

**ACADEMIC REGULATIONS**  
**COURSE STRUCTURE AND DETAILED SYLLABUS**  
**MASTER OF COMPUTER APPLICATIONS**  
**(MCA R14 REGULATIONS)**



**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

Yenkapally, Moinabad Mandal, P.O.Himayath Nagar, R.R.Dist, Hyderabad-500 075

Fax & Phone No.910-8413-235753, Tel: 08413-235755, 201301

Website: [www.jbiet.edu.in](http://www.jbiet.edu.in); e-mail:[principal@jbiet.edu.in](mailto:principal@jbiet.edu.in)

# **J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**

**(Autonomous)**

## **Academic Regulations 2011 for MCA (Regular)**

(Effective for the students admitted into first year from the academic year 2011-2012)

### **1.0 ELIGIBILITY FOR ADMISSIONS:**

Admission to the above program shall be made subject to the eligibility, qualifications and specialization prescribed by the JNT University Hyderabad 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 the 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 A student, who fails to fulfill all the academic requirements for the award of the degree within six academic years from the year of his admission, shall forfeit his seat in MCA course.
- 2.3 The minimum clear instruction days for each semester shall be 90 days.
- 2.4 The student shall register for 119 credits and secure 119 credits.

### **3.0 ATTENDANCE:**

- 3.1 A student 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.
- 3.3 Condonation of shortage of attendance shall be granted only on genuine and valid reasons on representation by the student with supporting evidence.
- 3.4 Shortage of attendance below 65% shall in no case be condoned.
- 3.5 A student shall not be promoted to the next semester unless he fulfils the attendance requirements of the previous semester.

### **4.0 EVALUATION AND ACADEMIC REQUIREMENTS:**

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

- 4.1 For the theory subjects 60 marks shall be awarded based on the performance in the External Examination, 40 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 duration of 120 minutes with 4 questions to be answered out of 6 questions.
- 4.2 For practical subjects, 60 marks shall be awarded based on the performance in the End Semester Examinations, 40 marks shall be awarded based on the day-to-day performance as Internal Marks.
- 4.3 There shall be a seminar presentation in VI Semester. For Seminar there will be only internal evaluation of 50 marks. A student has to secure a minimum of 50% 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 requirements in a subject if he secures a minimum of 40% of marks in the End Examination and a minimum aggregate of 45% 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/she earns 40 credits from
- i) One Regular and Two Supplementary Examinations of I Year I Semester,
  - ii) One Regular and One Supplementary Examinations of I year II Semester and,
  - iii) One Regular Examination of II year I Semester,
- Irrespective of whether the candidate takes the examinations or not.
- 4.6 In case the candidate does not secure the minimum academic requirement in any subject (as specified in 4.4) he has to reappear for the End Examination in that subject. A candidate shall be given one chance to re-register for each subject provided the internal marks secured by a candidate are less than 50% and he has failed in the end examination. In such case candidate must re-register for the subject(s) and secure required minimum attendance. The attendance of re-registered subject(s) shall be calculated separately to decide upon 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 obtained in the previous attempt are nullified.
- 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 for 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 the candidate has failed in the end examination.
  - (iv) In the event of availing improvement of internal marks, the internal marks and the end examination marks, secured in the previous attempt are cancelled.
  - (v) For each subject, the candidate has to pay a fee equivalent to one-third of the semester fee.
- 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 Laboratory examination for MCA course must be conducted with two Examiners, one of them being Laboratory Class Teacher and second examiner shall be other than Class Teacher.
- 4.9 i) A student shall register and put up minimum attendance in all 119 credits and earns the 119 credits. Marks obtained in 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.

## **5.0 EVALUATION OF PROJECT WORK:**

Every candidate shall be required to submit thesis or dissertation after taking up a topic approved by the Project Review Committee.

- 5.1 A Project Review Committee (PRC) shall be constituted with Principal as chair person, Heads of all the Departments which are offering the PG programs and two other senior faculty members.
- 5.2 Registration of Project Work: A candidate is permitted to register for the project work after satisfying the attendance requirement of all the subjects (theory and practical subjects).
- 5.3 After satisfying 5.2, a candidate has to submit, in consultation with his project supervisor, the title, objective and plan of action of his project work to the Departmental Committee for its approval. Only after obtaining the approval of Departmental Committee the student can initiate the Project work.
- 5.4 If a candidate wishes to change his supervisor or topic of the project he can do so with approval of Departmental Committee. However, the Departmental Committee shall examine whether the change of topic/supervisor leads to a major change of his initial plans of project proposal. If so, his date of registration for the project work starts from the date of change of Supervisor or topic as the case may be.
- 5.5 A candidate shall submit status report (in a bound-form) in two stages at least with a gap of 3 months between them.
- 5.6 The work on the project shall be initiated in the beginning of the **third year second semester** and the duration of the project is for **one** semester. A candidate is eligible to submit his/her PG Project Work/ Thesis / Dissertation, only after successful completion of the theory and practical courses, after getting the approval of PRC, and not earlier than 20 weeks from the date of registration of such PG Project work. For the approval of PRC, the candidate shall submit the draft copy of thesis to the Principal/Director (through Head of the Department) and shall make an oral presentation before the PRC.
- 5.7 Three copies of the Project Thesis certified by the supervisor shall be submitted to the College / School / Institute.
- 5.8 The thesis shall be adjudicated by one examiner selected by the JBIET. For this, Principal of the College shall submit a panel of 5 examiners, who are

eminent in that field with the help of the concerned guide and head of the department.

- 5.9 If the report of the examiner is not favorable, the candidate shall revise and resubmit the Thesis, in the time frame as described by PRC. If the report of the examiner is unfavorable again, the thesis shall be summarily rejected.
- 5.10 If the report of the examiner is favorable, viva-voce examination shall be conducted by a board consisting of the supervisor, Head of the Department and the examiner who adjudicated the Thesis. 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.

If the report of the viva-voce is unsatisfactory, the candidate will retake the viva-voce examination after three months. If he fails to get a satisfactory report at the second viva-voce examination, he will not be eligible for the award of the degree.

#### **6.0 AWARD OF DEGREE AND CLASS:**

After a 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:

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%
Pass Class	Below 50% but not less than 45%

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

#### **7.0 WITHHOLDING OF RESULTS:**

If the student has any dues to the College or if any case of indiscipline is pending against him, the result of the student 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:**

Students 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 the subjects are offered, subject to a maximum of 2 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 or ambiguity in the interpretation of the above rules, the decision of the JBIET is final.
- 9.3 The JBIET 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 JBIET.
- 9.4 Wherever the word he, him or his occurs, it will also include she, her and her's.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

**MASTER OF COMPUTER APPLICATIONS**

**COURSE STRUCTURE**

**MCA 1<sup>st</sup> Year (I-Semester)**

<b>Code</b>	<b>Subject</b>	<b>T</b>	<b>P</b>	<b>C</b>
09MCC111	Mathematical Foundations of Computer Science	4	-	3
09MCC112	Computer Organization	4	-	3
09MCC113	Object Oriented Programming through C++	4	-	3
09MCC114	Probability and Statistics	4	-	3
09MCC115	Accounting and Financial Management	4	-	3
09MCC116	IT Workshop	-	2	2
09MCC117	Object Oriented Programming Lab through C++	-	2	2
09MCC118	Computer Organization Lab	-	2	2

**MCA 1<sup>st</sup> Year (II-Semester)**

<b>Code</b>	<b>Subject</b>	<b>T</b>	<b>P</b>	<b>C</b>
09MCC121	Data Structures and Algorithms	4	-	3
09MCC122	Operating Systems	4	-	3
09MCC123	Organization Structure and Personnel Management	4	-	3
09MCC124	Operations Research	4	-	3
09MCC125	English Language Communication Skills	4	-	3
09MCC126	Data Structures Lab through C++	-	2	2
09MCC127	Operating Systems Lab	-	2	2
09MCC128	English Language Communication Skills Lab	-	2	2

**MCA 2<sup>nd</sup> Year (I-Semester)**

<b>Code</b>	<b>Subject</b>	<b>T</b>	<b>P</b>	<b>C</b>
09MCC211	Database Management Systems	4	-	3
09MCC212	Computer Networks	4	-	3
09MCC213	Linux Programming	4	-	3
09MCC214	Software Engineering	4	-	3
09MCC215	Java Programming	4	-	3
09MCC216	Database Management Systems Lab	-	3	2
09MCC217	Java and Linux Programming Lab	-	3	2

## COURSE STRUCTURE

### MCA 2<sup>nd</sup> Year (II-Semester)

Code	Subject	T	P	C
	Data Warehousing and Data Mining	4	-	3
09MCC222	Web Technologies	4	-	3
09MCC221	Information Security	4	-	3
	<b>Elective – I</b>	4	-	3
09MCC224B	Management Information Systems			
09MCC224C	Scripting Languages			
	Computer Graphics			
	E-Commerce			
09MCC225A	<b>Elective – II</b>	4	-	3
	Distributed Systems			
	Software Project Management			
	Soft Computing			
	Compiler Design			
Practicals	Web Technologies Lab	-	3	2
	Data Warehousing and Data Mining and Information Security Lab	-	3	2

### MCA 3<sup>rd</sup> Year (I-Semester)

Code	Subject	T	P	C
	Mobile Application Development	4	-	3
	Software Testing Methodologies	4	-	3
	Object Oriented Analysis and Design (using UML)	4	-	3
	<b>Elective – III</b>	4	-	3
	Ethical Hacking			
	Multimedia and Rich Internet Applications			
	Information Retrieval Systems			
	Semantic Web and Social Networks			
	<b>Elective – IV</b>	4	-	3
	Web Services			
	Human Computer Interaction			
	Design Patterns			
	Distributed Databases			
Practicals	Mobile Applications Development Lab	-	3	2
	Software Testing Methodologies and UML Lab	-	3	2

### MCA 3<sup>rd</sup> Year (II-Semester)

Code	Subject	T	P	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, Quantifiers, universal quantifiers.

Predicates: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving.

**UNIT II**

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

Algebraic structures: Algebraic systems, Examples and general properties, Semi groups and monoids, groups, and sub groups, homomorphism, Isomorphism.

**UNIT III**

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

**UNIT IV**

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

**UNIT V**

Graph Theory: Representation of Graphs, DFS, BFS, Spanning Trees, Planar Graphs.

Graph Theory and Applications, Basic Concepts, Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

**TEXT BOOKS:**

1. Elements of Discrete Mathematics- A Computer Oriented Approach, C.L.Liu, D.P. Mohapatra,3<sup>rd</sup> edition,TMH.
2. Discrete Mathematics for Computer Scientists & Mathematicians, second edition, J.L.Mott, A. Kandel, T.P. Baker, PHI
3. Discrete and Combinatorial Mathematics- An Applied Introduction-5th Edition– Ralph. P.Grimaldi, Pearson Education.

