COURSE STRUCTURE

I – SEMESTER

Code No.	Name of the Course	Scheme of Instruction			Scheme of Examination		Total	Credits
		Periods per Week			Maximum Marks			
		Lectures	Tutorial	Lab.	Internal	External		
MCA101	Accounting and Financial Management	4	--	--	40	60	100	3
MCA102	Digital Logic and Computer System Organization	4	--	--	40	60	100	3
MCA103	Discrete Structures and Graph Theory	4	--	--	40	60	100	3
MCA104	English Language for Communication	4	--	--	40	60	100	3
MCA105	Introduction to Computer Systems and Programming	4	1	--	40	60	100	3
MCA106	Probability and Statistical Applications	4	--	---	40	60	100	3
MCA151	C Programming Lab	--	--	4	25	50	75	2
MCA152	Digital Logic and Computer System Organization Lab	--	--	4	25	50	75	2
	TOTAL	24	01	08	290	460	750	22

II – SEMESTER

Code No.	Name of the Course	Scheme of Instruction			Scheme of Examination		Total	Credits
		Periods per Week			Maximum Marks			
		Lectures	Tutorial	Lab.	Internal	External		
MCA201	Automata Theory and Formal Languages	4	--	--	40	60	100	3
MCA202	Data Structures	4	1	--	40	60	100	3
MCA203	Database Management Systems	4	1	--	40	60	100	3
MCA204	Operating Systems	4	--	--	40	60	100	3
MCA205	Operations Research	4	--	--	40	60	100	3
MCA206	Organizational Structure and Personnel Management	4	--	--	40	60	100	3
MCA251	Data Structures Lab	--	--	4	25	50	75	2
MCA252	Database Management Systems Lab	--	--	4	25	50	75	2
	TOTAL	24	02	8	290	460	750	22

LBRCE ★ EXAMINATION SECTION ★

I. Rajendra kumar

HEAD
Master of Computer Applications
Lakireddy Bali Reddy College of Engg.
MYLAVARAM - 521 230., Krishna Dt, A.P.

III – SEMESTER

Code No.	Name of the Course	Scheme of Instruction			Scheme of Examination		Total	Credits
		Periods per Week			Maximum Marks			
		Lectures	Tutorial	Lab.	Internal	External		
MCA301	Compiler Design	4	--	--	40	60	100	3
MCA302	Computer Networks	4	--	--	40	60	100	3
MCA303	Design and Analysis of Algorithms	4	--	--	40	60	100	3
MCA304	OOPs through JAVA	4	01	--	40	60	100	3
MCA305	Software Engineering	4	--	--	40	60	100	3
MCA306	UNIX Network Programming	4	--	--	40	60	100	3
MCA351	Communication Skills Lab	--	--	4	25	50	75	2
MCA352	OOPs through JAVA Lab	--	--	4	25	50	75	2
MCA353	UNIX Network Programming Lab	--	--	4	25	50	75	2
	TOTAL	24	01	12	315	510	825	24

IV – SEMESTER

Code No.	Name of the Course	Scheme of Instruction			Scheme of Examination		Total	Credits
		Periods per Week			Maximum Marks			
		Lectures	Tutorial	Lab.	Internal	External		
MCA401	Cloud computing	4	--	--	40	60	100	3
MCA402	Cryptography and Network Security	4	--	--	40	60	100	3
MCA403	Object Oriented Analysis and Design	4	--	--	40	60	100	3
MCA404	Software Testing	4	--	--	40	60	100	3
MCA405	Web Technologies	4	01	--	40	60	100	3
	ELECTIVE – I							
MCA4061	Computer Graphics and Vision	4	--	--	40	60	100	3
MCA4062	Distributed Operating Systems							
MCA4063	Principles of Programming Languages							
MCA4064	Software Design Methodologies							
MCA451	Mini Project			2	25	50	75	2
MCA452	Object Oriented Analysis and Design Lab	--	--	4	25	50	75	2
MCA453	Web Technologies Lab	--	--	4	25	50	75	2
	TOTAL	24	01	10	315	510	825	24

LBRCE ★ EXAMINATION SECTION ★

I. Rajendra kumar

HEAD
Master of Computer Applications
Lakireddy Bali Reddy College of Engg.
MYLAVARAM - 521 230., Krishna Dt, A.P.

V – SEMESTER

Code No.	Name of the Course	Scheme of Instruction			Scheme of Examination		Total	Credits
		Periods per Week			Maximum Marks			
		Lectures	Tutorial	Lab.	Internal	External		
MCA501	Artificial Intelligence	4	--	--	40	60	100	3
MCA502	Business Intelligence	4	--	--	40	60	100	3
MCA503	Mobile Application Development	4	--	--	40	60	100	3
	ELECTIVE - II							
MCA5041	Database Tuning	4	--	--	40	60	100	3
MCA5042	E-Commerce							
MCA5043	Object oriented software Engineering							
MCA5044	Open Source Software							
	ELECTIVE - III							
MCA5051	Big Data Analytics	4	--	--	40	60	100	3
MCA5052	Database Administration							
MCA5053	Software Project Management							
MCA5054	UNIX Administration							
MCA551	Business Intelligence Lab	--	--	4	25	50	75	2
MCA553	Mobile Application development Lab	--	--	4	25	50	75	2
MCA554	Testing tools Lab	--	--	4	25	50	75	2
MCA552	Internship	--	--	--	25	50	75	2
	TOTAL	20	--	12	300	500	800	23

VI – SEMESTER

Code No.	Name of the Course	Scheme of Instruction			Scheme of Examination		Total	Credits
		Periods per Week			Maximum Marks			
		Lectures	Tutorial	Lab.	Internal	External		
MCA651	Comprehensive Viva	--------			75	--	75	2
MCA652	Project Work	--------			50	150	200	20
	TOTAL	--------			125	150	275	22

★ LBRCE ★ EXAMINATION SECTION

I. Rajendra kumar

HEAD
Master of Computer Applications
Lakireddy Bali Reddy College of Engg.
MYLAVARAM - 521 230., Krishna Dt, A.P.

MCA101 – ACCOUNTING AND FINANCIAL MANAGEMENT

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The overall view on economic, financial accounting & financial management.
- The basic concepts of economics like micro & macro economics.
- The concepts of financial management like capital budgeting and its techniques.
- Ratio analysis.
- Financial management application.

Course Outcomes:

At the end of the course, the student should understand

- The fundamental concepts of Accounting
- The company's proposal/project.
- How to handle the complicated Financial situations and able to take proper decision to solve the issues.

SYLLABUS

UNIT - I

Fundamentals of accounting:

Meaning, Nature and Scope of Accounting. Double entry system of accounting. User of accounting information.

Accounting Process:

Basic books of accounts – Journals, Ledger & subsidiary books. Preparation of trial balance, Final Accounts, company final accounts.

UNIT - II

Basics of Financial Management:

Meaning and scope of Financial Management, Role of Financial Manager in Modern organization.

Objectives of financial Management :

Time value of money , overview capitalization and under Capitalization, Financial accounting Vs Financial management Vs Cost accounting.

UNIT - III

Overview of Cost Accounting and Marginal Costing:

Meaning, nature, Scope and importance of Cost Accounting, basic cost concepts, cost sheet. Absorption costing Vs Marginal Costing, Cost – volume – profit analysis –break Even point – significance and limitation of C-V-P Analysis, simple problems on marginal costing and C-V-P analysis.

UNIT - IV

Budgetary control and standard costing:

Meaning and features of budgetary control – preparation of various types of budgets fixed and flexible budgets (production, sales, cash and master budgets) zero based budgeting.

Standard costing:

Meaning and features of Standard costing – standard costing Vs budgetary control. Variance analysis – material, labor, over head and sales variances – simple problems.

UNIT - V

Accounting packages and computerized accounting:

Computerization of Accounts – Accounting packages – Tally & wings – Documents used for data collection, files management, master files transaction files – processing of different files and output obtained.

Text Book:

T. Paresh Shah, “Basic Financial Accounting for Management”, Oxford University Press, New Delhi, 2008 edition.

Reference Books:

1. T. Vijay Kumar,” Accounting for Management”, TMH,2010.
2. John Tannent, “Guide to Financial Management”, Viva.
3. A. Mukherjee and M. Haneef,” Financial Accounting”, TMH.
4. Ambaresh Gupta,” Basic Financial Accounting for Management”, Pearson,2005.
5. S.N.Maheswari, “Cost & Management Accounting”, sultan chand publications.

MCA102 – DIGITAL LOGIC AND COMPUTER SYSTEM ORGANIZATION

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The basic concepts and elements of computer systems.
- Fundamentals of computer organization and its relevance to classical and modern problems of Computer design
- The basic hardware for processing, storing, and moving information, and how they are organized within the internal architecture of a computer.
- Combinational and Sequential logical circuits.
- Various Data representations and how arithmetic and logical operations are performed by computers.
- The Performance of CPU, Memory and I/O operations.

Course Outcomes:

At the end of this course student will be able to:

- Design a basic computer system using major components.
- Write low-level programs to perform different basic instructions.
- Design and analyze the main functional units of a computer.
- Analyze, Design, and build Combinational and Sequential Digital Logic solutions to everyday problems.

SYLLABUS

UNIT – I
Introduction to Digital Computer:
Number system - Binary, Octal, HEXA and their inter-conversion, 1's and 2's complement,
Boolean Algebra and Logic Gates, De-Morgan's Theorem, Duality Theorem, K-Maps, Binary Addition, Binary Subtraction, Addition/Subtraction of Numbers in 1's Complement Notation, addition/Subtraction of Numbers in Two's Complement Notation, Fixed and floating point representation of numbers.

UNIT – II
Combinational and Sequential circuits:
Half Adder, Full Adder, Binary Adder and Subtracter, Decoder / Encoder, Multiplexer / De-multiplexer, Sequential Circuits: Flip Flops - SR, D, JK, Master – Slave and Edge Triggered, Shift Registers. Introduction to Counters: Synchronous and Asynchronous counters .

