. Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, Pearson Education, 2004
2. Beginning Java ME Platform, Ray Rischpater, Apress, 2009
3. Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, A press, 2005
4. Kicking Butt with MIDP and MSA: Creating Great Mobile Applications, 1st edition, J.Knudsen, Pearson.

SOFTWARE TESTING METHODOLOGIES

Objectives:

- To understand different software testing techniques and strategies.

UNIT I

Introduction: - Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs.

Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT II

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

UNIT III

Domain Testing:-domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, 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

Logic Based Testing:- overview, decision tables, path expressions, kv charts, specifications.

State, State Graphs and Transition testing:- state graphs, good & bad state graphs, state testing, Testability tips.

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

OBJECT ORIENTED ANALYSIS AND DESIGN (using UML)

Object-oriented Analysis and Design (OOAD) is a modeling paradigm mainly used in computer programming. Prior to the rise of OOAD, the dominant paradigm was procedural programming, which emphasized the use of discrete reusable code blocks that could stand on their own, take variables, perform a function on them, and return values.

The object-oriented paradigm assists the programmer to address the complexity of a problem domain by considering the problem not as a set of functions that can be performed but primarily as a set of related, interacting Objects. The modeling task then is specifying, for a specific context, those Objects (or the Class the Objects belongs to), their respective set of Properties and Methods, shared by all Objects members of the Class. For more discussion, see object-oriented analysis and design and object-oriented programming. The description of these objects is a schema. As an example, in a model of a Payroll System, a Company is an Object. An Employee is another Object.

Employment is a Relationship or Association. An Employee Class (or Object for simplicity) has Attributes like Name, Birth date, etc. The Association itself may be considered as an Object, having Attributes, or Qualifiers like Position, etc. An Employee Method may be Promote, Raise, etc. The Model description or Schema may grow in complexity to require a Notation. Many notations have been proposed, based on different paradigms, diverged, and converged in a more popular one known as UML. An informal description or a Schema notation is translated by the programmer or a CASE tool in the case of Schema notation (created using a Module specific to the CASE tool application) into a specific programming language that supports object-oriented programming (or a Class Type), a declarative language or into a database schema.

Objectives

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

Outcomes:

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

UNIT I

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

UNIT II

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams. Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT III

Basic Behavioral Modeling-I: Interactions, Interaction diagrams.

Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

ETHICAL HACKING (ELECTIVE-III)

Objectives:

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

UNIT I

Introduction to Ethical Hacking, Ethics, and Legality: Ethical Hacking Terminology, Different Types of Hacking Technologies, Different Phases Involved in Ethical Hacking and Stages of Ethical Hacking: Passive and Active Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks, Hacktivism, Types of Hacker Classes, Skills Required to Become an Ethical Hacker, Vulnerability Research, Ways to Conduct Ethical Hacking, Creating a Security Evaluation Plan, Types of Ethical Hacks, Testing Types, Ethical Hacking Report Footprinting and Social Engineering: Footprinting , Information Gathering Methodology, Competitive Intelligence ,DNS Enumeration Who is and ARIN Lookups, Types of DNS Records, Traceroute, E-Mail Tracking , Web Spiders, Social Engineering, Common Types Of Attacks, Insider Attacks, Identity Theft, Phishing Attacks, Online Scams, URL Obfuscation, Social-Engineering Countermeasures.

UNIT II

Scanning and Enumeration

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

System Hacking

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

UNIT III

Trojans, Backdoors, Viruses, and Worms

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

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

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

UNIT IV

Hacking Web Servers, Web Application Vulnerabilities, and Web-Based Password Cracking Techniques

Hacking Web Servers, Types of Web Server Vulnerabilities, Attacks against Web Servers, IIS Unicode Exploits, Patch Management Techniques, Web Server Hardening Methods Web Application Vulnerabilities, Objectives of Web Application Hacking, Anatomy of an Attack, Web Application Threats, Google Hacking, Web Application Countermeasures Web-Based Password Cracking Techniques, Authentication Types, Password Cracker, Password Attacks: Classification, Password-Cracking Countermeasures

SQL Injection and Buffer Overflows SQL Injection, Steps to Conduct SQL Injection, SQL Server Vulnerabilities, SQL Injection Countermeasures Buffer Overflows, Types of Buffer Overflows and Methods of Detection, Stack-Based Buffer Overflows, Buffer Overflow Mutation Techniques