**REFERENCE BOOKS:**

1. Discrete Mathematics and its applications, 6th edition, K.H.Rosen, TMH.
2. Discrete Mathematical Structures, Mallik and Sen, CengageLearning.
3. Discrete Mathematical Structures, Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, PHI/ Pearson Education.
4. Discrete Mathematics with Applications, Thomas Koshy, Elsevier.
5. Discrete Mathematics, Lovasz, Springer.
6. Logic and Discrete Mathematics, Grass Man and Tremblay, Pearson Education.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - I SEM

**COMPUTER ORGANIZATION**

**Objectives:**

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

**Outcomes:**

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

**UNIT I**

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

**UNIT II**

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

**UNIT III**

BASIC CPU ORGANIZATION-Introduction to CPU, Instruction formats-INTEL-8086 CPU architecture-Addressing modes - generation of physical address- code segment registers, Zero, one, two, and three address instructions.

**UNIT IV**

INTEL 8086 ASSEMBLY LANGUAGE INSTRUCTIONS-Data transfer instructions-input-output instructions, address transfer, Flag transfer, arithmetic, logical, shift, and rotate instructions.conditional and unconditional transfer, iteration control, interrupts and process control instructions, assembler directives, Programming with assembly language instructions.

## **UNIT V**

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

### **TEXT BOOKS:**

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

### **REFERENCE BOOKS:**

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Vth Edition, McGraw Hill.
2. Fundamentals of Computer Organization and Design, Sivarama P.Dandamudi, Springer Int. Edition.
3. Computer Organization and Architecture, William Stallings, 7th Edition, Pearson/PHI, 2007.
4. Digital Design, M. Morris Mano, PHI/Pearson Education.
5. Computer Organization and Design, D.A.Paterson and John L.Hennessy, Elsevier.
6. Computer Architecture and Organization, M.Murdock and V.Heuring, Wiley Inda.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - I SEM

**OBJECT ORIENTED PROGRAMMING THROUGH C++**

**Objectives:**

- To understand the various steps in Program development.
- To understand the basic concepts in oops Programming Language.
- To learn how to write modular and readable C Programs
- To learn to write programs (using object 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**

Different paradigms for problem solving need for OOP paradigm, classes and instances, fundamental characteristics of OOP (Alan key), differences between OOP and Procedure Oriented Programming.

C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statements- if, switch, while, for, do, break, continue, goto statements.

**UNIT II**

C++ Functions-Scope of variables, Parameter passing methods, Default arguments, inline functions, Recursive functions, Pointers to functions.

C++ Classes And Data Abstraction: Class definition, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction, ADT and information hiding.

**UNIT III**

Dynamic memory allocation and deallocation operators-new and delete, Dynamic creation and destruction of objects, Preprocessor directives, name spaces.

Polymorphism: Function overloading, Operator overloading, generic programming-necessity of templates, Function templates and class templates.

**UNIT IV**

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.

Virtual Functions And Run Time Polymorphism: Overriding, Static and Dynamic bindings, Base and Derived class virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes.

## **UNIT V**

C++ I/O: I/O using C functions, C++ Stream classes hierarchy, Stream I/O, File streams and String streams, File Operations, Overloading << and >> operators, Error handling during file operations, Formatted I/O.

Exception Handling: Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

### **TEXT BOOKS:**

1. C++, The Complete Reference, 4th Edition, Herbert Schildt, TMH.
2. Object Oriented Programming in C++, 4th Edition, R.Lafore, SAMS, Pearson Education

### **REFERENCE BOOKS:**

1. An Introduction to OOP, 3rd Edition, T. Budd, Pearson Education, 2008.
2. Programming Principles and Practice Using C++, B.Stroustrup, Addison- Wesley, Pearson Education.
3. Problem solving with C++, 6th Edition, Walter Savitch, Pearson Education, 2007..
4. The Art, Philosophy and Science of OOP with C++, R.Miller, SPD.
5. OOP in C++, 3rd Edition, T.Gaddis, J.Walters and G.Muganda, Wiley DreamTech Press.
6. An Introduction to OOP in C++ with applications in Computer Graphics, 2nd Edition, G.M.Seed, Springer.
7. Programming with ANSI C++, B.Trivedi, and Oxford 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,  $\chi^2$  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, Pearson Education / PHI.
2. Introduction to Probability and Statistics, 12th edition, W.Mendenhall, R.J.Beaver and B.M.Beaver, Thomson. (Indian edition).

**REFERENCE BOOKS:**

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

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

**MCA**

**I Year - I SEM**

**ACCOUNTING AND FINANCIAL MANAGEMENT**

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

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

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

**Objectives**

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

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

### **Prerequisites**

Some basic accounting knowledge and mathematics level is recommended.

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

Ability to analyze market competency in the global village To become a Master in accounting and financial management

### **UNIT I**

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

### **UNIT II**

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

### **UNIT III**

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

### **UNIT IV**

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

### **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

**REFERENCES:**

1. Prasanna Chandra, Financial Management, TMH, 2009
2. S.N.Maheshwari, Financial Accounting, Sultan Chand, 2009.
3. Tulsian, Financial Accounting, S Chand, 2009.
4. Khan and Jain: Financial Management, TMH, 2009
5. Gokul Sinha: Financial Statement Analysis, PHI, 2009
6. Bhat Sundhindra:Financial Management, Excel:2009
7. Jawaharlal:Accounting for Management, Himalaya, 2009



**IT WORKSHOP****Objectives:**

**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.

**Productivity tools** module would enable the students in crafting professional word documents, excel spread sheets and power point presentations. (**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 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.

### **MS Word**

**Week 9&10:** The mentor needs to give an overview of Microsoft (MS) word 2007: Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word. Give a task covering 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, Inserting table, using Drawing toolbar in word.

### **MS Excel**

**Week 11&12:** The mentor needs to tell the importance of MS office 2007 Excel as a Spreadsheet tool covering Accessing, overview of toolbars, saving excel files, Using help and resources., Also give a task that is covering the features like Gridlines, Format Cells, Summation, auto fill, Formatting Text.

### **MS Power Point**

**Week 13&14:** Students will be working on MS power point 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 d Power point. Students will be given model power point presentation which needs to be replicated (exactly how it's asked).

### **REFERENCES:**

1. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dreamtech
2. The Complete Computer upgrade and repair book,3rd edition Cheryl A Schmidt, WILEY Dreamtech
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
4. PC Hardware and A+Handbook – Kate J. Chase PHI (Microsoft)
5. LaTeX Companion – Leslie Lamport, PHI/Pearson.
6. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – CISCO Press, Pearson Education.
8. Troubleshooting, Maintaining and Repairing PCs, S.J.Bigelow, 5th edition,TMH.
9. Office 2007, C.Grover, M.Donald and E.A. Vander Veer, O' Reilly, SPD.
10. Step by Step 2007 Microsoft Office System, J.Cox and others (Microsoft), PHI.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - I SEM

**OBJECT ORIENTED PROGRAMMING LAB THROUGH C++**

**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

**List of Sample Problems/Experiments:**

1. Write a C++ program to find the sum of individual digits of a positive integer.
2. 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.
3. Write a C++ program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
4. Write C++ programs that use both recursive and non-recursive functions
  - a. To find the factorial of a given integer.
  - b. To find the GCD of two given integers.
  - c. To find the nth Fibonacci number.
5. Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.
6. Write a C++ program that uses functions
  - a. To swap two integers.
  - b. To swap two characters.
  - c. To swap two reals. Note: Use overloaded functions.
7. Write a C++ program to find both the largest and smallest number in a list of integers.
8. Write a C++ program to sort a list of numbers in ascending order.
9. Write a C++ program that uses function templates to solve problems-7&8.
10. Write a C++ program to sort a list of names in ascending order.
11. Write a C++ program to implement the matrix ADT using a class. The operations Supported by this ADT are:
  - a)Reading a matrix.
  - b) Printing a matrix.
  - c) Addition of matrices.
  - d) Subtraction of matrices.
  - e) Multiplication of matrices.
12. Implement the matrix ADT presented in the problem-11 using overloaded operators ( $<<$ ,  $>>$ ,  $+$ ,  $-$ ,  $*$ ) and templates.
13. Implement the complex number ADT in C++ 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) Writing a complex number.
  - c) Addition of Complex numbers.
  - d) Subtraction of complex numbers.
  - e) Multiplication of complex numbers.
  - f) Division of complex numbers.
14. Write a C++ program that overloads the  $+$  operator and relational operators (suitable) to perform the following operations:

- a) Concatenation of two strings.    B) Comparison of two strings.
15. Implement the complex number ADT in C++ 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) Writing a complex number.
  - c) Addition of Complex numbers.
  - d) Subtraction of complex numbers.
  - e) Multiplication of complex numbers.
  - f) Division of complex numbers.
- Note: 1. overload << and >> operators in part a and part b.  
 2. Overload +, -, \*, / operators in parts c, d, e and f
16. Write a template based C++ program that determines if a particular value occurs in an array Of values.
17. Write a C++ program that uses functions to perform the following operations:
- a. Insert a sub-string into the given main string from a given position.
  - b. Delete n characters from a given position in a given string.
18. Write a C++ program that uses a function to reverse the given character string in place, without any duplication of characters.
19. Write a C++ program to make the frequency count of letters in a given text.
20. Write a C++ program to count the lines, words and characters in a given text.
21. Write a C++ program to determine if the given string is a palindrome or not.
22. Write a C++ program to make frequency count of words in a given text.
23. 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.
24. 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.
25. Write a C++ program that counts the number of 1 bit in a given integer.
26. Write a C++ program to generate Pascal's triangle.
27. Write a C++ program to construct of pyramid of numbers.
28. Write a C++ program to compute the Sine series.
29. Write a C++ program that converts Roman numeral into an Arabic integer.
30. Write a C++ program which converts a positive Arabic integer into its Corresponding Roman numeral.
31. Write a C++ program to display the contents of a text file.
32. Write a C++ program which copies one file to another.
33. Write a C++ program to that counts the characters, lines and words in the text file.
34. Write a C++ 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.
35. Write a C++ program to reverse the first n characters in a file.
36. Write a C++ program that uses a function to delete all duplicate characters in the given string.
37. Write a C++ program that uses a function to convert a number to a character string.
38. Write a C++ program that uses a recursive function to find the binary equivalent of a given non-negative integer n.
39. Write a C++ program to generate prime numbers up to n using Sieve of Eratosthenes method.
40. Write a C++ program
- a) To write an object to a file.
  - b) To read an object from the file.
41. Write C++ programs that illustrate how the following forms of inheritance are supported:
- a) Single inheritance      b) Multiple inheritance
  - c) Multi level inheritance      d) Hierarchical inheritance
42. Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class.

43. Write a C++ program that illustrates how time polymorphism runs is achieved using virtual functions.
44. Write a C++ program that illustrates the role of virtual base class in building class hierarchy.
45. Write a C++ program that illustrates the role of abstract class in building class hierarchy.