UNIT - III
Memory System:
Memory Hierarchy, Primary Memory – DRAM, SDRAM, DDR, RDRAM. ROM, PROM, EPROM, EEPROM, Concepts of Auxiliary, Associative, Cache and Virtual Memory

UNIT- IV

CPU Organization:

CPU Building Blocks, CPU Registers and BUS Characteristics, Addressing Modes, Instruction and Execution Interrupt cycle, Pipelining, Arithmetic Pipelining ,RISC Pipelining, Instruction Pipelining, Micro programmed control - control memory, address sequence, micro program example.

UNIT – V

Input- Output Organization:

IO interface, Modes of Transfer, Asynchronous Data Transfer, Interrupts, Direct Memory Access, Input Output Processor

Text BOOK:

MORRIS MANO, “Computer System Architecture”, Pearson Education, 3rd edition, 2002

Reference Books:

1. THOMAS C BARTEE, “Digital Computer Fundamentals”, TMH, 6th edition.
2. Linda Null and Julia Lobur,” The Essentials of Computer Organization and Architecture”, 2nd edition,2003.
3. William Stallings,” Computer Organization and Architecture”, 7th edition, Pearson.2006
4. Rajaraman, Radhakrishnan, “Digital Logic and Computer Organization”, PHI, 2009.
5. Pal Chaudhuri, “Computer Organization & Design”, PHI, 2rd edition ,1999.

MCA103 – DISCRETE STRUCTURES AND GRAPH THEORY

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Clear thinking and creative problem solving.
- The construction and understanding of mathematical proofs.
- Common mathematical arguments and proof strategies.
- A sense of familiarity and ease in working with mathematical notation and common concepts in discrete mathematics.
- The basic results in number theory, logic, combinatorics, and graph theory.
- The different properties of graphs.

Course Outcomes:

At the end of this course student will be able to

- Formulate and interpret statements presented in disjunctive normal form and determine their validity by applying the rules and methods of propositional calculus.
- Reformulate statements from common language to formal logic using the rules of propositional and predicate calculus, and assess the validity of arguments.
- Formulate short proofs using the following methods: direct proof, indirect proof, proof by contradiction, and case analysis.
- Apply the different properties of injections, surjections, bijections, compositions, and inverse functions, as well as the pigeon-hole principal.
- Construct elementary proofs using ordinary and strong induction in the context of studying the properties of recursion, relations, and graph theory, and identify fallacious inductive arguments.
- Apply basic counting principles including the pigeonhole principle and rules for counting permutations and combinations.
- Determine when a relation is reflexive, symmetric, antisymmetric, or transitive, apply the properties of equivalence relations and partial orderings, and explain the connection between equivalence relations and partitioning a set.
- Explain basic definitions and properties associated with simple planar graphs, including isomorphism, connectivity

Pre requisite:

- **F**undamentals in mathematics
- Theorem proving skills and
- Problem solving skills.

SYLLABUS

UNIT - I

Mathematical Logic:

Statements and Notations, Connectives, Well formed Formulas, Truth Tables, Equivalence Implications, Normal Forms.

Rules of Inference, Consistency of premises and indirect method of proof. Predicate Calculus: Predicates, statement functions, Variables and Quantifiers, Predicate formulas, Free and Bound variables, Universe of discourse, Inference theory of predicate calculus.

UNIT - II

Set theory and Relations:

Introduction, Relations and ordering, properties of binary relations, Equivalence, Compatibility relations, Partial Ordering, Hasse Diagram.

Functions:

Composition of functions, Inverse function, Recursive functions, Pigeonhole principle and its applications.

UNIT - III

Elementary Combinatorics:

Basics of Counting, Combinations and Permutations, Binomial Coefficients, Binomial and Multinomial theorems, Principle of inclusion and exclusion.

Generating Functions:

Generating function of sequences, calculating coefficient of generating functions

Recurrence Relations:

Solving recurrence relations by substitution, by characteristic roots, by generating functions. Solution of non homogeneous recurrence relations

UNIT - IV

Graph Theory-I:

Basic concepts, Representation of Graph, BFS, DFS, Isomorphism and sub graphs, Multi graphs, Planar graphs, Euler circuits, Hamiltonian graphs, Chromatic Numbers.

UNIT - V

Graph Theory-II:

Single source and All pairs shortest path problems- Dijkstra’s and Floyd’s algorithms, Transitive closure-Warshall’s Algorithm, Spanning Trees, Kruskal’s algorithm, Prim’s algorithm.

Text Book:

J.L.Molt, A.Kandel, and T.P.Baker, “Discrete Mathematics for Computer Scientists and Mathematicians, PHI,2001.

Reference Books:

1. J.P.Tremblery and R.Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, TMH,2004.
2. C L Liu and D P Mohopatra, “Elements of Discrete Mathematics”, TMH,2008.
3. Schaum’s Outlines, Lipschutz, and Lipson, “Discrete Mathematics”, TMH,2000.
4. Kolman, Busby and Ross, “Discrete Mathematical Structures”, 6th ed; PHI, 2007.

MCA104 – ENGLISH LANGUAGE COMMUNICATION SKILLS

Lecture : 4 Periods/week	**Internal Marks**	**: 40**
	External Marks	**: 60**
Credits : 3	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course, the students will learn

- English with emphasis on LSRW skills.
- The standard vocabulary along with the meaning and usage of the words.
- The concepts of effective writing with special focus on drafting reports, e-mails, letters, resume etc.
- The concepts of process, channels, and barriers of effective communication (verbal and non-verbal)
- The concepts of process and requisites of good Listening and Reading skills.
- The concepts of soft skills such as team work, professional etiquettes, leadership strategies, interpersonal and networking skills.

Course Outcomes :

After the completion of this course, prospective engineers will have the ability to

- Read, write and clearly understand whatever is written and spoken in English.
- Speak fluently with acceptable pronunciation and write using appropriate words, spellings, grammar and syntax
- Manage different situation and negotiate good English communication
- Think and analyze the situations and make good presentations of their work and decisions.
- Face interviews

SYLLABUS

UNIT – I

Communicative Grammar:

Tense forms, Subject - Verb Agreement, Question Tags, Spotting the errors, Active and Passive voice, Direct & Indirect speech.

Vocabulary:

One-Word Substitutes, Formation of Words, Using Words as Different Parts of Speech, Idioms and phrases, Words often confused.

UNIT – II

Communication

Process, Methods and Channels of Communication, Non Verbal Communication – Body Language, Barriers to Effective Communication.

UNIT - III

Listening

process & requisites of good listening, Reading skills- process & requisites

UNIT - IV

Features of Effective Writing

7 Cs- Letter writing, e-mail drafting, Resume Writing, Report writing, Synopsis and Statement of purpose.

UNIT - V

Soft Skills

Significance and Components, Interpersonal Skills, Professional Etiquettes, Empathy, Leadership development strategies - Team work- concept and exercises - Cross cultural communication - Networking skills.

REFERENCES:

1. John Seely, " Oxford Guide to Effective Writing and Speaking", Oxford University Press, New Delhi, 2005.

2. Murphy, " English Grammar", Cambridge University Press, New Delhi, 2009.

3. Rizvi Ashraf M, "Effective Technical Communication", Tata MCAGraw Hill, New Delhi,2008.

4.Ramesh & Ramesh, " Ace of Soft skills", Pearson Education, New Delhi, 2009.

5. Meenakshi Raman and Sharma, "Technical communication", Oxford University Press, 2007.

6. K. Alex, "Soft skills: know yourself & know the world", S.Chand, 2011.

MCA105 – INTRODUCTION TO COMPUTER SYSTEMS AND PROGRAMMING

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
Tutorial	**: 1 Period/Week**	**External Marks**	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Fundamentals of computer.
- The programming fundamentals.
- Developing the logical and programming skills in C language.
- Various hardware and software components of computer.
- Writing small to medium scale programs in C language

Course Outcomes:

At the end of this course student will be able to

- Assemble and dismantle a PC.
- Develop logical solution for a given problem.
- Programming for small to large scale applications through C language.
- Recognize correct C programming language syntax
- Trace the execution of code (i.e., playing computer)
- Understand the importance of algorithms and data structures in the problem-solving process

Pre requisite:

- Knowledge about various components in computer
- Algorithm writing
- Flowchart drawing skills and
- Problem solving skills.

SYLLABUS

UNIT - I

Introduction to Computers: History and Generations of Computers, Primary and Secondary memory, Input-output devices, ALU, Processor, Applications of computers.

C-Basics:

Algorithm / pseudo code, Flowcharts, Program development steps, Structure of C program, Identifiers, Basic data types , Constants, variables, types of operators, expressions, precedence and order of evaluation, type conversions, conditional expressions.

UNIT - II

Control structures:

If, If-Else and switch statements, Loops- while, do-while and for statements, break, continue and goto, Programming examples.

Arrays:

Introduction, Single-dimensional Arrays : Declaration of a Single-dimensional Array, Usage of Single-dimensional Array, Memory Representation of Single-dimensional Array , Operations on a Single-dimensional Array, Array of Arrays (Multi-dimensional Arrays) , Advantages and Limitations of Arrays

Strings and Character Arrays: Introduction, Strings, Character Arrays, Reading Strings from the Keyboard, Printing Strings on the Screen, String Library Functions, Array of strings.

UNIT - III

Functions:

Introduction, Functions, Classification of Functions - Based Upon who Develops the Function, Based Upon the Number of Arguments a Function Accepts. **Storage Classes**- auto, register, static, extern. **C Pre-Processor:** Translators, Phases of Translation, Preprocessor Directive Handling.

Pointers:

Operations on Pointers , void pointer , Null Pointer, Relationship Between Arrays and Pointers, Array of Pointers, Pointer to a Pointer , Pointer to an Array, Array of Character Pointers, Command Line Arguments, Dynamic Memory Allocation.