UNIT V

Linux Hacking

Linux Basics, Compile a Linux Kernel, GCC Compilation Commands, Install Linux Kernel Modules, Linux Hardening Methods

Penetration Testing Methodologies: Security Assessments, Penetration Testing Methodologies, Penetration Testing Steps, Pen-Test Legal Framework, Automated Penetration Testing Tools, Pen-Test Deliverables

TEXT BOOK:

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

REFERENCE BOOKS:

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

2013-14

MR13PA33

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - III Year - I SEM

L T/P C
4 -/- 3

MULTIMEDIA AND RICH INTERNET APPLICATIONS (ELECTIVE-III)

Objectives:

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

UNIT I

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

UNIT II

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

UNIT III

Basic Video compression techniques, Case study: MPEG Video Coding I, Basic Audio compression techniques, Case study: MPEG Audio compression.

Web 2.0

What is web 2.0, Search, Content Networks, User Generated Content, Blogging, Social Networking, Social Media, Tagging, Social Marking, Rich Internet Applications, Web Services, Mashups, Location Based Services, XML, RSS, Atom, JSON, and VoIP, Web 2.0 Monetization and Business Models, Future of the Web.

UNIT IV

Rich Internet Applications(RIAs) with Adobe Flash : Adobe Flash- Introduction, Flash Movie Development, Learning Flash with Hands-on Examples, Publish your flash movie, Creating special effects with Flash, Creating a website splash screen, action script, web sources.

Rich Internet Applications (RIAs) with Flex 3 - Introduction, Developing with Flex 3, Working with Components, Advanced Component Development, Visual Effects and Multimedia,

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

2013-14

MR13PA34

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MCA - III Year - I SEM

L T/P C
4 -/- 3

INFORMATION RETRIEVAL SYSTEMS (ELECTIVE-III)

Objectives:

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

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

Web search basics. Web crawling and indexes. Link analysis.

TEXT BOOK:

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

REFERENCE BOOKS:

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

2013-14

MR13PA35

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)

MCA - III Year - I SEM

L T/P C

4 -/- 3

SEMANTIC WEB AND SOCIAL NETWORKS (ELECTIVE-III)

Objectives:

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

UNIT I

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

UNIT II

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

UNIT III

Logic, Rule and Inference Engines. Semantic Web applications and services, Semantic Search, elearning, Semantic Bioinformatics, Knowledge Base

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Semantic Web Technologies, Trends and Research in Ontology Based Systems, J.Davies, Rudi Studer, Paul Warren, John Wiley & Sons.

WEB SERVICES (ELECTIVE-IV)

Objectives:

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

UNIT I

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

Introduction to Web Services – The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services.

Web Services Architecture – Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication models, basic steps of implementing web services.

UNIT II

Core fundamentals of SOAP – SOAP Message Structure, SOAP encoding, SOAP message exchange models, SOAP communication and messaging, SOAP security.

Developing Web Services using SOAP – Building SOAP Web Services, developing SOAP Web Services using Java and Axis, limitations of SOAP.

UNIT III

Describing Web Services – WSDL – WSDL in the world of Web Services, Web Services life cycle, anatomy of WSDL definition document, WSDL bindings, WSDL Tools, limitations of WSDL.

Discovering Web Services – Service discovery, role of service discovery in a SOA, service discovery mechanisms, UDDI – UDDI Registries, uses of UDDI Registry, Programming with UDDI, UDDI data structures, Publishing API, Publishing, searching and deleting information in a UDDI Registry, limitations of UDDI.

UNIT IV

Web Services Interoperability – Means of ensuring Interoperability, Overview of .NET, Creating a .NET client for an Axis Web Service, Challenges in Web Services Interoperability.

Web Services Security – XML security framework, Goals of Cryptography, Hash Cipher, Symmetric Cipher, Asymmetric Cipher, XML encryption, Digital signature, Digital Certificate, XML Encryption, SAML, structure.