## TEXT BOOKS:

1. Mastering C++, K.R.Venu Gopal, Raj Kumar and T.Ravi Shankar, TMH.
2. C++ Programming, D.S.Malik, Cengage Learning.
3. Practical C++ Programming,S.Qualline,SPD.
4. Object Oriented Programming with C++, E.Balaguruswamy, 4<sup>th</sup> Edition, TMH,2008.
5. OOP with C++,S.Sahay,Oxford Higher Education.
6. C++ and OOP Paradigm,D.Jana,2<sup>nd</sup> Edition, PHI
7. Fundamentals of C++ Programming,S.Subramanian,Jaico Publishing House.
8. C++ Programming, Al Stevens,7<sup>th</sup> edition,Wiley India.
9. Learning C++, Eric Nagler, Jaico Publishing House.
10. C How to program,P.J.Deitel and H.M.Deitel,PHI.

## COMPUTER ORGANIZATION LAB

### **Objectives:**

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

### **List of Sample Problems/Experiments:**

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.
- How many chips are needed, and how should their address lines be connected to provide a memory capacity of 1024 bytes?
  - How many chips are needed to provide a memory capacity of 16K bytes?
11. A computer employs RAM chips of 256X8 and ROM chips of 1024 X 8. The computer needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, 10 for interface registers.
- How many RAM and ROM chips are needed?
  - Draw a memory-address map for the system.
  - 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.
- Formulate all pertinent information required to construct the cache memory.
  - 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.
- How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.
  - How many bits are there in the tag, index, block, and word fields of the address format?
  - How many blocks can the cache accommodate?
15. An address space is specified by 24 bits and the corresponding memory space by 16 bits.
- How many words are there in the address space?
  - How many words are there in the memory space?
  - 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:

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

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - II SEM

**DATA STRUCTURES AND ALGORITHMS**

**Objectives:**

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

**Outcomes:**

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

**UNIT I**

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

Linear data structures- Linear Lists, Sequential and Linked allocation ,The list ADT, array and linked Implementations, Singly Linked Lists-Operations-Insertion, Deletion, Doubly Linked

Lists- Operations- Insertion, Deletion, Stack ADT, definitions, operations, array and linked implementations, applications-infix to postfix conversion, recursion implementation, Queue ADT, definitions and operations ,array and linked Implementations

**UNIT II**

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

Graphs – Introduction, Basic Terminology, Graph Representations- Adjacency matrix, Adjacency lists, Adjacency multilists, 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 dead lines, 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, and Suffix tries.

### **TEXT BOOKS:**

1. Data structures and Algorithm Analysis in C++, Mark Allen Weiss,3<sup>rd</sup> 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.

### **REFERENCES:**

1. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Cengage Learning.
2. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
3. Data structures, algorithms and OOP, G.L.Heileman, TMH edition.
4. Data Structures using C++, D.S. Malik, Cengage Learning, India Edition.
5. Data structures with C++ Using STL, 2<sup>nd</sup> 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

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - II SEM

**OPERATING SYSTEMS**

**Objectives:**

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

**Outcomes:**

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

**UNIT I**

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

**UNIT II**

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

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

**UNIT III**

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

**UNIT IV**

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

System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance, Case Studies: Linux, Windows.

Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management

## UNIT V

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

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

### TEXT BOOKS:

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

### REFERENCE BOOKS:

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, Pearson/PHI
2. Operating Systems A concept-based Approach, 2nd Edition, D.M.Dhamdhere, TMH.
3. Principles of Operating Systems, B.L.Stuart, Cengage learning, India Edition.
4. Operating Systems, A.S.Godbole,2nd Edition, TMH
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems, G.Nutt, N.Chaki and S.Neogy, 3rd Edition, Pearson Education.
7. Operating Systems, R.Elmasri, A. G.Carrick and D.Levine, Mc Graw Hill.

# **J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**

## **(Autonomous)**

**MCA**

**I Year - II SEM**

### **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 programs 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**

**Introduction to Management & Organisation:** Concepts of Management and organization-nature, importance and Functions and Theories of Management, Systems Approach to Management, Leadership Styles, Social responsibilities of Management. Designing Organisational Structures: Basic concepts related to Organisation - Departmentation and Decentralisation, Types and Evaluation of mechanistic and organic structures of organisation and suitability.

## **UNIT-II**

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

## **UNIT-III**

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

## **UNIT-IV**

**Understanding Human Behaviour:** Personality – Johari Window- Transactional Analysis. Perception: Perceptual process, Development of Attitudes and Values, Understanding Group Dynamics, Team Effectiveness, Strategies to deal with conflicts and stress.

## **UNIT-V**

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

## **TEXT BOOKS**

1. A.R.Aryasri, Organisational Structure and Personnel Management, TMH, 2009.
2. Robbins: Organisational Behaviour, Pearson, 2008

## **REFERENCES**

1. Udai pareek: Understanding Organisational Behaviour, PHI, 2009
2. P.Subbarao, Management and Organisational Behaviour, HPH, 2009
3. Mamoria & Gankar, Personnel Management, HPH, 2009
4. VSP Rao, Organisational Behaviour, Excel, 2009
5. Lawrence R Jauch, R.Gupta & William F.Glueck: Business Policy and Strategic Management, Frank Bros.2009.
6. Sharma: Organisational Behaviour, Jaico, 2009.
7. Koontz & Weihrich: Essentials of Management, TMH, 2009
8. Mullins: Management and Organisational Behaviour, Pearson, 2009.
9. R Satyaraju & A.Parthasarathy: Management Text and Cases, PHI, 2009.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - II SEM

**OPERATIONS RESEARCH**

**Objectives:**

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

**Prerequisites:**

- Probability and Statistics

**Outcomes:**

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

**UNIT I**

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

**UNIT II**

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

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

**UNIT III**

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

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

**UNIT IV**

Dynamic programming. Characteristics of dynamic programming. Dynamic programming approach for Priority Management employment smoothening, capital budgeting, Stage Coach/Shortest Path, cargo loading 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.

**TEXT BOOKS:**

1. A.M. Natarajan, P. Balasubramani, A. Tamilarasi, "Operations Research", Pearson Education, 2005.
2. P Sankara Iyer, "Operations Research", Tata McGraw-Hill, 2008.

**REFERENCE BOOKS:**

1. R. Panneerselvam, "Operations Research", PHI-2e, 2006, rp2008.
2. N.V.S. Raju, "Operations Research", HI-TECH, 2002.
3. P. K. Gupta and D. S. Hira, "Operations Research", S. Chand & co., 2007.
4. J K Sharma, "Operations Research – Theory & Applications – 3e", Macmillan India Ltd, 2007.
5. Col. D. S. Cheema, "Operations Research", Laxmi Publications Ltd., 2005.
6. F.S. Hillier, G.J. Lieberman, "Introduction to Operations Research – 8ed", Tata McGraw-Hill, 2005, rp2007.
7. H.S. Kansa & 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. Operations Research, A.B.Rao,Jaico Publishers.

**DATA STRUCTURES LAB THROUGH C++****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.

**List of Sample Problems/Experiments:**

1. Write a C++ program to perform the following operations:
  - a) Create a Singly linked list of integers.
  - b) Delete an integer from the above singly linked list.
2. Write a C++ program to perform the following operations:
  - a) Create a doubly linked list of integers.
  - b) Delete an integer from the above doubly linked list
3. Write C++ programs to implement the following using an array.
  - a) Stack ADT b) Queue ADT
4. Write C++ programs to implement the following using a singly linked list.
  - a) Stack ADT b) Queue ADT
5. Write a C++ program to convert a given infix expression into postfix form using stack.
6. Write C++ programs to implement the deque (double ended queue) ADT using a doubly linked list and an array.
7. Write C++ programs that use recursive functions to traverse the given binary tree in
  - a) Preorder b) inorder and c) postorder
8. Write a C++ program to perform the following operations:
  - a) Insert an element into a binary search tree.
  - b) Delete an element from a binary search tree.
  - c) Search for a key element in a binary search tree.
9. Write C++ programs that use non-recursive functions to traverse the given binary tree in
  - a) Preorder b) inorder and c) postorder.
10. Write C++ programs for the implementation of bfs and dfs for a given graph.
11. Write C++ programs for implementing the following Searching methods:
  - a) Linear Search b) Binary Search
12. Write C++ programs for implementing the following sorting methods:
  - a) Bubble Sort b) Selection Sort c) Insertion Sort
13. Write C++ programs for implementing the following sorting methods:
  - a) Merge sort b) Quick sort c) Heap sort
14. Write a C++ program to perform the following operations
  - a) Insertion into a B-tree b) Deletion from a B-tree
15. Write a C++ program to perform the following operations
  - a) Insertion into an AVL-tree b) Deletion from an AVL-tree
16. Write a C++ program to implement all the functions of a dictionary (ADT) using hashing.
17. Write a C++ program for implementing Knuth-Morris- Pratt pattern matching algorithm.
18. Write a C++ program for generating Minimum cost spanning tree using Kruskal's algorithm.

**(Note: Use Class Templates in the above Programs)**

### **TEXT BOOKS:**

1. Data Structures A Pseudocode Approach with C++, India Edition, R.F.Gilberg and B.A.Forouzan, Cengage Learning.
2. Programming Principles and Practice using C++, B.Stroustrup,(Pearson education).
3. Data Structures and STL, W.J.Collins, Mc Graw Hill, International edition.
4. Data structures and Algorithms with OODesign patterns in C++, B.R.Priess, John Wiley& sons.
5. Advanced Data structures & Algorithms in C++, V.V.Muniswamy, Jaico Publishing House.
6. C++ for Programmers, P.J.Deitel and H.M.Deitel,PHI/Pearson.
7. Data structures via C++, A.M.Berman, and Oxford University Press.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - II Sem

**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 Sample Problems/Experiments:**

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. Write a C program to create a child process and allow the parent to display “Hello” and the child to display “Welcome” on the screen.
7. Simulate all page replacement algorithms
  - a) FIFO
  - b) LRU
  - c) LFU
  - Etc...
8. Simulate Paging Technique of memory management.
9. Write C programs that make a copy of a file using i) standard I/O and ii) system calls.
10. Write C programs that count the number of blanks in a text file using i) standard I/O and ii) system calls.