UNIT - IV

Structures, Unions, Enumerations and Bit-fields:

Introduction, Structure definition, Structure declaration , Operations on Structures ,Pointers to Structures, Array of Structures, Structures within a Structure (Nested Structures) Functions and Structures, type def and Structures, Unions, Practical Application of Unions, Enumerations ,Bit-Fields

UNIT – V

Files:

Introduction, Files, Streams, I/O Using Streams :Opening a Stream, Closing Streams Character Input , Character Output, File Position Indicator , End of File and Errors, Line Input , Line Output , Formatted Input, Formatted Output, Block Input , Block Output, Stream Buffering and Flushing the Streams, File Type, Files and Command Line Arguments

Text Book:

1. Ajay Mittal, “Programming in C: A Practical Approach” , First Edition , Person Education, 2010

Reference Books:

1. Reema Thareja “Programming in C”, First Edition, Oxford Publications,2011.
2. Pradip Dey, Manas Ghosh “Programming in C”, First Edition, Oxford Publications.
3. Brian W. Kernighan, Dennis M. Ritchie“ The C Programming Language”, 2nd Edition, PHI/Pearson Education, 1988
4. B.A. Forouzan and R.F. Gilberg Computer science, “A structured programming approach using C”, Third edition, Thomson

MCA106 – PROBABILITY AND STATISTICAL APPLICATIONS

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educaitonal Objectives:

The main objectives of this course are

- To revise elementary concepts and techniques encountered in probability.
- To extend and formalise knowledge of the theory of probability and random variables
- To introduce new techniques for carrying out probability calculations and identifying probability distributions.
- To motivate the use of statistical inference in practical data analysis
- To study elementary concepts and techniques in statistical methodology.
- To develop the skills for applying the probability and statistical techniques in computer science.

Course outcomes:

Finally this course is intended to contribute to the following program outcomes:

- An ability to apply knowledge of mathematics and statistics in computer science.
- An ability to design and conduct experiments, as well as to analyze and interpret data
- An ability to identify, formulate and solve the errors.
- An ability to use the techniques, skills and modern probabilistic and statistical tools necessary for computer science projects.

SYLLABUS :

UNIT – I :

PROBABILITY

Probability Theory: Sample space,Events & Probability, axioms of Probability . addtion law of probabilty conditional probability,Multiplication theorem and Baye’s theorem.

UNIT – II:

PROBAILITY DISTRIBUTIONS

Random variables and distributions : Discrete and continuous Random variables, Mathematical Expectation or Expectation.Discrete and continuous probability distributions-Binomical, Poisson and Normal Distributions and its applications.

UNIT - III :

SAMPLING DISTRIBUTION AND ESTIMATION

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:

TESTS OF HYPOTHESIS

Testing of hypothesis: Null and alternative hypothesis, one-tail and two-tail tests, tests of Hypothesis concerning means, Test of Hypothesis concerning proportions, F-test for variances , chi-squre test for goodness of fit and independence of attributes.

UNIT – V :

CORRELATION AND CURVE FITTING

Correlation and Regression: Linear correlation coefficient, rank correltion and Linear regression;
Non Linear regression: Least square fit , straight line, polynomial and Exponential curves.

TEXT BOOK

1. S.C.Gupta & V.K.Kapoor "Fundamentals of Mathematical statistics" Sultan chand and sons, New Delhi. 11th edition,2002.

REFERENCES

1. Jay L.Devore "Probability and Statistics for engineering and the sciences." Cengage Learning india, 8th edition,2012.
2. Murray R.Spiegel "Probability and Statistics" Schaum;s outline series, ,MCA Graw Hill, 4th edition.
3. T.K.V. Iyengar "Probability & Statistics for MCAA ". S. Chand & company, New Delhi, 2009 edition.
4. Miller and Freund's "Probability and Statistics for Engineers" Prentice Hall of India,NewDelhi, 8th edition,2011.

MCA151- C – PROGRAMMING LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Period/Week**	**External Marks**	**: 50**
Credits	**: 2**	**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course student will learn about

- The fundamentals of computer.
- The programming fundamentals.
- The logical and programming skills in C language.

Course Outcomes:

At the end of this course student will be able to

- Assemble and dismantle a PC.
- Develop logical solution for a given problem.
- Implement solutions for various problems using C language.

Pre requisite:

- Knowledge about various components in computer
- Algorithm writing
- Flowchart drawing skills and problem solving skills.

LIST OF EXPERIMENTS

Cycle-1:

1) Student is expected to observe various components of computer physically and assembling and dissembling of components.

2) Write programs in C Language:

a. To Exercise preliminary data types
b. To illustrate the usage of various Operators
c. To illustrate the order of evaluation of expressions

Cycle-2:

Write **programs in C Language**

a) To check whether the given year is leap year (or) not
b) To convert given two digit number into words using switch statement
c) To illustrate the usage of ‘goto’ statement
d) To find smallest& biggest number from the given set of 4 numbers using ‘if’ statement.
e) To calculate the student grade in the examination – assume suitable constraints.
f) To prepare electricity bill for the consumed units – assume suitable constraints.
g) To find roots of Quadratic Equation.

Cycle-3:

Write **programs in C Language**

a) To display first N natural numbers
b) To find whether the given number is Armstrong (or) not
c) To find reverse of the given number and to check whether it is palindrome (or) not.
d) To find whether given number is strong number (or) not.
e) To check whether a given number is Prime (or) not
f) To display prime numbers with in the given range(Nesting of Loops).

g) To display the following formats (Nesting of Loops)

```
 i)                  1                           ii) 5       4        3       2
            1             2                          4       3        2       1
            1              2    3                    3       2        1
      1              2            3        4         2       1
1           2                3        4         5            1
```

Cycle -4:

Write programs in C Language

a) To Find the sum and average of given numbers using Arrays.

b) To display elements of array in reverse order

c) To search whether the given element is in the array (or) not using linear search & binary search.

d) Addition, subtraction and multiplication of Matrices

e) Transpose of given matrix

f) To illustrate the use of any 5 string handling functions.

g) To accept line of text and find the number of characters, number of vowels and number of blank spaces in it.

h) To find whether the given string is palindrome (or) not.

Cycle -5:

Write programs in C Language

a) To find factorial of a given number using functions.

b) To swap two numbers using functions.

c) To find GCD of two numbers using recursion

d) To solve Towers of Honai problem.

e) To illustrate the use of external & static storage classes.

Cycle-6 :

Write programs in C Language

a) Illustrating pointer declaration, initialization and Pointer arithmetic.

b) To illustrate *call by reference*.

c) To find sum of the elements of the array using functions.

d) To illustrate the usage of command line arguments.

e) To illustrate the usage of dynamic memory management functions.

Cycle-7:

Write programs in C Language

a) To process the student records. Assume suitable fields for student structures (Different kinds of initialization of structure variables are to be exercised)

b) To read records of 10 employees and find their average salary (exercise array of structures & Nested structures concepts).

c) To handle a structure variable using pointers and implement self - referential structure

d) To exercise i) unions ii) bit fields iii) enum

Cycle -8 :

Write programs in C Language

a) Accessing content from files and writing content in to it. (Exercise different file operation modes)

b) Accessing structured data using files.

c) Copy the contents of one file into another (Exercise different file operation modes)

c) Exercise random access files operations

MCA152 – DIGITAL LOGIC AND COMPUTER SYSTEMS ORGANIZATION LAB

	Internal Marks : 25
Lab/Practical : 4 Periods/Week	**External Marks : 50**
Credits : 2	**External Examination : 3 Hrs**

Course Educational Objectives:

In this course student will learn about

- Fundamental designing concepts of different types of Logic Gates, Minimization techniques etc.
- Design, analysis and implementation of a digital circuit.
- Practical knowledge of DeMorgan's Law, Simplifying circuits using Karnaugh maps.
- Testing of all digital systems.

Course Outcomes:

At the end of this course student will be able to

- Analyze and design gate-level combinational logic circuits using Boolean algebra.
- Analyze, design and Utilize combinational components such as adders,multiplexes and decoders.
- Analyze and design simple synchronous sequential circuits.
- Simplify Boolean expressions in Product-of-Sums(POS) and Sum-of- Products forms.
- Use Multiplexers and Decoders in designing combinational circuits.
- Express Signed, Unsigned and fixed-point values in binary representation.
- Perform Signed and Unsigned Addition, subtraction, Division and Multiplication by observing errors.

LIST OF EXPERIMENTS

EXERCISE 1

Using Logic Gates:- AND, OR, NOT , NOR, XOR, NAND, XNOR, Buffer

EXERCISE 2

Boolean Algebra: Theorems and logical Gates, verification of truth tables for some given expressions

EXERCISE 3

Realization of Boolean expressions ; Using (i) AND – OR-NOT Gates (ii) NAND Gates (iii) NOR Gates

EXERCISE 4

Latches Flip – Flops : RS, JK,T,D, Master –Slave FF, Edge – Triggered Flip – Flops

EXERCISE 5

Registers: All types of Shift Register s and Adder, Subtractor, Divider, Negator, Comparator

EXERCISE 6

Counters: Binary Counter, Synchronous Binary Counter, Synchronous Up/Down counter

EXERCISE 7

Asynchronous Binary Counter, Ripple Counter, Decade Counter, Up/Down Counter

EXERCISE 8

Modulo Counter: Modulo - 5, Modulo – 10

EXERCISE 9

Adders / Sub tractors: Half Adder, Full Adder, 1 ‘s and 2’s complement addition

EXERCISE 10

Multiplexers/ Data Selector: 2- input and 8- input, Demultiplexers , Logic Function Generator

EXERCISE 11

Decoders and Encoders

EXERCISE 12

BCD adders and Comparators

EXERCISE 13

Code Converters : Decimal –to-Binary, Binary – to – Decimal, Decimal – to- Hexa Decimal, BCD- to –Decimal, Binary – to- gray, gray- to –Binary

EXERCISE 14

RAM, ROM, PROM, EPROM – Testing Memory Chips

Reference Books:

1. Rajaraman, Radhakrishnan, “Digital Logic and Computer Organization”, PHI, 2006.
2. Floyd & Jain, “Digital Fundamentals”, Pearson, 2000.

MCA201– AUTOMATA THEORY AND FORMAL LANGUAGES

Lecture : 4 Periods/week	**Internal Marks**	**: 40**
	External Marks	**: 60**
Credits : 3	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Various system software components
- Designing of assemblers
- Working of compilers, loaders and assemblers.
- Designing of a compiler.
- Application of finate automata.