UNIT V

Overview of Service Oriented Architecture – SOA concepts, Key Service Characteristics, Technical Benefits of a SOA

SOA and Web Services – Web Services Platform, Service-Level Data Models, Discovery, Security and Interaction Patterns, Atomic and Composite services, Service-level communication and alternative transports.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

AD HOC AND SENSOR NETWORKS (ELECTIVE-IV)

Objectives:

- To understand the concepts of sensor networks
- To understand the MAC and transport protocols for adhoc networks
- To understand the security of sensor networks
- To understand the applications of adhoc and sensor networks

UNIT I

Introduction to Ad Hoc Wireless Networks: Characteristics of MANETs, Applications of MANETs, Challenges.

Routing in MANETsL: Topology-based versus Position-based approaches, Topology based routing protocols, Position based routing, Other Routing Protocols.

UNIT II

Data Transmission in MANETs: The Broadcast Storm, Multicasting, Geocasting

TCP over Ad Hoc Networks: TCP Protocol overview, TCP and MANETs, Solutions for TCP over Ad Hoc

UNIT III

Basics of Wireless Sensors and Applications: The Mica Mote, Sensing and Communication Range, Design Issues, Energy consumption, Clustering of Sensors, Applications

Data Retrieval in Sensor Networks: Classification of WSNs, MAC layer, Routing layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

UNIT IV

Security : Security in Ad hoc Wireless Networks, Key Management, Secure Routing, Cooperation in MANETs, Intrusion Detection Systems.

Sensor Network Platforms and Tools: Sensor Network Hardware, Sensor Network Programming Challenges, Node-Level Software Platforms

UNIT V

Operating System – TinyOS

Imperative Language: nesC, Dataflow style language: TinyGALS, Node-Level Simulators, ns-2 and its sensor network extension, TOSSIM

TEXT BOOKS:

1. Ad Hoc and Sensor Networks – Theory and Applications, *Carlos Corderio Dharma P. Aggarwal*, World Scientific Publications / Cambridge University Press, March 2006
2. Wireless Sensor Networks: An Information Processing Approach, *Feng Zhao, Leonidas Guibas*, Elsevier Science imprint, Morgan Kauffman Publishers, 2005, rp2009

REFERENCE BOOKS:

1. Adhoc Wireless Networks – Architectures and Protocols, C.Siva Ram Murthy, B.S.Murthy, Pearson Education, 2004

2. Wireless Sensor Networks – Principles and Practice, Fei Hu, Xiaojun Cao, An Auerbach book, CRC Press, Taylor & Francis Group, 2010
3. Wireless Ad hoc Mobile Wireless Networks – Principles, Protocols and Applications, Subir Kumar Sarkar, et al., Auerbach Publications, Taylor & Francis Group, 2008.
4. Ad hoc Networking, *Charles E.Perkins*, Pearson Education, 2001.
5. Wireless Ad hoc Networking, *Shih-Lin Wu, Yu-Chee Tseng*, Auerbach Publications, Taylor & Francis Group, 2007
6. Wireless Ad hoc and Sensor Networks – Protocols, Performance and Control, Jagannathan Sarangapani, CRC Press, Taylor & Francis Group, 2007, rp 2010.
7. Security in Ad hoc and Sensor Networks, Raheem Beyah, et al., World Scientific Publications / Cambridge University Press, , 2010
8. Ad hoc Wireless Networks – A communication-theoretic perspective, Ozan K.Tonguz, Gialuigi Ferrari, Wiley India, 2006, rp2009.
9. Wireless Sensor Networks – Signal processing and communications perspectives, Ananthram Swami, et al., Wiley India, 2007, rp2009.

DESIGN PATTERNS (ELECTIVE-IV)

Objectives:

- Understand the design patterns that are common in software applications.
- Understand how these patterns are related to Object Oriented design.

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

A Case Study: Designing a Document Editor : Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary.
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

Behavioral Patterns Part-II (cont'd): State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns.
What to Expect from Design Patterns, A Brief History, The Pattern Community An Invitation, A Parting Thought.

TEXT BOOK:

1. Design Patterns By Erich Gamma, Pearson Education

REFERENCE BOOKS:

1. Pattern's in JAVA Vol-I By Mark Grand, Wiley DreamTech.
2. Pattern's in JAVA Vol-II By Mark Grand, Wiley DreamTech.
3. JAVA Enterprise Design Patterns Vol-III By Mark Grand, Wiley DreamTech.
4. Head First Design Patterns By Eric Freeman-Oreilly-sp
5. Design Patterns Explained By Alan Shalloway, Pearson Education.
6. Pattern Oriented Software Architecture, F.Buschmann & others, John Wiley & Sons.