**TEXT BOOKS:**

1. Operating Systems, P.P. Choudhury, PHI Learning Private Ltd.
2. Operating Systems, R.Chopra, S.Chand and Company Ltd.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

I Year - II SEM

**ENGLISH LANGUAGE COMMUNICATION SKILLS LAB**

**Objectives:**

The language Lab focuses computer-aided multi-media instruction and language acquisition to achieve the following targets: To expose the students to a variety of self-instructional, learner-friendly modes of language learning. To help the students cultivate the habit of reading passages from the computer monitor, thus providing them with the required facility to face computer-based competitive exams such GRE, TOEFL, GMAT etc. To enable them to learn better pronunciation through stress on word accent, intonation, and rhythm. To train them to use language effectively to face interviews, group discussions, public speaking. To initiate them into greater use of the computer in resume preparation, report writing, format-making etc. However, depending upon the available of infrastructure and budget, the above targets can also be achieved by procuring the minimum required equipment suggested for the establishment of a Conventional Lab the details of which are given below. The lab should cater to the needs of the students to build up their confidence to help them develop leadership qualities through their communicative competence.

**ENGLISH LANGUAGE LABORATORY PRACTICE**

**Syllabus:** The following course content is prescribed for the English Language Laboratory Practice

Introduction to Phonetics.

Introduction to Vowels and Consonants and associated Phonetic symbols.

Introduction to Accent, Intonation and Rhythm.

Situational Dialogues / Role Play.

Public Speaking.

Debate

Group discussions

Facing Interviews

Resume preparation

e-correspondence

**Minimum Requirement**

Computer aided multi media language lab with 30 systems with LAN facility.

Conventional Language Lab. with audio and video systems, speakers, headphones and a teacher console to accommodate 30 students.

### **Suggested Software:**

Cambridge Advanced Learners' Dictionary with exercises

The Rosetta Stone English Library

Clarity Pronunciation Power

Mastering English in Vocabulary, Grammar, Spellings,

Composition

Dorling Kindersley series of Grammar, Punctuation,

Composition etc.

Oxford Advanced Learner's Compass, 7th Edition

Language in Use, Foundation Books Pvt Ltd

Learning to Speak English - 4 CDs

Microsoft Encarta

Murphy's English Grammar, Cambridge

Time series of IQ Test, Brain-teasers, Aptitude Test etc.

English in Mind, Herbert Puchta and Jeff Stranks with

Meredith Levy, Cambridge

### **Books Suggested for English lab :**

1. **Developing Communication Skills** by Krishna Mohan & Meera Benerji (Macmillan)
2. **Speaking English Effectively** by Krishna Mohan & NP Singh (Macmillan)
3. **Better English Pronunciation** by JDO Connor (UBS – Cambridge)
4. **Oxford Practice Grammar with Answers**, John Eastwood, Oxford
5. **Handbook of English Grammar and Usage**, Mark Lester and Larry Beason, Tata McGraw-Hill
6. **A text book of English Phonetics for Indian Students** by T.Balasubramanian (Macmillan)
7. **Lingua TOEFL CBT Insider**, by Dreamtech
8. **TOEFL & GRE( KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)**
9. **English Skills for Technical Students**, WBSCTE with British Council, OL
10. **A Handbook of English for Competitive Examinations**, by B Shyamala Rao, Blakie Books, Chennai.

# **J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**

## **(Autonomous)**

**MCA**

**II Year - I SEM**

### **DATA BASE MANAGEMENT SYSTEMS**

#### **Objectives:**

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

#### **Outcomes:**

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

#### **UNIT I**

Data base 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 Dead Locks, Specialized Locking Techniques – Concurrency Control without Locking.

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

## **UNIT V**

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

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

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

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

### **TEXT BOOKS:**

1. Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, TMH, 3<sup>rd</sup> 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.Navarthe, Pearson Education, 2008.

### **REFERENCE BOOKS:**

1. Introduction to Database Systems, C.J.Date, Pearson Education.
2. Database Management System Oracle SQL and PL/SQL,P.K.Das Gupta,PHI.
3. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning,2008.
4. Data base Systems, A Practical approach to Design Implementation and Management Fourth edition, Thomas Connolly, Carolyn Begg, Pearson education.
5. Sql & Pl/Sql for Oracle 10g, Black Book, Deshpande, Dream Tech,2007.
6. Fundamentals of Relational Database Management Systems,S.Sumathi,S.Esakkirajan, Springer.
7. Introduction to Database Management, M.L.Gillenson and others,Wiley Student Edition.
8. Database Development and Management, Lee Chao,Auerbach Publications,Taylor & Francis Group.

# **J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**

## **(Autonomous)**

**MCA**

**II Year - I SEM**

### **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**

Introduction to Networks, internet, protocols and standards, the OSI model, layers in OSI model, TCP/IP suite, Addressing, Analog and digital signals.

Physical Layer: digital transmission, multiplexing, transmission media, circuit switched networks, Datagram networks, virtual circuit networks, switch and Telephone network.

#### **UNIT II**

Data link layer: Introduction, Block coding, cyclic codes, checksum, framing, flow and error control, Noiseless channels, noisy channels, HDLC, point to point protocols

Medium Access sub layer: Random access, controlled access, channelization, IEEE standards, Ethernet, Fast Ethernet, Giga-Bit Ethernet, wireless LANs.

#### **UNIT III**

Connecting LANs, backbone networks and virtual LANs, Wireless WANs, SONET, frame relay and ATM.

Network Layer: Logical addressing, internetworking, tunneling, address mapping, ICMP, IGMP, forwarding, uni-cast routing protocols, multicast routing protocols.

#### **UNIT IV**

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

#### **UNIT V**

Application Layer – Domain name space, DNS in internet, electronic mail, FTP, WWW, HTTP, SNMP, multi-media, network security

## **TEXT BOOKS:**

1. Data Communications and Networking – Behrouz A. Forouzan, Fourth Edition TMH, 2006.
  2. Computer Networks -- Andrew S Tanenbaum, 4th Edition. Pearson Education/PHI

## **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 and Communication Networks Nader F. Mir, Person Education
  4. Computer Networking: A Top-Down Approach Featuring the Internet, James F.Kurose, K.W.Ross, 3rd Edition, Pearson Education.



**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - I SEM

**LINUX PROGRAMMING**

**Objectives:**

- To understand the LINUX system structure.
- To understand and use command line shell.
- To make effective use of UNIX utilities and Shell scripting language such as bash.
- To produce programs similar to standard UNIX utilities such as ls, mv, cp etc. using Unix system calls.
- To develop the skills necessary for Unix systems programming including file system programming, process and signal management, and inter process 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, functions, using system commands in awk.

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

**UNIT II**

Linux Files: File Concept, File System Structure,Inodes, File types, The standard I/O (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputc, putc, putchar, fgets, gets etc.), formatted I/O, stream errors, kernel support for files, System calls, library functions, file descriptors, low level file access - usage of open, creat, read, write, close, lseek, stat family, umask, dup, dup2, fcntl, file and record locking.

file and directory management - chmod, chown, links(soft links & hard links - unlink, link, symlink), mkdir, rmdir, chdir, getcwd, Scanning Directories-opendir, readdir, closedir, rewinddir, seekdir, telldir functions.

**UNIT III**

Linux Process – Process concept, Kernel support for process, process attributes, process hierarchy,process states,process composition, process control - process creation, waiting for a process, process termination, zombie process,orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, system.

Linux 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, FIFOs, Introduction to three types of IPC(Linux)-message queues, semaphores and shared memory.

Message Queues- Kernel support for messages, Linux APIs for messages, client/server example.

Semaphores-Kernel support for semaphores, Linux APIs for semaphores, file locking with semaphores.

Shared Memory- Kernel support for shared memory, Linux APIs for shared memory, semaphore and shared memory example.

#### **UNIT V**

Multithreaded Programming – Differences between threads and processes,Thread structure and uses,Threads and Lightweight Processes,POSIX Thread APIs, Creating Threads,Thread Attributes,Thread Synchronization with semaphores and with Mutexes,Example programs.

Sockets: Introduction to Linux Sockets, Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs.

#### **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, 4<sup>th</sup> Edition, N.Matthew, R.Stones,Wrox, Wiley India Edition, rp-2008.

#### **REFERENCE BOOKS:**

1. Linux System Programming, Robert Love, O'Reilly, SPD, rp-2007.
2. Unix Network Programming, W.R.Stevens, PHI.
3. Unix for programmers and users, 3<sup>rd</sup> Edition, Graham Glass, King Ables, Pearson Education, 2003.
4. Advanced Programming in the UNIX environment, 2<sup>nd</sup> Edition, W.R.Stevens, Pearson Education.
5. System Programming with C and Unix, A.Hoover, Pearson.
6. Unix Programming, Kumar Saurabh, 1<sup>st</sup> Edition, Wiley India pvt Ltd.
7. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - I SEM

**SOFTWARE ENGINEERING**

**Objectives:**

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

**Outcomes:**

- Ability to identify the minimum requirements for the development of application.
- Ability to develop, maintain, efficient, reliable and cost effective software solutions
- Ability to 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.

McGrawHill International Edition, 2005

2. Software Engineering, Ian Sommerville, seventh edition, Pearson education,2004.

### **REFERENCE BOOKS:**

1. Software Engineering,A Precise Approach,Pankaj Jalote,Wiley India,2010.

2. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008

3. Fundamentals of Software Engineering,Rajib Mall,PHI, 2005

4. Software Engineering, Principles and Practices,Deepak Jain,Oxford University Press.

5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.

6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition , 2006.

7. Software Engineering Foundations,Yingxu Wang,Auerbach Publications,2008.

8. Software Engineering Principles and Practice,Hans Van Vliet,3<sup>rd</sup> edition,John Wiley & Sons Ltd.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - I SEM

**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**

**Data structures creation and manipulation in java** – Introduction to Java Collections, Overview of Java Collection framework, Commonly used Collection classes – ArrayList, LinkedList, HashSet, HashMap, TreeMap, Collection Interfaces – Collection, Set, List, Map, Legacy Collection classes – Vector, Hashtable, Stack, Dictionary (abstract), Enumeration interface, Iteration over Collections – Iterator interface, ListIterator interface. Other Utility classes – StringTokenizer, Formatter, Random, Scanner, Observable, Using java.util. **Files** – streams- byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class, Using java.io. **Networking in Java** – Introduction,

Manipulating URLs,Ex. Client/Server Interaction with Stream Socket Connections,Connectionless Client/Server Interaction with Datagrams,Using 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, JDialo, 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, 7<sup>th</sup> edition, Herbert Schildt, TMH.
2. Java for Programmers,P.J.Deitel and H.M.Deitel,Pearson education / Java: How to Program P.J.Deitel and H.M.Deitel ,8<sup>th</sup> edition,PHI.

#### REFERENCES:

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. Starting out with Java,T.Gaddis,dreamtech India Pvt. Ltd.
6. Object Oriented Programming with Java, R.Buyya,S.T.Selvi,X.Chu,TMH.
7. Object Oriented Programming through Java,P.Radha Krishna,Universities Press.
8. An introduction to programming and OO design using Java,J.Nino,F.A.Hosch, John Wiley&Sons.
9. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
10. Maurach's Beginning Java2,D.Lowe,J.Murach,A.Stelman, SPD.
11. Programming with Java,M.P.Bhave,S.A.Patekar, Pearson Education

# J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY

## (Autonomous)

MCA

II Year - I SEM

### DATA BASE 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.

**Objective:** This lab enables the students to practice the concepts learnt in the subject DBMS 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 travel”. 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
- Ticketing
- Cancellations

#### Reservations:

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

#### Cancellation and Modifications:

Cancellations are also directly handled at the booking office. Cancellation charges will be charged.