Course Outcomes:

At the end of this course student will be able to

- Demonstrate various data structures used in compilers and assembles
- Designing of parsers
- Deign assemblers and compilers
- Develop system software

SYLLABUS

Unit-I

Fundamentals, Introduction to Finite Automata:
Finite automaton model, acceptance of strings, and languages, deterministic finite automaton and non deterministic finite automaton, transition diagrams, NFA with E-transitions-Significance, acceptance of Languages, Conversions and Equivalence: Equivalence between NFA with and without E-transitions, NFA to DFA conversion, minimization of FSM, equivalence between two FSM's Finite Automata with output-Moore and Melay machines

Unit-II

Regular Languages:
Regular sets, regular expressions, Identity rules, Constructing Finite Automata for a given regular expressions, Conversion of finite Automata to regular expressions. Pumping lemma of regular sets, closure properties of regular sets.

Unit-III

Grammar Formalism:
Regular grammars – right linear and left linear grammars, Context free grammar, derivation trees, Right most and leftmost derivation of strings, Ambiguity in context free grammars.
Minimization of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context free Languages. Enumeration pf properties of CFL

UNIT-IV

Push Down Automata:

push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA and Interco version

UNIT-V

Turing Machine:

Turing Machine, Definition, Model, design of Tm, Computable functions, Techniques of Turing Machine Construction

Undesirability :

Properties of recursive and Recursively Enumerable Languages; Universal Turing Machines (without any reference to undecidable problems), undesirability of posts. Correspondence problem,

The Chomsky Hierarchy : Regular grammars**,** unrestricted grammars, context sensitive languages.

TEXT BOOK

.“Introduction to Automata Theory Languages and Compilation “. Hopcroft H.E. and Ullman J.D.
Pearson Education

REFERENCES

1. Introduction to languages and the Theory of Computation, john C Martin , TMH

2. “Elements of theory of Computation”, Lewis H.P. & Papadimition C.H. Pearson / PHI

3. Introduction to computer Theory, Daniel I.A. Cohen, John Wiley.

4. Theory of Computer Science – Automata languages and computation – Mishra and Chandrashekaran, 2nd edition , PHI

5. Introduction to Theory of Computation – Sipser 2nd edition Thomson

MCA202 – DATA STRUCTURES

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The specification, representation, and implementation of Data Types and Data Structures.
- The Analysis of various Algorithms for Time and Space Complexities.
- Applications of Data Structures.
- Various Sorting and Searching Techniques.

Course Outcomes:

At the end of this course student will be able to

- Decide the appropriate data type and data structures for a given problem.
- Select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- Write the algorithms for various operations on Queues, Stacks, Linked Lists, Trees, Graphs, Sorting and Searching.
- Compare algorithms with respect to the time and space complexity.

SYLLABUS

UNIT - I
Introduction to Data structures:
Linear and nonlinear, Static and dynamic data structures, Introduction to complexity analysis & asymptotic notations.
Searching & Sorting techniques:
Linear and Binary search, Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort.

UNIT – II
Stacks and Queues:
Definition, representation, Applications, Infix to postfix conversion using stacks, Evaluation of Postfix expressions using stacks, Operations on Stacks and Queues, Circular Queues, DEQueues and Priority Queues.

UNIT – III
Linked Lists:
Single and Double Linked List – Implementation of various operations

UNIT - IV
Trees:
Terminology, Binary trees representation and Tree traversals, Expression trees, BST and operations on BST, Heap and Heap sort, Threaded binary trees.

UNIT - V

Graphs:

Terminology, Graph representation, Graph traversals-DFS, BFS, Dijkstra's, Warshall's and Floyd's algorithms, Minimum Cost spanning trees-Prim's and Kruskal's algorithms

Text Book:

Horowitz and Sahani , "Fundamentals of Data structures", Galgotia, 1st Edition,1999.

Reference Books:

1. Jean Paul Trembly & Paul G.Sorenson, "Introduction to Data structures with applications", Second Edition, TMH.,2009.
2. Langsam, Augenstein and Tenenbaum," Data structures using C and C++", PHI,1999.
3. W.Savitch, "Problem solving with C++, the OOP", 4th Edition, Pearson Education,2005.
4. Seymourand Lipschutz," Schaum's outline series – Theory and Problems of Data Structures", MGH, International Edition,1986.
5. John R. Hubbard, "Schaum's outlines Data structures with C++,2000.

MCA203 – DATABASE MANAGEMENT SYSTEMS

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Database and Database Management System.
- Creation of Database, and functions of Database Management System.
- Various Data Models, Schemas, Instances, Three Schema Architecture and DBMS Component
- Conceptual modeling of databases using ER diagrams.
- Relational Algebra and Relational Calculus.
- Enforcement of various integrity constraints.
- Database scheme refinement using normalization.
- Writing stored procedures on databases.
- Concurrency controlling and transaction management.
- Crash recovery management.
- Indexing Techniques for physical implementation of databases.

Course Outcomes:

At the end of this course student will be able to

- Develop conceptual scheme of database using conceptual model.
- Implement logical scheme of database.
- Create and manage database with all integrity constraints.
- Perform various DDL and DML operations.
- Refine the scheme of database by applying normal forms.
- Implement the transaction management protocols and crash recovery algorithms
- Create views, procedures, functions and triggers on databases.

Pre requisite:

- Requirements engineering,
- knowledge of various transaction processing schemes in organizations
- Knowledge of data storing using files and file system implementation.

SYLLABUS

UNIT – I
Introduction to Database Management Systems:

Database system Applications, Database system Vs File system, Data abstraction, Instances and Schemas, Database users, Database system structure, Database design and ER diagrams, ER Design - Entities, Attributes, Entity sets, Relationships and Relationship sets, Additional features of ER model.

UNIT - II
Introduction to Relational model:

Integrity constraints over the relations, Enforcing integrity constraints, Database Languages, DDL, DML, TCL, basic form of SQL query, Querying relational data, Logical database design, views, Destroying and altering tables/views. Nested queries correlated nested queries, Null values, Relation Algebra- selection, projection, renaming, join, examples.

UNIT – III

Normalization:

Schema refinement – Problems Caused by redundancy – Decompositions – Problem related to decomposition – reasoning about FDS – FIRST, SECOND, THIRD Normal forms – BCNF – Lossless-join Decomposition , Dependency- preserving Decomposition – Schema refinement in Data base Design – Multi valued Dependencies – Fourth Normal Form and Fifth Normal form.

UNIT - IV

Transaction Management and Concurrency Control :

ACID properties, Transactions and Schedules, Concurrent Execution of transactions, Serializability and Recoverability

Introduction to Lock Management:

Lock Conversions, Dealing with Dead Locks, Concurrency without Locking. Performance Locking, Transaction Support in SQL

Crash Recovery-Aries Recovery Algorithm.

UNIT - V

File organizations:

Comparison of File Organizations**,** Index data Structures, Tree based Indexing-Indexed Sequential Access Methods (ISAM), B+ Trees:Dynamic Index Structure.

Hash Based Indexing:

Static Hashing – Linear Hashing, Extendable hashing.

Text Book:

Raghurama Krishnan and Johannes Gehrke, “Data Base Management Systems”, TMH 3rd Edition,2003.

Reference Books:

1. Silberschatz, Korth, “Data Base System Concepts”, 5/e, TMH,2006.
2. Elmasri Navathe, “Data Base Management System”, 5/e, , Pearson,2005.
3. C.J.Date, “Introduction to Data Base Systems”, 8/e, , Pearson,2008.
4. Majumdr, Bhattacharyya,” Data Base Management Systems”, TMH ,96.
5. Peter ROB,Coronel,Cengage, “Data Base System Concepts”.

MCA204 – OPERATING SYSTEMS

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The structure of modern computers
- The purpose, structure, and functionalities of operating system
- The key aspects of Operating system

Course Outcomes:

At the end of this course student will be able to

- Understand the difference between processes and threads.
- Understand the issues and use of locks, semaphores and monitors for synchronizing multithreaded systems and implement them in multithreaded programs.
- Understand the issues of scheduling of user-level processes/threads.
- Understand the concepts of deadlock in operating systems and how they can be managed / avoided.
- Understand virtual memory management.
- Understand the differences between multiprocessor and multicomputer configurations.

SYLLABUS

UNIT-I

Introduction to Operating System:

Definition of Operating System, Functions of Operating System, Multi-user, Multiprocessing, Multiprogramming, Time Sharing, Real Time Systems, Virtual Computer, Operating System Structure, System Components, Services, System Calls, System Programs, System Design and Implementation.

UNIT - II

Process Management:

Process concept, Context Switching, Process Control Block, Process Scheduling, Operations on Processes, Co-operating Processes, Inter Process Communication,

CPU Scheduling:

Scheduling Concepts, Criteria, Scheduling Algorithms, Multiprocessor Scheduling, Real time Scheduling.

UNIT - III

Process Synchronization:

Critical Section, Synchronization Hardware, Semaphores, Problems of Synchronization, Critical Regions, Monitors.

Deadlocks:

Characterization, Handling Deadlocks, Deadlock Prevention, Avoidance, Detection, Deadlock Recovery

UNIT - IV
Memory Management:

Storage Hierarchy, Storage Management Strategies: Contiguous, Non Contiguous Storage Allocation, Single User-Fixed Partition, Variable Partition, Paging, Segmentation, Swapping-Virtual Memory concept, Demand paging and its performance, Need for Page Replacement, Page Replacement Algorithms, Thrashing.

UNIT - V
File System Interface and Implementation:

Access Methods, Directory Structure, Protection, File system structure, Allocation Methods, Free space Management, Directory Management, Directory Implementation, Efficiency and Performance, RAID Levels

Device management:

Physical characteristics Disk Scheduling: FCFS, SST, and C- SCAN.

Text Book:

Silberschatz, Galvin ,Gagne, “Operating System Principles”, 7th Edition, Wiley,2006.

Reference Books:

1. Tenenbaum A.S., Modern Operating Systems, 2nd edition, Pearson Education, 2001
2. William Stallings, “Operating Systems”, Sixth edition, PHI,2009.
3. Milan Milankovic, “Operating Systems, Concepts and Design”, MCAGraw-Hill.
4. Harvey M Deital, "Operating Systems", Addison Wesley ,1999.
5. Stuart .E. Madnick & John. J. Donovon , “Operating System”

MCA205 – OPERATIONS RESEARCH

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The reasoning and techniques used in formulating and solving deterministic problems in operations research.
- The connections between operations research and other math-related courses .
- The concepts of operations research by way of modeling real-world problems as Linear Programming (LP).
- Formulation of the mathematical, economical and statistical models for decision and control problems, to deal with the situations arising out of risk and uncertainty.
- Application of scientific techniques to analyze the firm's ongoing activities like production scheduling, assignment etc.
- Different techniques of analyzing the time involved in completing a project and the related costs are presented after defining the prerequisites of networks under project management.
- The concept of queuing system under various disciplines.
- The connections between LPs and Dynamic programming.
- Bellmen's principle of optimality to get an optimum solution of any multi stage decision problem.