DISTRIBUTED DATABASES (ELECTIVE-IV)

Objectives:

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

UNIT I

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

UNIT II

Translation of Global Queries to Fragment Queries, Equivalence transformations for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parametric Queries.

Optimization of Access Strategies, A Framework for Query Optimization, Join Queries, General Queries

UNIT III

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

UNIT IV

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

UNIT V

Architectural Issues, Alternative Client/Server Architectures, Cache Consistency, Object Management, Object Identifier Management, Pointer Swizzling, Object Migration, Distributed Object Storage, Object Query Processing, Object Query Processor Architectures, Query Processing Issues, Query Execution, Transaction Management, Transaction Management in Object DBMSs, Transactions as Objects

Database Integration, Scheme Translation, Scheme Integration, Query Processing Query Processing Layers in Distributed Multi-DBMSs, Query Optimization Issues Transaction Management Transaction and Computation Model, Multidatabase Concurrency Control, Multidatabase Recovery, Object Orientation and Interoperability, Object Management

Architecture CORBA and Database interoperability, Distributed Component Object Model, COM/OLE and Database Interoperability, PUSH-Based Technologies

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Distributed Database Systems, Chanda Ray, Pearson.

MOBILE APPLICATIONS DEVELOPMENT LAB

Mobile Application Development (Through J2ME) LABORATORY

Objective:98

Week - 1: Installation of Java Wireless Toolkit (J2ME)

Week - 2 Working with J2ME Features

Week - 3 Threads & High Level UI

Week - 4 Working on Drawing and Images

Week - 5 Developing Networked Applications using the Wireless Toolkit

Week - 6 Authentication with a Web Server

Week - 7 & 8 Web Application using J2ME

Mobile Application Development (Through J2ME) LABORATORY

Objective:

In this lab, a student is expected to design, implement, document and present a mobile client/server system using standard Java and Java 2 Micro Edition (J2ME) platform. Specifically it is required to design and implement a system that consists mainly of a mobile client (MC) and a Proxy Server (PS). MC will be written in J2ME, MIDP 2.0, while PS will be written in standard Java. It is necessary to use a mobile phone emulator to develop and demonstrate the experiments.

It may be necessary to use other components or existing resources (servers) as needed. For instance a database local to PS or a web service available on the Internet that can be invoked by the PS.

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.

2) Next, download the **Java Wireless Toolkit** (formerly called J2ME Wireless Toolkit) from: <http://java.sun.com/products/sjwtoolkit/download.html>.

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;
```

```

public HelloWorld(){
super();
}
public void startApp(){
form = new Form("Hello World");
String msg = "Hello World!!!!!!";
form.append(msg);
display = Display.getDisplay(this);
display.setCurrent(form);
}
public void pauseApp(){ }
public void destroyApp(boolean unconditional){
notifyDestroyed();
}
}

```

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)

2.1 Create a program which creates to following kind of menu.

- * cut
- * copy
- * past
- * delete
- * select all
- * unselect all

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 after 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** from 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 how to make a SOCKET Connection from a J2ME Phone.

Many a 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

SOFTWARE TESTING METHODOLOGIES AND UML LAB

i)Software Testing :

Objectives:

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

Sample problems on testing:

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

Additional problems on testing:

1. Test the following using JUnit and CPPUnit:
 - i) Sorting problems ii)Searching problems iii)Finding gcd of two integers iv)Finding factorial of a number.
2. Test web based forms using HTML Unit.
3. Test database stored procedures using SQLUnit.

(Use sufficient number of test cases in solving above Problems)

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

ii)UML:

Objectives:

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

List of Sample Problems:

1. The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
2. Student has to take up another case study of his/her own interest and do the same whatever mentioned in first problem. Some of the ideas regarding case studies are given in reference books which were mentioned in theory syllabus can be referred for some idea.

Additional Problems on UML:

Unified Modeling Language Lab:

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

UML diagrams to be developed are:

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

Description for an ATM System

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

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

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

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
3. A customer must be able to make a transfer of money between any two accounts linked to the card.
4. A customer must be able to make a balance inquiry of any account linked to the card.
5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

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

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

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

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

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

REFERENCE BOOKS:

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