*Wait listed tickets that do not get confirmed are fully refunded.*

#### Week1: E-R Model

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

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

#### Example: Entities:

1. BUS
2. Ticket
3. Passenger

#### PRIMARY KEY ATTRIBUTES:

1. Ticket ID (Ticket Entity)

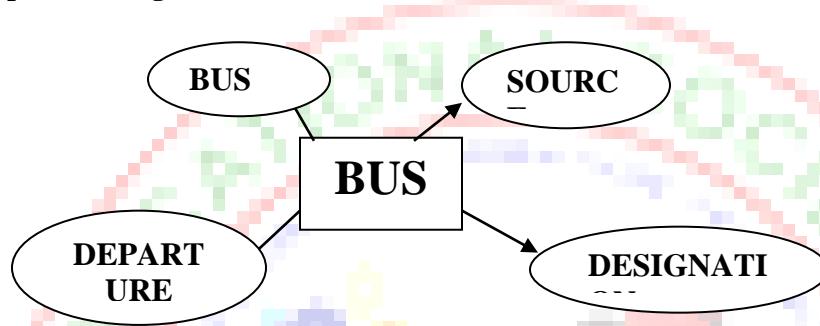
## 2. Passport ID (Passenger Entity)

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

### Week2: Concept design with E-R Model

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

#### Example: E-r diagram for bus



### Week3: Relational Model

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

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model.

Passenger				
Name	Age	Sex	Address	Passport ID

### Week 4: Normalization

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

### Week5: Installation of MySql and practicing DDL commands

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

Example for creation of a table.

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

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

## **Week6: Practicing DML commands**

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

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

**Inserting values into Bus table:** Insert into Bus values (1234,'hyderabad', 'tirupathi');

Insert into Bus values (2345,'hyderabd','Banglore');

**Inserting values into Bus table:**

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

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

**Few more Examples of DML commands:**

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

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

## **Week7: Querying**

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

**Practice the following Queries:**

1. Display unique PNR\_no of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Display the source and destination having journey time more than 10 hours.
5. Find the ticket numbers of the passengers whose name start with 'A' and ends with 'H'.
6. Find the names of passengers whose age is between 30 and 45.
7. Display all the passengers names beginning with 'A'
8. Display the sorted list of passengers names
9. Display the Bus numbers that travel on Sunday and Wednesday
10. Display the details of passengers who are traveling either in AC or NON\_AC(Using only IN operator)

## **Week8 and week9: Querying (continued...)**

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

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

## **Week10: Triggers**

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

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger**

**FOR EACH ROW**

**BEGIN**

**IF NEW.TicketNo > 60 THEN**

```

SET New.Ticketno = Ticket no;
ELSE
SET New.Ticketno = 0;
END IF;
END;

```

### **Week11: Procedures**

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

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

### **Week12: Cursors**

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

```

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

```

### **Tables**

#### **BUS**

Bus No: Varchar: Pk

Source : Varchar

Destination : Varchar

#### **Passenger**

PNR\_No : Numeric(9) : PK

Ticket\_No: Numeric (9)

Name: Varchar(15)

Age : int (4)

Sex:Char(10) : Male / Female

PPNO: Varchar(15)

#### **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.



**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - I SEM

**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/Experiments**

**I) JAVA PROGRAMS:**

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.
1. 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.
2. b) Write a Java program to multiply two given matrices.
2. c) Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)
3. Write a Java program to find both the largest and smallest number in a list of integers.
4. Write a Java program to illustrate method overloading.
5. Write a Java program that implements the Sieve of Eratosthenes to find prime numbers.
6. Write a Java program to sort a list of names in ascending order.
7. Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:
  - a) Reading a matrix.
  - b) Printing a matrix.
  - c) Addition of matrices.
  - d) Subtraction of matrices.
  - e) Multiplication of matrices.
8. Write a Java Program to solve Tower's of Hanoi problem .
9. Write a Java Program that uses a recursive function to compute ncr. (Note: n and r values are given.)
10. Write a Java program to perform the following operations:
  - a) Concatenation of two strings.
  - b) Comparison of two strings.
11. Implement the complex number ADT in Java using a class. The 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) Addition of complex numbers.
  - c) Subtraction of complex numbers.
  - d) Multiplication of complex numbers.

- b) Writing a complex number.      e) Multiplication of complex numbers.  
c) Addition of Complex numbers.    f) Division of complex numbers.

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

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

- a) Inserting a sub-string in to the given main string from a given position.  
b) Deleting n characters from a given position in a given string.

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

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

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

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

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

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

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

16. Write a Java program that:

- i) Implements stack ADT.  
ii) Converts infix expression into Postfix form  
iii) Evaluates the postfix expression.

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

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

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

19. Write a Java program for handling mouse events.

20. 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.

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

22 . Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result

back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)

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

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

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

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

25. Write a Java program that illustrates the following

a) Creation of simple package. b) Accessing a package.

c) Implementing interfaces.

26. Write Java programs that illustrates the following

a) Handling predefined exceptions

b) Handling user defined exceptions

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

a) Linear search      b) Binary search

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

a) List ADT

29. Write Java program to implement the following using an array.

a) Queue ADT

30. Write a Java program for handling Key events.

31. Write a Java program that uses both stack and queue to test whether the given string is a palindrome.

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

a) Stack ADT              b) Queue ADT

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

a) Bubble sort

b) Selection sort

c) Insertion

d) Quick sort

## II) LINUX PROGRAMMING:

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

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

2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a c program that makes a copy of a file using standard I/O and system calls.
11. Implement in C the following Unix commands using System calls  
A . cat B. ls C. mv
12. Write a program that takes one or more file/directory names as command line input and reports the following information on the file.
  - A. File type.
  - B. Number of links.
  - C. Time of last access.
  - D. Read, Write and Execute permissions.
13. Write a C program to emulate the Unix ls -l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C program that demonstrates redirection of standard output to a file.Ex: ls > f1.
16. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
17. Write a C program to create a Zombie process.
18. Write a C program that illustrates how an orphan is created.
19. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls -l | sort.
20. Write C programs that illustrate communication between two unrelated processes using named pipe.
21. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.

22. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (21)) and displays them.

**Note: You may use packages like java.io, java.util, java.net, java.awt etc in solving the above Java problems.**

### TEXT BOOKS

1. Core Java 2, Vol I, Fundamentals, 7<sup>th</sup> Edition, C.Horstman, Gary Cornel, Pearson Education.
2. Core Java 2, Vol 2, Fundamentals, 7<sup>th</sup> Edition, C.Horstman, Gary Cornel, Pearson Education.
3. Introduction to Java programming, Sixth edition, Y.Daniel Liang, Pearson Education
4. Data structures and algorithms in Java, 2<sup>nd</sup> Edition, R.Laforte, Pearson Education.
5. Data Structures using Java, D.S.Malik and P.S. Nair, Cengage Learning.
6. Advanced Unix Programming, N.B.Venkateswarulu, BS Publications.
7. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.
8. Unix and Shell Programming, M.G. Venkatesh Murthy, Pearson Education, 2005.
9. Unix Shells by Example, 4th Edition, Ellie Quigley, Pearson Education.
10. Sed and Awk, O.Dougherty&A.Robbins,2<sup>nd</sup> edition,SPD.
11. Java Programming, J.Wigglesworth and P.McMillan, Cengage Learning.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - II SEM

**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, Major 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, Further Development of Data Cube Technology, From Data Warehousing to Data Mining

**Data Cube Computation and Data Generalization:** Efficient Methods for Data Cube Computation, Further Development of Data Cube and OLAP Technology, Attribute-Oriented Induction.

**UNIT III**

**Mining Frequent Patterns, Associations and Correlations:** Basic Concepts, Efficient and Scalable Frequent Itemset Mining Methods, Mining various kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining

**- Classification and Prediction:** Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Backpropagation, Support Vector Machines, Associative Classification, Lazy Learners, Other Classification Methods, Prediction, Accuracy and Error measures, Evaluating the accuracy of a Classifier or a Predictor, Ensemble Methods.

## UNIT IV

**Cluster Analysis Introduction :** Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Clustering High-Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis - Mining Streams, Time Series and Sequence Data: Mining Data Streams, Mining Time-Series Data, Mining Sequence Patterns in Transactional Databases, Mining Sequence Patterns in Biological Data, Graph Mining, Social Network Analysis and Multirelational Data Mining.

## UNIT V

**Mining Object, Spatial, Multimedia, Text and Web Data:** Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Spatial Data Mining, Multimedia Data Mining, Text Mining, Mining the World Wide Web. - **Applications and Trends in Data Mining:** Data Mining Applications, Data Mining System Products and Research Prototypes, Additional Themes on Data Mining and Social Impacts of Data Mining.

### TEXT BOOKS:

1. Data Mining – Concepts and Techniques - Jiawei Han & Micheline Kamber, Morgan Kaufmann Publishers, 2<sup>nd</sup> Edition, 2006.
2. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.

### REFERENCES:

1. Data Warehousing in the Real World – Sam Aanhory & Dennis Murray Pearson Edn Asia.
2. Insight into Data Mining,K.P.Soman,S.Diwickar,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,2<sup>nd</sup> 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.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - II SEM

**WEB TECHNOLOGIES**

**Objectives:**

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

**Outcomes:**

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

**UNIT-I:**

**HTML Common tags-** List, Tables, images, forms, Frames; Cascading Style sheets;

Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script

**XML:** Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

**UNIT-II:**

**Java Beans:** Introduction to Java Beans, Advantages of Java Beans, BDK

Introspection, Using Bound properties, Bean Info Interface, Constrained properties

Persistence, Customizes, Java Beans API, Introduction to EJB's

**UNIT-III:**

**Web Servers and Servlets:** Tomcat web server, Introduction to Servlets: Lifecycle of a Serverlet, JSDK, The Servlet API, The javax.servelet Package, Reading Servelet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues,

**Introduction to JSP:** The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

**UNIT-IV:**

**JSP Application Development:** Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations

**UNIT V:**

**Database Access:** Database Programming using JDBC, Studying javax.sql.\* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

**TEXT BOOKS:**

1. Web Programming, building internet applications, Chris Bates 2<sup>nd</sup> edition, WILEY Dreamtech (UNITs 1,2 ,3)
2. The complete Reference Java Seventh Edition by Herbert Schildt. TMH.
3. Java Server Pages –Hans Bergsten, SPD O'Reilly (UNITs 5, 6, 7, 8)

**REFERENCE BOOKS:**

1. Programming world wide web-Sebesta, Pearson
2. Core SERVLETS AND JAVA SERVER PAGES VOLUME 1: CORE TECHNOLOGIES By Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia.
4. Jakarta Struts Cookbook, Bill Siggelkow, S P D O'Reilly for chap 8.
5. Murach's beginning JAVA JDK 5, Murach, SPD
6. An Introduction to web Design and Programming –Wang-Thomson
7. Web Applications Technologies Concepts-Knuckles,John Wiley
8. Java Script,D.Flanagan,O'Reilly,SPD.
9. Struts 2,Black Book, 2<sup>nd</sup> Edition,deamtech Press.
10. Web Warrior Guide to Web Programming-Bai/Ekedaw-Cengage Learning.
11. Beginning Web Programming-Jon Duckett, WROX.
12. Java Server Pages, Pekowsky, Pearson.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - II SEM

**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**

Security Goals, Security Attacks (Interruption, Interception, Modification and Fabrication), Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, access Control and Availability) and Mechanisms, A model for Internetwork security, Internet Standards and RFCs Conventional Encryption Principles & Algorithms(DES, AES, RC4), Block Cipher Modes of Operation, Location of Encryption Devices, Key Distribution.