Course Outcomes:

At the end of this course student will be able to

- Know the need of using Operations Research- A quantitative approach for effective decision making.
- Recognize, classify and use various models for solving a problem under consideration.
- Identify the situations in which linear programming technique can be applied.
- understand the fundamental concepts and general mathematical structure of a linear programming model.
- Handle the problem of degenerate and unbalanced transportation problems. Examine multiple optimal solutions and prohibited routes in the transportation problem.
- Appreciate the aims of study sequencing techniques and how to extend Johnson's rule to more complicated problems.
- Construct the network diagrams. Determine critical path and floats associated with non-critical activities and events along with total project completion time.
- Understand various components of a queuing system and analyze the verity of performance measures of queuing system.
- Understand various dynamic programming models and their applications in solving a decision problem.

SYLLABUS

UNIT - I

Development:

Characteristics and Phases scientific method, Types of models, General methods for solving OR problems, Operations Research models, Significance of operations research.

Linear Programming:

Introduction to Linear Programming, Two phase Simplex method, Big-M method ,Duality, Interpretation, Applications.

UNIT - II

Transportation Problem:

Introduction, Optimal solution, Un-balanced transportation problem, Degeneracy, Assignment problem: formulation optimal solution, variations. 1. a non-square (mxn) matrix, Restrictions.

Sequencing Model:

Classification of self-problems, processing of n jobs through two machines, three machines, processing of two jobs through m machines.

UNIT - III

Network optimization Models:

Shortest path problem, Minimum spanning tree problem, Maximum flow problem, Minimum cost flow problem, The project management with PERT/CPM, Scheduling a problem with PERT/CPM, Dealing with uncertain activity durations, Considering time cost trades Offs, Scheduling and Controlling, Projects costs, Evaluation of PERT/CPM.

UNIT - IV

Waiting Lines:

Introduction, Single channel, Poisson arrivals, Exponential service times, Unrestricted queue, with infinite population models, Single channel, Exponential Service times with infinite population and restricted queue, Multi-channel, Exponential service times with infinite population and unrestricted queue.

UNIT - V

Dynamic Programming:

Introduction, Billman's principal of optimality, Solution of problems with finite number of stages.

Text Book:

S.D.SHARMA: Operations Research, Kedarnath Ramnath, Meerut, 2001.

Reference Books:

1. P.K.GUPTA & D.S.HIRA: Operations Research, S. Chand,2002.
2. Taha, Operations Research, Macmillan, 1998.

MCA206 – ORGANIZATION STRUCTURE AND PERSONNEL MANAGEMENT

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Principles of management, Human Resource Management, communication and Strategic Management.
- The HR practices in the organization.
- The basic tips for better communication.
- The company's strength, weakness, opportunities and threats through SWOT analysis.
- The recruitment methods, selection process, training methods and induction about the job.

Course Outcomes:

At the end of this course student will be able to

- Tackle the business situations effectively.
- Gain the practical implication of theories and principles of management.
- Understand about the organization structure and hierarchy of the organization.
- Handle the organization problems with excellence.

Pre requisite:

- To run a successful business need a diverse range of business management skills. Study Skills for Business and Management Students has been carefully structured to be used throughout the course in order to hone the skills necessary to get a good degree and begin a successful career.

SYLLABUS

UNIT - I

Introduction to Management:

Concepts, Nature and definitions of Management-Management and administration, principles of management

Functions of management:

Planning, organizing, directing and controlling-importance of management-introduction to motivation

UNIT - II

Classical Theories of Organization & Behavior Theories of Organization:

Functional approach-division of labor, levels of authority, span of control, authority & responsibility, Efficiency of management. Concept of organization structure-formal and informal organization, difficulties due to informal organization-group behavior-Committee-motivation and theories of motivation.

UNIT - III

Human Resource Management:

Objectives, functions of HRM, duties and responsibilities of HR department in the organization-changing, concepts of personal management

UNIT - IV

HR Planning, Training and Development:

Preparation of man power inventory and forecasting, job description, recruitment, job specification and selection, Interviewing techniques, transfers, promotion and its policies. Objectives of training-identifying training needs-training methods-on the job training-off the job training.

UNIT - V

Communication, Strategic Management:

Importance of communication, communication process-methods of -two way communication, barriers of communication, Organizational barriers-essentials of effective Communication system.

Introduction-study of Strategic Management-environmental scanning-internal environment and external environment SWOT analysis-challenges in LPG.

Text Book:

Recenzo,Robins, "Personnel and Human Resource Management", PHI,India.

Reference Book:

1. Gay Dessler, "Human Resource Management", PHI, India,2005.
2. Agarwal," Organization and Management", TMH.
3. Venkat Ratnam, "Personnel management and Human Resources", TMH,2009.
4. L.M.Prasad, "Human Resource Management", S.Chand Publications.
5. Aswathappa, "Human Resource & Personnel Management", TMH,2006.

MCA251 – DATA STRUCTURES LAB

		Internal Marks	**: 25**
Lab/Practical : 4 Periods/Week		**External Marks**	**: 50**
Credits : 2		**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course student will learn about

- The specification, representation, and implementation of Data Types and Data Structures.
- The Analysis of various Algorithms for mainly Time and Space Complexity.
- Applications of Data Structures.
- How to get a base for advanced computer science study

Course Outcomes:

At the end of this course student will be able to

- Decide the appropriate data type and data structure for a given problem.
- Select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- Write the algorithms for various operations on Queues, Stacks, Linked Lists, Trees, Graphs, Sorting and Searching.
- Compare algorithms with respect to the time and space complexity.

LIST OF EXPERIMENTS

Implement the following programs using C language.

1. Implement Linear and Binary Search mechanisms.
2. Sort the given list of numbers using a) Selection Sort b) Bubble Sort c) Insertion Sort d) Merge sort e) Quick sort
3. Implement PUSH and POP operations on Stacks using Arrays. Handle the OVERFLOW and UNDERFLOW problems also.
4. Implement Insertion and Deletion operations on Queues using Arrays. Handle the OVERFLOW and UNDERFLOW problems also.
5. Perform various operations on Circular Queue using Arrays
6. Perform various operations on DEQueue using Arrays
7. To convert infix notation to postfix notation

8. Create a single linked list and implement the following operations:
a) Insert a node at specific position
b) Delete a node from a specific position
c) Counting the nodes
d) Reversing the linked list

9. Implement PUSH and POP operations on Stacks using Linked List. Handle the OVERFLOW and UNDERFLOW problems also.

10. Implement Insertion and Deletion operations on Queues Linked List. Handle the OVERFLOW and UNDERFLOW problems also.

11. Create a Double linked list and implement the following operations:
a) Insert a node at specific position
b) Delete a node from a specific position
c) Counting the nodes
d) Reversing the linked list

12. To implement Heap Sort

13. Write program to perform various operations on BST.

14. Write a program to find the Path Matrix of a graph using Wars hall’s algorithm.

15. Implement BFS and DFS traversal techniques on a given graph.

16. Write a program to find the All Pairs Shortest Path matrix using Floyd’s

MCA252 – DATABASE MANAGEMENT SYSTEMS LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Periods/Week**	**External Marks**	**: 50**
Credits	**: 2**	**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course student will learn about

- The discipline of database management as a subject in its own right, rather than as a compendium of techniques and product-specific tools.
- The Database environments towards an information-oriented data-processing oriented framework
- The relational model of data
- Present SQL and procedural interfaces to SQL comprehensively
- Introduction to systematic database design approaches covering conceptual design, logical design and an overview of physical design
- The concepts and techniques relating to query processing by SQL engines

Course Outcomes:

At the end of this course student will be able to

- Develop conceptual scheme of database using conceptual model.
- Implement logical scheme of database.
- Create and manage database with all integrity constraints.
- Refine the scheme of database by applying normal forms.
- Define data dictionary
- Perform various DDL, DCL and DML operations.
- Implement the transaction management protocols and crash recovery algorithms
- Create procedures, functions, cursors, triggers on databases using PL/SQL.

DATABASE MANAGEMENT SYSTEMS LAB (Using Oracle, SQL & PLSQL)

LIST OF EXPERIMENTS

1. Creating tables for various relations(in SQL)
2. Construct a bank database with ER diagrams and tables with all IC’s
3. Create sailors, reserves and boats tables and implement all algebraic operations.
4. Create a database for university with all IC’s
5. Aggregate functions
6. String functions
7. Nested queries
8. Views
9. Writing Triggers on bank database
10. Writing triggers on university database
11. Writing functions
12. Writing procedures

MCA301 – COMPILER DESIGN

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Finite Automation Model
- Differences between deterministic and non deterministic finite automation
- Conversion of Finite Automata into Regular Expressions.
- Context Free Grammars and Normal Forms.
- Designing of Turning Machine

Course Outcomes:

At the end of this course student will be able to

- Differentiate deterministic and non deterministic Finite automation.
- Convert Finite Automata to Regular Expressions.
- Construct parse trees for context free grammars.
- Demonstrate Push Down Automata
- Design Turning Machine

SYLLABUS

UNIT-I

Overview of Compilation:

Phases of Compilation – Lexical Analysis , pass and Phases of translation, interpretation, bootstrapping, data structures in compilation – LEX lexical analyzer generator

UNIT-II

Context Free grammars:

Context free grammars, derivation, parse trees, ambiguity grammars

Top down parsing:

Top down parsing – Backtracking, LL(1), recursive descent parsing, Predictive parsing, Preprocessing steps required for predictive parsing.

UNIT-III

Bottom up parsing :

Shift Reduce parsing , LR and LALR parsing, Error recovery in parsing, handling ambiguous grammar, YACC – automatic parser generator.