**UNIT II**

Public key cryptography principles, public key cryptography algorithms(RSA, RABIN, ELGAMAL, Diffie-Hellman, ECC), Key Distribution. Approaches of Message Authentication, Secure Hash Functions(SHA-512, WHIRLPOOL) and HMAC Digital Signatures: Comparison, Process- Need for Keys, Signing the Digest, Services, Attacks on Digital Signatures, Kerberos, X.509 Directory Authentication Service

**UNIT III**

Email Security: Pretty Good Privacy (PGP) and S/MIME. IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management

**UNIT IV**

Web Security Requirements, Secure Socket Layer (SSL) and Transport Layer Security (TLS), Secure Electronic Transaction (SET). Basic concepts of SNMP, SNMPv1 Community facility and SNMPv3.

**UNIT V**

Intruders, Viruses and related threats, Virus Countermeasures. Firewall Design principles, Trusted Systems, Intrusion Detection Systems .

**TEXT BOOKS:**

1. Network Security Essentials (Applications and Standards) by William Stallings Pearson Education, 2008.
2. Cryptography & Network Security by Behrouz A. Forouzan, TMH 2007.
3. Cryptography and Network Security by William Stallings, Fourth Edition,Pearson Education 2007.

**REFERENCE BOOKS:**

1. Information Security by Mark Stamp, Wiley – INDIA, 2006.
2. Fundamentals of Computer Security , Springer.
3. Network Security: The complete reference, Robert Bragg, Mark Rhodes, TMH
4. Computer Security Basics by Rick Lehtinen, Deborah Russell & G.T.Gangemi Sr., SPD O'REILLY 2006.
5. Modern Cryptography by Wenbo Mao, Pearson Education 2007.
6. Principles of Information Security, Whitman, Thomson.
7. Information Systems Security,Godbole,Wiley Student Edition.
8. Cryptography and Information Security,V.K.Pachghare,PHI.
9. Network Security and Cryptography,B.Menezes,Cengage Learning.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - II SEM

**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 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 the organization and the problems encountered by him. These Elements will be useful to the student in career.

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

**Prerequisites**

- Organization working which the student will pursue in earlier classes

**UNIT I**

The meaning and role of MIS : What is MIS?, Decision support systems, systems approach, the systems view of business, MIS organization within the company, Managers view of Information systems,Contemporary Approaches to Information Systems,How Information Systems Impact Organizations and Business Firms.

**UNIT II**

Conceptual system design: Define the problems, set system objectives, establish system constraints, determine information needs, determine information sources, develop alternative conceptual designs and select one, document the system concept, prepare the conceptual design report. Organising data and information: Datawarehouses, Datamart and datamining

**UNIT III**

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

**UNIT IV**

Implementation, evaluation and maintenance of the MIS: Plan the implementation, acquire floor space and plan space layouts, organize for implementation, develop procedures for implementation, train the operating 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. Introduction to E-Commerce and E-commerce challenges, An overview of ERP, Applications of information systems to business. Security and ethical issues of information systems.

### **TEXT BOOKS:**

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

### **REFERENCE BOOKS:**

1. Management information Systems, 4th edition by Robert Schultheis, Mary Sumner, PHI-Seventeenth Reprint 2007.
- 2 Principles of Information systems, Sixth edition by Ralph M.Stair, George W.Reynolds, Cengage learning.
- 3 Management Information Systems, J.A.O'brien,G.M.Marakas,R.Behl,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 Univ.Press.

# **J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**

## **(Autonomous)**

**MCA**

**II Year - II SEM**

### **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, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines,advance perl - finer points of looping, pack and unpack, filesystem, eval, datastructures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

#### **UNIT II**

##### **PHP Basics**

PHP Basics- Features, Embedding PHP Code in your Web pages, Outputting the data to the browser, Datatypes, 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 – TranslatingWebsites- Updating Web sites Scripts, Creating the Localization Repository, Translating Files, text,Generate Binary Files, Set the desired language within your scripts, Localizing Dates, Numbers and Times.

#### **UNIT IV**

##### **TCL – Tk**

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 langauge, 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 India.
2. Python Web Programming, Steve Holden and David Beazley, New Riders Publications.
3. Beginning PHP and MySQL, 3<sup>rd</sup> Edition, Jason Gilmore, Apress Publications (Dream tech.).

**REFERENCE BOOKS:**

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

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - II SEM

**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 input devices

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.

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 IV**

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.

**UNIT V**

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", Zhiqiang Xiang, Roy Plastock, Schaum's outlines, Tata McGraw Hill edition.
2. "Procedural elements for Computer Graphics", David F. Rogers, Tata McGraw Hill, 2nd edition.
3. "Principles of Interactive Computer Graphics", Neuman and Sprout, 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.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

**II Year - II SEM**

**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. Corporate Digital Library - Document Library, digital Document types, corporate Data Warehouses.

**UNIT - IV**

Advertising and Marketing - Information based marketing, Advertising on Internet, on-line marketing process, market research. Consumer Search and Resource Discovery - Information search and Retrieval, Commerce Catalogues, Information Filtering.

**UNIT - V**

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

**TEXT BOOK :**

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

**REFERENCES :**

1. E-Commerce fundamentals and applications Hendry Chan, Raymond Lee, Tharam Dillon, Ellizabeth 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 – Cengage Learning..
5. E-Commerce – Business, Technology, Society, Kenneth C.Taudon,

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

II Year - II SEM

**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. Design Issues, User Requirement, Network Technologies and Protocols, IPC, Client-Server Communication, Group Communication, IPC in UNIX - Remote Procedure Calling, Design issues, Implementation, Asynchronous RPC

**UNIT II**

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

**UNIT III**

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

**UNIT IV**

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

**UNIT V**

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

**TEXT BOOKS:**

1. Distributed Systems Concepts and Design, G Coulouris, J Dollimore and T Kindberg, Third Edition, Pearson Education.

**REFERENCES:**

1. Distributed Operating Systems,Pradeep K.Sinha,PHI.
2. Advanced Concepts in Operating Systems, M Singhal, N G Shivarathri, Tata McGraw-Hill Edition.
3. Distributed Systems,S.Ghosh,Chapman&Hall/CRC,Taylor&Francis Group,2010.
4. Distributed Systems – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, PearsonEducation.
5. Distributed Algorithms,N.A.Lynch,Elsevier.

**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:

- Understanding the specific roles within a software organization as related to project and process management
- Understanding the basic infrastructure competences (e.g., process modeling and measurement)
- 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:

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

**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 dicriminants, 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, 1998
2. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition,Tata Mc-Graw Hill,2006

### 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 Manager, 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.

**J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY**  
**(Autonomous)**

MCA

**II Year - II SEM**

**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, Backpropogation Network. Assocciative Memory Networks. Traing 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 Propogation 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- Intrdution 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.

**References:**

1. Artificial Intelligence and SoftComputing- Behavioural and Cognitive Modelling of the Human Brain- Amit Konar, CRC press, Taylor and Francis Group.
2. Artificial Intelligence – Elaine Rich and Kevin Knight, TMH, 1991, rp2008.
3. Artificial Intelligence – Patric Henry Winston – Third Edition, Pearson Education. 4. A first course in Fuzzy Logic-Hung T Nguyen and Elbert A Walker, CRC. Press Taylor and Francis Group.

**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 analyse 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, 3<sup>rd</sup> Edition, Pearson Education.

#### **REFERENCE BOOKS:**

1. Introduction to Theory of computation. Sipser, 2nd Edition, Thomson.
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, Thomson, 1997.

### **WEB TECHNOLOGIES LAB**

#### **Objectives:**

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

#### **Hardware and Software required :**

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

#### **Week-1:**

Design the following static web pages required for an online book store web site.

##### **1) HOME PAGE:**

The static home page must contain three **frames**.

Top frame : Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

Left frame : At least four links for navigation, which will display the catalogue of respective links.

For e.g.: When you click the link “CSE” the catalogue for CSE Books should be displayed in the Right frame.

Right frame: The *pages to the links in the left frame must be loaded here*. Initially this page contains

description of the web site. Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE	Description of the Web Site			
ECE				
EEE				
CIVIL				

##### **2) LOGIN PAGE:**

This page looks like below: Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE	Login : Password:			
ECE				
EEE				
CIVIL				

### **3) CATALOGUE PAGE:**

The catalogue page should contain the details of all the books available in the web site in a table.

The details should contain the following:

5. Snap shot of Cover Page.
6. Author Name.
7. Publisher.
8. Price.
9. Add to cart button.

Note: Week 2 contains the remaining pages and their description.

### **Week-2:**

#### **4) CART PAGE:**

The cart page contains the details about the books which are added to the cart.

The cart page should look like this: Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
<b>Book name Price Quantity Amount</b>				
CSE	Java 2	\$35.5	2	\$70
ECE	XML bible	\$40.5	1	\$40.5
EEE				
CIVIL	<b>Total amount - \$130.5</b>			

#### **5) REGISTRATION PAGE:**

Create a “registration form “with the following fields

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

### **WEEK 3:**

#### **VALIDATION:**

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

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

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

### **Week-4:**

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

- 1) Use different font, styles:

In the style definition you define how each selector should work (font, color etc.). Then, in the body of your pages, you refer to these selectors to activate the styles.

For example:

```
<HTML>
<HEAD>
<style type="text/css"> B.headline {color:red; font-size:22px; font-family:arial; text-decoration:underline} </style> </HEAD>
<BODY> <b>This is normal bold</b>
<br> Selector {cursor:value} For example:
<html>
<head>
<style type="text/css"> .xlink {cursor:crosshair} .hlink{cursor:help}
</style>
</head>
<body> <b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br><a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html> <b class="headline">This is headline style bold</b>
</BODY>
</HTML>
```

3) Control the repetition of the image with the background-repeat property.

As background-repeat: repeat Tiles the image until the entire page is filled, just like an ordinary background image in plain HTML.

4) Define styles for links as

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

Example:

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

5) Work with layers:

For example:

```
LAYER 1 ON TOP: <div style="position:relative; font-size:50px; z-index:2;">LAYER 1</div>
<div style="position:relative; top:-50; left:5; color:red; font-size:80px; z-index:1">LAYER 2</div> LAYER 2 ON TOP: <div style="position:relative; font-size:50px; z-index:3;">LAYER 1</div> <div style="position:relative; top:-50; left:5; color:red; font-size:80px; z-index:4">LAYER 2</div>
```

6) Add a customized cursor:

Selector {cursor:value}

For example:

```
<HTML>
<HEAD>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>
<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
```

```
<a href="mypage.htm" class="hlink">HELP LINK</a> </b> </body> </html> <b  
class="headline">This is headline style bold  
</b>  
</BODY>  
</HTML>
```

### **Week-5:**

Write an XML file which will display the Book information which includes the following:

- 1) Title of the book
- 2) Author Name
- 3) ISBN number
- 4) Publisher name
- 5) Edition
- 6) Price

Write a Document Type Definition (DTD) to validate the above XML file.

Display the XML file as follows.

The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns.

Use XML schemas XSL and CSS for the above purpose.

Note: Give at least for 4 books. It should be valid syntactically.

Hint: You can use some xml editors like XML-spy

### **Week-6:**

#### **VISUAL BEANS:**

Create a simple visual bean with a area filled with a color.

The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false.

The color of the area should be changed dynamically for every mouse click. The color should also be changed if we change the color in the “property window”.

### **Week-7:**

- 1) Install TOMCAT web server and APACHE.

While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.

- 2) Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root.

## **Week-8:**

### **User Authentication :**

Assume four users user1,user2,user3 and user4 having the passwords pwd1,pwd2,pwd3 and pwd4 respectively. Write a servelet for doing the following.

1. Create a Cookie and add these four user id's and passwords to this Cookie.
2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords ) available in the cookies.

If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display " You are not an authenticated user ".

Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInitParameters() method.

## **Week-9:**

Install a database(Mysql or Oracle).

Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).

Practice 'JDBC' connectivity.

Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.

Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

## **Week-10:**

Write a JSP which does the following job:

Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database ( similar to week8 instead of cookies).

## **Week-11:**

Create tables in the database which contain the details of items (books in our case like Book name , Price, Quantity, Amount )) of each category. Modify your catalogue page (week 2)in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using JDBC.

## **Week-12:**

**HTTP** is a stateless protocol. Session is required to maintain the state.

The user may add some items to cart from the catalog page. He can check the cart page for the

selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time(i.e., from different systems in the LAN using the ip-address instead of localhost). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using

the method session.invalidate() ). Modify your catalogue and cart JSP pages to achieve the above mentioned functionality using sessions.

#### TEXT BOOKS:

1. Java Server Programming for Professionals, 2nd Edition, Bayross and others, O'reilly,SPD, 2007.
2. JDBC, Servlets, and JSP ,Black Book, K. Santosh Kumar, dreamtech.
3. Core Web Programming, 2nd Edition, Volume 1, M.Hall and L.Brown, PHPTR.
4. Core Web Programming, 2nd Edition, Volume 2, M.Hall and L.Brown, PHPTR.
5. Core Java, Volume 1, Horstman and Cornell, 8th Edition, Pearson Education, 2008.
6. Core Java, Volume 2, Horstman and Cornell, 8th Edition, Pearson Education, 2008.
7. Java Programming: Advanced Topics, 3rd Edition, J.Wiggles worth and P.McMillan,Thomson, 2007.
8. Struts 2 for Beginners,S.Shah & V.Shah,The X Team,SPD,2<sup>nd</sup> edition.



**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.

**List of Sample Problems/Experiments:****i) Data Warehousing and Data Mining:**

Task 1: Credit Risk Assessment

**Description:**

The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the 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

### Task 2: Hospital Management System

**Data Warehouse consists Dimension Table and Fact Table.**

#### REMEMBER The following

##### Dimension

The dimension object (Dimension):

- Name
- Attributes (Levels) , with one primary key
- Hierarchies

One time dimension is must.

##### About Levels and Hierarchies

**Dimension objects (dimension) consist of a set of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation. Hierarchies describe parent-child relationships among a set of levels.**

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

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

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

## About Unique Key Constraints

When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for each level of the hierarchy and a unique key constraint on the lowest level (Base Level)

Design a Hospital Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier, Time. Where measures are ‘ NO UNITS’, UNIT PRICE.

Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

PATIENT (patient\_name, Age, Address, etc.,)

MEDICINE ( Medicine\_Brand\_name, Drug\_name, Supplier, no\_units, Uinit\_Price, etc.,)

SUPPLIER :( Supplier\_name, Medicine\_Brand\_name, Address, etc., )

If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably.

Design the Hospital Management system data warehouse using all schemas. Give the example 4-D cube with assumption names.

## ii) Information Security

1. Working with Sniffers for monitoring network communication (Ethereal)
2. Understanding of cryptographic algorithms and implementation of the same in C or C++
3. Using openssl for web server - browser communication
4. Using GNU PGP
5. Performance evaluation of various cryptographic algorithms
6. Using IPTABLES on Linux and setting the filtering rules
7. Configuring S/MIME for e-mail communication
8. Understanding the buffer overflow and format string attacks
9. Using NMAP for ports monitoring
10. Implementation of proxy based security protocols in C or C++ with features like confidentiality, integrity and authentication

Following are some of the web links, which help to solve the above assignments

- [http://linuxcommand.org/man\\_pages/openssl1.html](http://linuxcommand.org/man_pages/openssl1.html)
- <http://www.openssl.org/docs/apps/openssl.html>
- <http://www.queen.clara.net/pgp/art3.html>
- <http://www.ccs.ornl.gov/~hongo/main/resources/contrib/gpg-howto/gpg-howto.html>
- <https://netfiles.uiuc.edu/ehowes/www/gpg/gpg-com-0.htm>
- <http://www.ethereal.com/docs/user-guide/>

### **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 client side 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:
  - Design, implement and evaluate a User Interface for a mobile application using J2ME.
  - Create a small but realistic working mobile application for small computing devices.
  - Categorize the challenges posed by developing mobile applications and be able to propose and evaluate and select appropriate solutions.

#### **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 BOOK

- 1 J2ME: The Complete Reference, James Keogh, Tata McGrawHill.

#### 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, Apress, 2005
4. Kicking Butt with MIDP and MSA:Creating Great Mobile Applications,1<sup>st</sup> 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 Win-runner).

### **TEXT BOOKS:**

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

### **REFERENCE BOOKS:**

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing,3<sup>rd</sup> edition,P.C.Jorgensen,Aurabach Publications(Dist by SPD).
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley, 2<sup>nd</sup> 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.

### **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,TATA McGrawHill.
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 them 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 Whois 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 LanManager 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 reventing Trojans, Trojan-Evading Techniques, System File Verification Subobjective to Trojan Countermeasures Viruses and Worms, Difference between a Virus and a Worm ,Types of Viruses,,Understand Antivirus Evasion Techniques,Understand Virus Detection Methods

#### **Sniffers**

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

#### **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 BOOKS:**

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 Mcgraw hill Edition
2. Ethical Hacking & Network Defense, Michael T. Simpson edition
3. Hacking Exposed Windows, Joel Scambray, cissp, Stuart McClure, Cissp, Third Edition, Tata Mcgraw hill edition
4. Hacking Exposed Window server 2003, Joel Scambray Stuart McClure, Tata Mcgraw hill edition

**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.**

**Introduction to Multimedia :** Internet and Multimedia communications,Multimedia Networks,Multimedia Applications,Multimedia Information representation- Digitization Principles,Text,Images,Audio and Video,Compression Methods-Basic Coding Methods – Run Length coding, Huffman coding, Arithmetic coding, Discrete Cosine Transform, Differential PCM, Motion Compensated Prediction, Video Compression – JPEG, H.261, MPEG-1 Video,MPEG 2 and 3 Video, H.263, Wavelet and Fractal Image Compression, Audio Compression.

**UNIT II.**

**Multimedia Applications in Networks ;** Introduction, Application Level Framing,Audio/Video Conferencing-Session Directories,Audio/Video Conferencing,Adaptive Applications,Reciever Hetrogenity,Real Time Application with Resource Reservation ,Video Server, Applications requiring reliable multicast – White Board , Network Text Editor for Shared Text Editing, Multi Talk, Multicast file transfer,MultiMedia Applications on the World Wide Web – Multicast Web Page Sharing, Audio/Video Streams in the www, Interactive Multiplayer Games.

**UNIT III**

**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 and Flex :** 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. Adobe Flex 2- Introduction, Flex Platform Overview, Creating a Simple User Interface, Accessing XML data from your application, Interacting with Server Side Applications, Costumizing your User Interface, Creating Charts and Graphs, Connection Independent RIAs on the desktop -Adobe Integrated Runtime(AIR), Flex 3 Beta.

## **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. Multimedia Communications: Protocols and Applications , Franklin F Kuo, J.Joaquin Garcia , Wolf gang Effelsberg,Prentice Hall Publications.
2. Multimedia Communications : Applications, Networks, Protocols and Standards, Fred Halsall,Addison Wesley Publications.
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. Professional Adobe Flex 2 , Rich Tretola , Simon barber and Renaun Erickson,Wrox,Wiley India Edition.
2. Multimedia Information Networking , Nalin K Sharda,PHI Learning.
3. Multimedia Computing, Communications & Applications , Ralf Steinmetz and Klara Nahrstedt,Pearson Education.
4. Multimedia Communication Systems: techniques, standards and networks, K.R.Rao,Bojkovic and Milovanovic.,PHI Learning.
5. Programming Flex 3,C.Kazoun and J.Lott,SPD.
6. Dojo,J.E.Harmon,Pearson Education.
7. Adobe Flex 3:Training from the Source,Tapper&others,Pearson Education.
8. Principles of Multimedia,R.Parekh,TMH.
9. Mastering Dojo,R.Gill,C.Riecke and A.Russell,SPD.

**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**

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

**UNIT II**

**Cataloging and Indexing:** Objectives, Indexing Process, Automatic Indexing, Information Extraction, **Data Structures:** Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure - **Automatic Indexing:** Classes of automatic indexing, Statistical indexing, Natural language, Concept indexing, Hypertext linkages

**UNIT III**

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

**UNIT IV**

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

**UNIT V**

**Multimedia Information Retrieval** – Models and Languages – Data Modeling, Query Languages, Indexing and Searching - **Libraries and Bibliographical Systems** – Online IR Systems, OPACs, Digital Libraries.