UNIT-IV

Semantic analysis :

Syntax directed translation, S-attributed and L-attributed grammars, type checker, intermediate code – abstract syntax tree, polish notation and three address codes, translation of simple statements and control flow statements

Runtime Storage:

Storage organization, storage allocation strategies scope access to now local names, parameters, language facilities for dynamics storage allocation.

UNIT-V

Code optimization:

Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding DAG representation.

Code generation:

Machine dependent code generation, object code forms, generic code generation algorithm, Register allocation and assignment. Using DAG representation of Block.

TEXT BOOK

1. Compilers Principles, techniques and Tools Aho, Ullman, Ravisethi, Pearson Education

REFERENCE BOOKS

1. Modern Compiler construction in C, Andrew W.Appel Cambridge University Press.
2. Compiler Construction, LOUDEN, Thomson

MCA302 – COMPUTER NETWORKS

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The fundamental principles of computer networking.
- Details and functionality of layered network architecture.
- Internetworking principles and how the Internet protocols, routing algorithms, and applications operate.
- Services of Transport layer and the protocols of Application layers.
- The necessary background in computer networks that allows students to practice in this field and that forms the foundation for advanced courses in networking

Course Outcomes:

At the end of this course student will be able to

- Network requirements and performance issues.
- Identify various layers in a network.
- Implement various routing and congestion controlling algorithms.
- Analyze Basic models of switched networks: datagram-based and virtual circuit-based Internet Protocols**.**

SYLLABUS

UNIT - I
Introduction to Computer Networks:
Uses of Computer Networks, Network Hardware: LAN, MAN, WAN, Bridges, Repeaters, Gateways, Network Software: Protocol hierarchies, Design issues, Types of services, Reference models: OSI, TCP/IP, ATM
Physical Layer:
Guided Transmission Media

UNIT - II
Data Link Layer:
Design issues of Data Link Layer, Error Correction and Detection, Elementary Data Link Protocols: Unrestricted Simplex Protocol, Stop and Wait, Simplex Protocol for noisy channel, Sliding Window Protocol, Go back N, Selective Repeat, HDLC
Medium Access Control sub layer (MAC):
Multiple Access Protocols, Ethernet-802.3, Wireless LAN, Bluetooth

UNIT - III
Network Layer:
Design Issues of Network Layer, Routing Algorithms: Optimality, Shortest path, Flooding, Distance Vector Routing, Hierarchical Routing, and Routing for Mobile Hosts.
Congestion Control Techniques:
Leaky Bucket, Token Bucket. Congestion Prevention Techniques: Traffic Shaping, Choke Packet, Load Shedding, Jitter Control.

UNIT - IV
Transport Layer:
Services of Transport Layer, Elements of Transport Protocols:UDP and TCP, Service Model, Protocol segment Header, Connection Establishment, Connection Release, TCP Connection Management.

UNIT - V
Application Layer:

DNS: Domain Name System, Electronic Mail (SMTP) :Architecture, User Agent , Message Format, Message Transfer, Delivery, FTP, The World Wide Web (HTTP) , Introduction to Storage Area Networks(SAN), Peer-to-Peer Networks

Text Book:

Andrew S. Tanenbaum “Computer Networks” Fourth Edition, Pearson Education-2006.

Reference Books:

1. Behrouz A.Frouzon “Data Communications and Networks” Tata MCAGraw Hill Publication, 200
2. William Stallings “Data and Computer Communication “Sixth Edition , Pearson Education Asia, 2008.
3. Larry L.Peterson and Bruce S.Davie “Computer Networks” A system approach Third Edition, Kaufmann Publisher, 2011.
4. Michel A. Gallo, William M.Hancock “Computer Communication and Networking Technologies”, Thomson Publication,2001.
5. William Ashay, “Understanding Data Communications and Networks” , 3nd Edition, Vikas Publishing House,2008.

MCA303 – DESIGN AND ANALYSIS OF ALGORITHMS

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The importance of studying the complexity of a given algorithm.
- various algorithmic design techniques.
- Data structures and/or algorithmic design techniques in solving new problems including recursion, divide-and-conquer, greedy algorithms, and dynamic programming.
- Fundamental computing algorithms: sorting, searching, and graph algorithms.
- The basic computability concepts and the complexity classes P, NP, and NP-Complete.
- Some techniques for solving hard problems.

Course Outcomes:

At the end of this course student will be able to

- Apply and analyze the complexity of certain divide and conquer, greedy, and dynamic programming algorithms.
- Differentiate the lower and upper bounds of various problems and their importance in deciding the optimality of an algorithm.
- Differentiate between various algorithms for sorting (e.g., insertion, merge, quick-sort, and heap sort), searching (e.g., linear and binary search), and selection (e.g., min, max) and when to use them.
- Implement the techniques used for designing fundamental graph theory algorithms (e.g., breath-first and depth-first algorithms) and apply them to solve other related problems (e.g., single source shortest path as in Dijkstra's and Bellman-Ford algorithm, multiple source shortest path as in Floyd's Algorithm, minimum spanning trees as in Prim's and Kruskal's algorithms)
- Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.
- Explain and apply backtracking and branch and bound techniques to deal with some hard problems.

SYLLABUS

UNIT - I

Introduction:

Algorithm, Pseudo code for expressing algorithms, Performance Analysis: Space complexity, Time complexity, Asymptotic Notation: Big Oh notation, Omega notation, Theta notation,

Divide and Conquer:

General method, Applications, Binary search, Quick sort, Merge sort, Stassen's matrix multiplication

UNIT - II

Greedy Method:

General method, Applications: Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem, Optimal storage on tapes.

Basic Search and traversal Techniques:

AND/OR graphs, Bi-connected components, Depth-first search, Breadth - first Search.

UNIT - III

Dynamic Programming:

General method, Applications: Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales man problem, Reliability Design.

UNIT - IV

Backtracking:

General method, Applications: n-queens problem, sum of subsets problem, graph colouring, Hamiltonian cycles.

UNIT - V

Branch and Bound:

General method, Applications - Travelling sales person problem,0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution.NP-Hard and NP-Complete problems: Basic concepts, Non deterministic algorithms, NP - Hard and NP Complete classes, Cook's theorem.

Text Book:

Ellis Horowitz, Satraj Sahni and Rajasekharam**,** "Fundamentals of Computer Algorithms", Galgotia publications pvt. Ltd.

Reference Books:

1. M.T.Goodrich and R.Tomassia, "Algorithm Design: Foundations, Analysis and Internet examples", John wiley and sons.
2. *C*.E.Leiserson & T.H.Cormen, "Introduction to Algorithms", second edition.
3. Aho, Ullman and Hopcroft Design and Analysis of algorithms, ,Pearson education.

MCA304 – OOPS THROUGH JAVA

Lecture : 4 Periods/week	**Internal Marks**	**: 40**
	External Marks	**: 60**
Credits : 3	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- The OOP s concepts.
- Object oriented Programming
- Window based applications using AWT and swing components
- Multi-threading concepts

Course Outcomes:

At the end of this course student will be able to

- Mini projects after completion of Applets, AWT and SWING Components
- Students are get benefited in subject in practical and industry orientation manner
- Students are confident in java programming
- Students feel confident all areas of java
- get confident on all aspects of the subject

SYLLABUS

UNIT - I

Features of OOPS:

OOPS concepts, Introduction to Java, History of Java, Features of Java,Java Virtual Machine, Garbage collection , Why Java is important for Internet?, Programming concepts of basic Java, Identifiers and Keywords, Data types in Java, Java coding conventions, Expressions in Java, Control structures, Decision making statements ,Arrays.

UNIT - II

Objects and Classes:

Object fundamentals, Pass by value, Pass by reference, Overloading, Overriding, Constructors, Finalization, Subclasses(Inheritance), this, super, final with inheritance, Dynamic method dispatch, Scope rules, Static data, Static methods, Static blocks, class modifiers , String Handling, Command line arguments, Abstract Classes, Interfaces, Inner classes, Packages, Package access, Importing packages and classes, User define packages, Class-path.

UNIT – III

Exception Handling:

Types of Exceptions, try, catch, finally, throw keywords, Handling User defined Exceptions.

Multithreading:

processes and threads, Thread states, Thread life cycle, Creating threads, Interrupting threads, Thread priorities, Synchronizing threads, Inter thread communication, Thread groups, Daemon threads.

JAVA I/O:

Files and Streams, Stream classes, Reader-Writer classes, Utilities, Serialization and Deserialization.

UNIT – IV

Applets:

Types of Applets, Applet life cycle, Graphics, Parameter Passing

AWT:

Abstract Window Toolkit, Components and Graphics, Containers, Frames and Panels, Layout Managers, Border Layout, Flow Layout, Grid Layout, Card Layout, Event delegation model, Event sources and Event handlers, Event categories, Event Listeners, Adapters classes, Anonymous classes.

UNIT – V

Swings:

Introduction, Handling Swing Controls like Icons, Buttons, Textboxes, Combo Boxes, Tabbed Panes, Scroll Panes, JTree, JTable, Differences between AWT Controls & Swing Controls, developing home page using Applets & Swings.

Java.util:

Java.util package: Vector, Array List, Hash map, Hash table, StringTokenizer, and Date class..

Text Book:

Herbert Schildt,” The Complete Reference Java J2SE”, 7th Edition, TMH, New Delhi,2009.

Reference Books:

1. Cay Horstmann, John Wiley and Sons, “Big Java 2nd Edition”, John Wiley and Sons, Pearson Edu,2006.
2. H.M.Dietel and P.J.Dietel,” Java How to Program” , Sixth Edition, Pearson Education/PHI,2007.
3. Cay.S.Horstmann and Gary Cornell, “Core Java 2, Vol 1, Fundamentals, Seventh Edition, Pearson Education,2011.
4. R.Krishna Murthy-“Java and Internet Programming”
5. Somasundaram, Jaico, “Java.
6. Y.Daniel Liang, “Introduction to Java programming”, Pearson,2012.

MCA305 – SOFTWARE ENGINEERING

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- How to analyze and Model User's Requirements.
- Selecting an appropriate Process Model and apply it to All Stages of Software Development Life Cycle.
- Selecting and Appling Appropriate Design Methodology.
- Assuring Software Quality, Select and Apply Appropriate Testing Strategies.
- Selecting and Applying Appropriate Metrics to Estimate Software Size, Effort, and Cost.
- Preparing Project Schedule, and Monitor the Project Progress.
- The Characteristics and Applicability of Various Software Tools.

Course Outcomes:

At the end of this course student will be able to

- Prepare SRS, design, coding and test plan for small projects.
- Identify the Functional and Non-Functional requirements.
- Identify the suitable Software process model for a given project.
- Develop Design models for Object-Oriented Testing
- Collect and analyze various Metrics for software
- Perform manual testing for simple projects.
- Conduct various Verification and Validation activities during Software Development.
- Identify and analyze various Risks during Software development

SYLLABUS

UNIT – I

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

A Generic view of process:
Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI)

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

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

UNIT – II

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

Design Engineering:
Design process and Design quality, Design concepts, the design model

Creating an architectural design:
Software architecture, Data design, Architectural styles and patterns, Architectural Design

UNIT - III

Object-Oriented Design:
Objects and object classes, An Object-Oriented design process, Design evolution.

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, 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 Book:

Roger S. Pressman, "Software Engineering, A practitioner's Approach", 6th edition. MCAGraw-Hill International Edition,2005

Reference Books:

1. Somerville**, "**Software Engineering", 7th edition, Pearson education,2009.
2. K.K. Agarwal & Yogesh Singh," Software Engineering", New Age International Publishers,2008.
3. James F. Peters, Witold Pedrycz," Software Engineering an Engineering approach", John Wiley,2007.
4. Waman S Jawadekar, "Software Engineering principles and practice-, MGH.

MCA306 – UNIX NETWORK PROGRAMMING

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about

- Various utilities and filters in Unix operating system.
- Programming in networking environment.
- Various operating system functionalities including process management, file management, networking etc.
- Shell programming
- Inter Process Communication.
- Network Programming through sockets.

Course Outcomes:

At the end of this course student will be able to

- Utilize various utilities in Unix operating system.
- Implement various OS functionalities using system calls.
- Implement Inter Process Communication.
- Implement network programming using sockets.
- Implement Remote Procedure Call

SYLLABUS

UNIT – I

Introduction:

UNIX file system, Features of Unix OS, vi editor

Utilities and commands:

File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, text processing utilities and backup utilities, detailed commands to be covered are cat, tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, comm, cmp, diff, tr, tar, cpio, cp, mv, ln, rm,unlink, mkdir, rmdir, du, df, mount, umount, find, umask, ulimit, ps, who, w, finger, arp, ftp, telnet, rlogin, Introduction to shell programming, shell responsibilities, building own command library.

UNIT - II

Working with the Bourne shell:

Pipes and input redirection, Output redirection, here documents, Shell as a programming language, Shell meta characters, Shell variables, Shell commands, Environment, Control structures, Shell script examples.

UNIX Files:

Unix file structure, Directories, Files and devices, System calls, Library functions, Low level file access, Usage of open, creat, read, write, close, lseek, stat, fstat, umask, dup, dup2, Standard I/O (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputc, putc, putchar,fgets, gets

UNIT-III

I/O:

Formatted I/O, Streams and file descriptors, File and directory maintenance (chmod, chown, unlink, link, symlink, mkdir, rmdir, chdir, getcwd), Directory handling system calls (opendir, readdir, closedir,rewinddir, seekdir, telldir)

UNIX Process:

What is Process, Process structure, Starting new process, Waiting for a process, Zombie process, Process control, Process identifiers, interface for process management-fork, vfork, exit, wait, waitpid, system

UNIT - IV

IPC:

Pipes, FIFOS. Semaphores-Unixsystem-V Semaphores, Unix kernel support for Semaphores, Unix APIs for **Semaphores. Message Queues**:

UNIX system-V Messages Queues, UNIX kernel support for Message Queues, Unix APIs for Message Queues, Client/Server example.

Shared Memory:

UNIX system V Shared memory, Unix kernel support for shared memory, Unix APIs for shared memory, Semaphore and Shared memory example.

UNIT – V

Sockets:

Berkeley Sockets, Socket system calls for connection oriented protocol and connectionless protocol, Client/Server example.

Sun RPC:

Client/Server example

Text Book:

W.R.Stevens , “Unix Network Programming”, Pearson/PHI,2010.

Reference Books:

1. Sumitabha Das, “Unix Concepts and Applications”, 3rd Edition, TMH,2010.
2. Graham Glass, King Ables,” Unix for programmers and users”, 3rd Edition, Pearson Education,2009.
3. T.Chan, “Unix system programming using C++”, PHI
4. Stephen A.Rago, “Unix System-V Network Programming”, Pearson Education.
5. Kernighan and Pike, ”Unix programming environment” , PHI. / Pearson Education,1984.

MCA351-COMMUNICATION SKILLS LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Periods/Week**	**External Marks**	**: 50**
Credits : 2		**External Examination**	**: 3Hrs**

Course Educational Objectives

In this course, the students will learn to

- Gather ideas and information, to organize ideas relevantly and coherently.
- Engage in debates.
- Participate in group discussions, Face interviews.
- Write project/research reports/technical reports/ formal letters.
- Make oral presentations.
- Transfer information from non-verbal to verbal texts and vice versa.
- To take part in social and professional communication.

Course Outcomes:

After the completion of this course, prospective engineers will have the ability to

- Make power point presentations and oral presentations.
- Articulate English with good pronunciation.
- Face interviews and skillfully manage themselves in group discussions

SYLLABUS:

The following course content is prescribed for the Communication and presentations Lab:

- Oral Presentations – JAM
- Functional English - starting a conversation – responding appropriately and relevantly – using the right body language – role play in different situations.
- Group Discussion – dynamics of group discussion, intervention, summarizing, modulation of voice, body language, relevance, fluency and coherence.
- Making power point presentations.
- Interview Skills – concept and process, pre-interview planning, opening strategies, answering strategies, practicing mock-interviews.
- Resume' writing – structure and presentation, planning, defining the career objective, projecting ones strengths and skill-sets, summary, formats and styles, letter-writing.
- Reading comprehension – reading for facts, guessing meanings from context, scanning, skimming, inferring meaning, and critical reading.

Minimum Requirement:

The English Language Lab shall have two parts:

i. **The Computer aided Language Lab** for 60 students with 60 systems, one master console, LAN facility and English language software for self- study by learners.

ii. **The Communication Skills Lab** with movable chairs and audio-visual aids with a P.A System, a T. V., a digital stereo –audio & video system and caMCAorder etc.

System Requirement (Hardware component):

Computer network with LAN with minimum 60 multimedia systems with the following specifications:

i. P – IV Processor
   1. Speed – 2.8 GHZ
   2. RAM – 512 MB Minimum
   3. Hard Disk – 80 GB

ii. Headphones of High quality

Suggested Software:

- Globarena’s software
- Young India’s Clarity software

Books Recommended:

1. Stephen Bailey , “Academic Writing- A Practical guide for students”, Rontledge Falmer, London & New York, 2004.
2. Dr A Ramakrishna Rao, Dr G Natanam & Prof SA Sankaranarayanan, “English Language Communication : A Reader cum Lab Manual, Anuradha Publications, Chennai
3. DELTA’s key to the Next Generation TOEFL Test: Advanced Skill Practice, New Age International (P) Ltd., Publishers, New Delhi.
4. Books on TOEFL/GRE/GMAT/CAT by Barron’s/cup
5. IELTS series with CDs by Cambridge University Press.

MCA 352 – OOPS THROUGH JAVA LAB.

		Internal Marks	**: 25**
Lab/Practical	**: 4 Periods/Week**	**External Marks**	**: 50**
Credits : 2		**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course student will learn about

- The OOP s concepts.
- Object oriented Programming
- Window based applications using AWT and swing components
- Multi-threading concepts

Course Outcomes:

At the end of this course student will be able to

- Develop Object Oriented Projects
- Develop Applications using AWT and SWING Components
- Develop Applications using SERVELTS and JSP.

SYLLABUS

1) Write a Java program to find the roots of a quadratic equation?

2) Write a Java program to generate first n Fibonacci numbers?

3) a) Write a Java program to reverse the given number?

b) Write a Java program to check whether given number is Prime or not?

4) Write a Java program to check whether given number is Palindrome or not?

5) Write a Java program to check whether given number is Armstrong or not?

6) Write a Java program to find factorial of the given number using recursions?

7) Write a Java program to find min and max number of given array?

8) Write a Java program to search an element by using linear search and binary search?

9) Write a Java program to sort the elements of an Array?

10) Write a Java program to perform Matrix Multiplication?

11) Write a Java program using constructors

12) Write a Java program for constructor Overloading.

13) Write a Java program using inheritance?

14) Write a Java program to implement Method over Loading and Method over riding?

15) Write a Java program by using this and super key word.

16) Write a Java program by using final variables and final methods.

17) Write a Java program to implement dynamic method dispatch.

18) Write a Java program using abstract class?

19) Write a Java program to implement Multiple Inheritance (Interface)?

20) Write a Java program on demonstration of packages?

21) Write a Java program illustrating string handling functions

22) a) Write a Java program to check whether given string is palindrome (or) not. ?

b) Write a Java program to sort the set of strings in sorting order?

c) Write a Java program to find sum of the numbers using String Tokenizer?

23) Write a Java program by using length () and capacity () Methods of String Buffer?

24) Write a Java program to find the sum of the numbers by using command line arguments?

25) (a) Write a Java program by using Exception handling Mechanism including Finally block?

(b) Write a Java program to Handle User Defined Exceptions?

26) (a) Write a Java program to create Multithreads?

(b) Write a Java program on Thread Synchronization

27) Write a Java program to implement Inter thread communication?

28) (a) Write a sample Applet program to Display Message?

(b) Write an Applet program using Graphics?

(c). Write an Applet program to pass parameters to Applet.

29) Write a Java program to create user login by using AWT components?

30) Write a Java program to implement border layout.

31) (a) Write an applet program to handle Mouse Events?

(b) Write an applet program to handle Key Events using adapter Class?

32) Write a Java program by using StringTokenizer class.

MCA 353 – UNIX NETWORK PROGRAMMING LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Periods/Week**	**External Marks**	**: 50**
Credits	**: 2**	**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course student will learn about

- Various utilities and filters in Unix operating system.
- Programming in networking environment.
- Various operating system functionalities including process management, file management, networking etc.
- Shell programming
- Inter Process Communication.
- Network Programming through sockets.