**TEXT BOOKS:**

1. Information Storage and Retrieval Systems: Theory and Implementation By Kowalski, Gerald, Mark T Maybury Kluwer Academic Press, 2000.
2. Modern Information Retrieval By Ricardo Baeza-Yates, Pearson Education, 2007.
3. Information Retrieval: Algorithms and Heuristics By David A Grossman and Ophir Frieder, 2<sup>nd</sup> Edition, Springer International Edition, 2004.

**REFERENCE BOOKS :**

1. Information Retrieval Data Structures and Algorithms By William B Frakes, Ricardo Baeza-Yates, Pearson Education, 1992.
2. Information Storage & Retrieval By Robert Korfhage – John Wiley & Sons.
3. Introduction to Information Retrieval By Christopher D. Manning and Prabhakar Raghavan, Cambridge University Press, 2008.



**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: Web Intelligence**

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

**UNIT II: Knowledge Representation for the Semantic Web**

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.

**UNIT III:Ontology Engineering**

Ontology Engineering,Constructing Ontology,Ontology Development Tools,Ontology Methods,Ontology Sharing and Merging,Ontology Libraries and Ontology Mapping,Logic,Rule and Inference Engines.

**UNIT IV: Semantic Web Applications, Services and Technology**

Semantic Web applications and services, Semantic Search,e-learning,Semantic Bioinformatics,Knowledge Base ,XML Based Web Services,Creating an OWL-S Ontology for Web Services,Semantic Search Technology,Web Search Agents and Semantic Methods,

**UNIT V:.Social Network Analysis and semantic web**

What is social Networks analysis,development of the social networks analysis, Electronic Sources for Network Analysis – Electronic Discussion networks, 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 interscience,2008.
2. Social Networks and the Semantic Web ,Peter Mika,Springer,2007.

**REFERENCE BOOKS:**

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

**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, Micro Soft DCOM, MOM, Challenges in Distributed Computing, role of J2EE and XML in distributed computing, emergence of Web Services and Service Oriented Architecture (SOA).

**UNIT II**

**Introduction to Web Services** – The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services - **Web Services Architecture** – Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication, basic steps of implementing web services, developing web services enabled applications.

**UNIT III**

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

**UNIT IV**

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

**UNIT V**

**Web Services Interoperability** – Means of ensuring Interoperability, Overview of .NET and J2EE. **Web Services Security** – XML security frame work, XML encryption, XML digital signature, XKMS structure, guidelines for signing XML documents.

**TEXT BOOKS:**

1. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India, rp – 2008.
2. Developing Enterprise Web Services, S. Chatterjee, J. Webber, Pearson Education, 2008.
3. XML, Web Services, and the Data Revolution, F.P.Coyle, Pearson Education.

**REFERENCE BOOKS:**

1. Building Web Services with Java, 2<sup>nd</sup> 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. Java Web Services Programming,R.Mogha,V.V.Preetham,Wiley India Pvt.Ltd.

**HUMAN COMPUTER INTERACTION  
(ELECTIVE-IV)**

**Objectives:**

You will gain an overview of Human-Computer Interaction (HCI), with an understanding of user interface design in general, and alternatives to traditional "keyboard and mouse" computing; become familiar with the vocabulary associated with sensory and cognitive systems as relevant to task performance by humans; be able to apply models from cognitive psychology to predicting user performance in various human-computer interaction tasks and recognize the limits of human performance as they apply to computer operation; appreciate the importance of a design and evaluation methodology that begins with and maintains a focus on the user; be familiar with a variety of both conventional and non-traditional user interface paradigms, the latter including virtual and augmented reality, mobile and wearable computing, and ubiquitous computing; and understand the social implications of technology and their ethical responsibilities as engineers in the design of technological systems. Finally, working in small groups on a product design from start to finish will provide you with invaluable team-work experience.

**UNIT I**

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

**UNIT II**

Design process – Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, understanding business junctions.  
Screen Designing:- Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

**UNIT III**

Windows – New and Navigation schemes selection of window, selection of devices based and screen based controls.  
Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

**UNIT IV**

Software tools – Specification methods, interface – Building Tools.

**UNIT V**

Interaction Devices – Keyboard and function keys – pointing devices – speech recognition digitization and generation – image and video displays – drivers.

## **TEXT BOOKS:**

1. The essential guide to user interface design, Wilbert O Galitz, Wiley DreamTech.
  2. Designing the user interface. 3rd Edition Ben Shneidermann , Pearson Education Asia

## **REFERENCE BOOKS:**

1. Human – Computer Interaction. Alan Dix, Janet Fincay, Gre Goryd, Abowd, Russell Bealg, Pearson Education
  2. Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech.
  3. User Interface Design, Soren Lauesen , Pearson Education.
  4. Human –Computer Interaction,D.R.Olsen,Cengage Learning.
  5. Human –Computer Interaction,Smith - Atakan,Cengage Learning.



**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, and 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, acade, 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

**REFERENCES :**

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-spd
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,Chhanda Ray,Pearson.



## **MOBILE APPLICATIONS DEVELOPMENT LAB**

### **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();
}
}

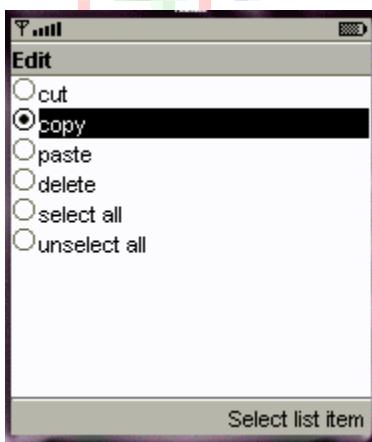
```

### **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.**

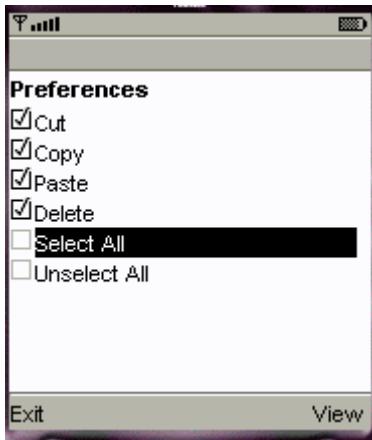
- \* cut
- \* copy
- \* past
- \* delete
- \* select all
- \* unselect all



#### **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

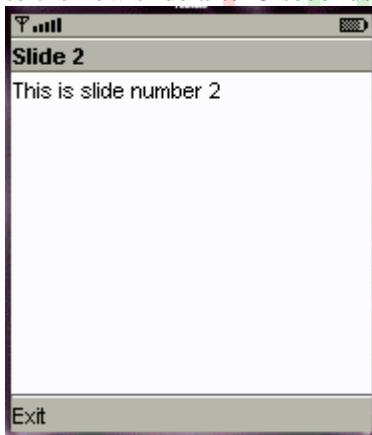
Create an MIDP application which examine, that a phone number, which a user has entered is in the given format.

- \* 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.

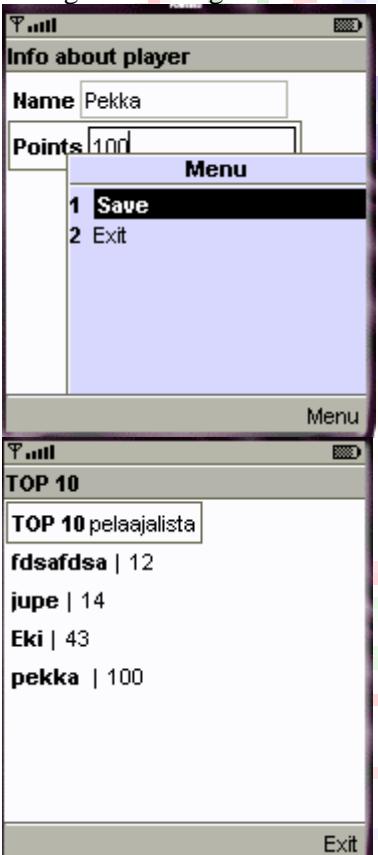


### 3.2 High-level UI

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.

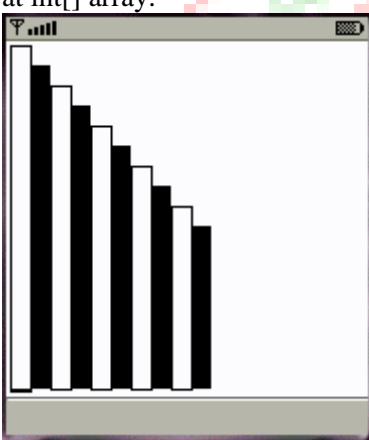


#### ***Week - 4 Working on Drawing and Images***

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

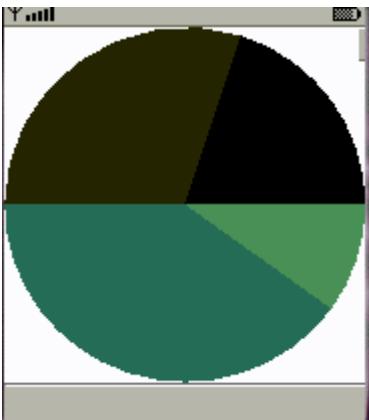


4.2 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array.



4.3 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array. You can enter four data (integer) values to the input text field.

A screenshot of a MIDP application window titled "Draw Pie". It features four input text fields labeled "1. value", "2. value", "3. value", and "4. value", each containing the integer values 20, 30, 40, and 10 respectively. Below these fields is a grey horizontal bar with two buttons: "Exit" on the left and "OK" on the right.



## **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 make a SOCKET Connection from a J2ME Phone. Many times there is a need to connect backend HTTP server from the J2ME application. shows how to make a SOCKET connection from the phone to port 80.

#### **6.2 Login to HTTP Server from a J2ME Program**

This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server.

Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server.

Note: Use Apache Tomcat Server as Web Server and Mysql as Database Server.

### **Week - 7 & 8 Web Application using J2ME**

The following should be carried out with respect to the given set of application domains: (Assume that the Server is connected to the well-maintained database of the given domain. Mobile Client is to be connected to the Server and fetch the required data value/information)

- Students Marks Enquiry
- Town/City Movie Enquiry
- Railway/Road/Air (For example PNR) Enquiry/Status
- Sports (say, Cricket) Update
- Town/City Weather Update
- Public Exams (say Intermediate or SSC)/ Entrance (Say EAMCET) Results Enquiry

Divide Student into Batches and suggest them to design database according to their domains and render information according the requests.



## **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:

1. Win Runner/QTP for functional testing.
2. Load Runner for Load/Stress testing.
3. Test Director for test management.
4. JUnit, HTML Unit, CPP Unit.

### **II) UML:**

#### **Objectives:**

1 Understand how UML supports the entire OOAD process.

2 Become familiar with all phases of OOAD

### **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.

### **Software Testing Lab List of Experiments**

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

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