Course Outcomes:

At the end of this course student will be able to

- Utilize various utilities in Unix operating system.
- Implement various OS functionalities using system calls.
- Implement Inter Process Communication.
- Implement network programming using sockets.
- Implement Remote Procedure Call.

SYLLABUS

1. Write a Shell script to generate a multiplication table.

2. Write a Shell script that copies multiple files to a directory.

3. Write a Shell script that counts the number of lines and words present in a given file.

4. Write a Shell script that displays the list of all files in the given directory.

5. Write a Shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two division options: one returns the quotient and the other returns reminder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add (-a), subtract (-s), multiply (-m), quotient (-c) and reminder (-r).

6. Write a Shell script to reverse the rows and columns of a matrix.

7. Write a C program that counts the number of blanks in a text file.
   (a) Using standard I/O
   (b) Using system calls.

8. Implement in C the following Unix commands using system calls.
   a) cat b) ls c) mv

9. 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.

10. Write a C program that illustrates uses of the mkdir, opendir, readdir, closedir, and rmdir APIs.

11. Write a C program that illustrates how to execute two commands concurrently with a command pipe.

12. Write a C programs that illustrates the following:
a) Two-way communication with unidirectional pipes.
b) Two-way communication with bidirectional pipes

13. Write a C program that illustrates the creation of child process using fork system call.

14. Write a C program that displays the real time of a day every 60 seconds.

15. Write a C program that illustrates file-locking using Semaphores.

16. Write a C program that implements a Producer-Consumer system with two processes. (Using semaphores)

17. Write a C program that illustrates InterProcessCommunication(IPC) using shared memory system calls.

18. Write a C program that illustrates the following.
a) Creating a Message Queue.
b) Writing to a Message Queue.
c) Reading from a Message Queue.

19. Write a C program to develop simple Client /Server application using Sockets(system calls).

Reference Book:

N.B.Venkateswarulu, Advanced Unix Programming, , BS Publications.

MCA401 – CLOUD COMPUTING

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Fundamentals of Cloud Computing
- The Concepts of Virtualization and the Cloud delivery and Deployment Models.
- The fundamental concepts of Cloud computing software security.
- The cloud computing risks and threats.
- Management challenges and opportunities regarding security management in cloud.
- Addresses the Cloud computing security architectural issues.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing.
- Identify the architecture and infrastructure of cloud computing, including cloud delivery and deployment models.
- Explain the core issues of cloud computing such as security, privacy, and interoperability.
- Choose the appropriate technologies, algorithms, and approaches for the related issues.
- Identify problems, and explain, analyze, and evaluate various cloud computing solutions.
- Provide the appropriate cloud computing solutions and recommendations according to the applications used.
- Attempt to generate new ideas and innovations in cloud computing.

SYLLABUS

UNIT - I

Cloud Computing fundamentals: Essential characteristics, Architectural Influences, Technological Influences, Operational Influences.

UNIT - II

Cloud Computing Architecture: Cloud Delivery models, The SPI Framework, Cloud Software as a Service (SaaS) , Cloud Platform as a Service(PaaS), Cloud Infrastructure as a Service(IaaS), Cloud deployment models, Public Clouds, Community Clouds, Hybrid Clouds, Alternative Deployment models, Expected benefits.

UNIT - III

Cloud Computing Software Security fundamentals: Cloud Information Security Objectives, Confidentiality, Integrity, Availability, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Secure Development practices, Approaches to Cloud Software Requirement Engineering, Cloud Security Policy Implementation.

UNIT - IV

Cloud Computing Risk Issues: The CIA Traid**,** Privacy and Compliance Risks, Threats to Infrastructure, Data and Access Control , Cloud Access Control Issues ,Cloud Service Provider Risks.

Cloud Computing Security challenges: Security Policy Implementation, Policy Types, Computer Security Incident Response Team(CSIRT).

UNIT - V

Cloud Computing Security Architecture: Architectural Considerations, General Issues, Trusted Cloud Computing, Secure Execution environments and Communications, Micro architectures, Identity Management and Access Control, Autonomic Security.

TEXT BOOK

Ronald L. Krutz, Russell Dean Vines, "*Cloud Security A comprehensive Guide to secure Cloud Computing*" Wiley.

REFERENCE BOOKS:

1. John W. itinghouse james F.Ransome, *"Cloud Computing Implementation, Management and Security*" , CRC Press.
2. Borko Furht. Armando Escalante, "Handbook of Cloud Computing", Springer
3. Charles Badcock, "Cloud Revolution" , TMH

MCA402 – CRYPTOGRAPHY AND NETWORK SECURITY

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn to

- The objectives of information security
- The symmetric block ciphers (DES, AES, other contemporary symmetric ciphers)
- The public-key cryptosystem (RSA and others), including a necessary complexity-theoretic assumption for its security.
- The Authentication and Hash Functions.
- The Network Security: email, web security and IP Security.
- Intrusions and intrusion detection.
- The System Security: Password Management, viruses, Firewalls, Trusted systems.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Identify and classify computer and security threats and develop a security model to prevent, detect and recover from attacks.
- Encrypt and decrypt messages using block ciphers.
- Implement various cryptographic techniques that provide information and network security.
- Evaluate the security of communication systems, networks and protocols based on a multitude of security metrics.
- Describe the ethical issues related to the misuse of computer security.

SYLLABUS

UNIT - I
INTRODUCTION:
Security Trends, OSI Security Architecture, Security Attacks, Security Services, Security Mechanism, A model for Network Security
Symmetric Cipher:
Classical Encryption Techniques, Block Ciphers, Data Encryption Standard, Advanced Encryption Standard, Triple DES, Placement of Encryption Function, Traffic confidentiality, Key Distribution, Random Number Generation.

UNIT - II
PUBLIC-KEY CRYPTOGRAPHY
Number Theory, Principles of public-key Cryptosystems, RSA, Key Management, Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic and cryptography.

UNIT - III
AUTHENTICATION AND HASH FUNCTIONS
Authentication Requirements and Functions, Message Authentication Codes, Hash Functions, Security of Hash Function and MACs, Secure Hash Algorithm,HMAC,CMAC,Digital Signatures, Authentication Protocols, Digital Signature Standard.

UNIT - IV
NETWORK SECURITY
Authentication Applications: Kerberos,X.509 Authentication Service, Public-Key Infrastructure, Electronic Mail Security: PGP,S/MIME, IP Security, Web Security

UNIT - V
SYSTEM SECURITY
Intruders, Intrusion Detection, Password Management, Viruses and Related Threads, Virus Counter Measure, Firewall Design principles, Trusted Systems.

TEXT BOOK

"Cryptography and Network Security Principles and practices" by William Stallings 4th Edition, Prentice Hall.

REFERENCE BOOKS

1. "Fundamentals of Network Security" by Eric Maiwald Dreamtech press
2. "Principles of Information Security", by Whitman, Thomson
3. "Network Security: the complete reference", by Robert Bragg, Mark Rhodes,TMH
4. "Introduction to Cryptography" by Buchmann, Springer.

MCA403 – OBJECT ORIENTED ANALYSIS & DESIGN

Lecture : 4 Periods/week	**Internal Marks**	**: 40**
	External Marks	**: 60**
Credits : 3	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn to

- Concept and techniques necessary to effectively use system requirements to drive the development of a robust design model.
- Applying UML to fundamental OOAD concepts.
- Forward engineering
- Reverse engineering

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Capture and communicate analysis and design decisions.
- Use object oriented technologies.
- Perform reverse and forward engineering.
- Implement any object oriented design with object oriented programming languages like C++, JAVA etc.
- Manage the complexity of artifacts.

SYLLABUS

UNIT - I

Introduction to UML: Object, Object Orientation, Development, Modeling, Object Modeling, Importance of Modeling, Principles of Modeling, Conceptual model, Model Driven Architecture with UML, Software Development Life Cycle of UML, UML Architecture

UNIT - II

Basic Structural Modeling: Classes, Relationships, Diagrams.
Advanced structural Modeling: Advanced Classes, Advanced relations, Interfaces, Types and Roles

UNIT - III

Class & Object diagrams: Terms, Concepts, Common Modeling techniques for Class & Object diagrams.
Basic Behavioral Modeling –I: Interactions, Interaction diagrams.

UNIT – IV

Basic Behavioral Modeling –II: UseCases, UseCase Diagrams, Activity Diagrams.
Advanced Behavioral Modeling: Events and Signals, State machines, State chart diagrams.

UNIT - V

Architectural Modeling: Component, Development, Component Diagrams, and Deployment Diagrams.

TEXT BOOK

Grady Booch, James Rambaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Edition

REFERENCE BOOKS:

1. Craig Larman," Applying UML and Patterns- An Introduction to Object oriented Analysis and Design and Iterative Development", 3rd Edition Pearson Edition,2004.
2. HansEriksson, Magnus,Penker, BrainLyons, DavidFado:UML2Toolkit, WILEY-Dreamtech India Pvt.Ltd,2007.
3. Meilir Page-Jones:Fundamentals of Object Oriented Design in UML-Pearson Education,2008.
4. Atul Kahate: Object Oriented and Design,The MCAGraw-Hill Company
5. Grady Booch, James Rambaugh, Ivar Jacobson: The Unified Modeling Language Reference Manual, Addison Wesley,1999
6. Object Oriented Analysis and Design Bennett,Simon MCAGraw Hill

\

MCA404 – SOFTWARE TESTING

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course, the students will learn to

- Test process and continuous quality improvement
- Test generation from requirements
- Types and levels of testing.
- Special kinds of tests.
- Quality plan and Test plan preparation.
- Collection of Test metrics.
- Preparation Test Reports.
- Identify uses of various testing tools.

Course Outcomes:

After the completion of this course, prospective engineers will have the ability to

- Understand and apply various levels of testing.
- Perform special kinds of Tests
- Develop Quality plan or Test plan.
- Gather and analyze various test metrics.
- Demonstrate the uses of testing tools.

SYLLABUS

UNIT - I

Basics of Software Testing:

Humans, Errors and Testing, Software Quality, Requirements, Behavior and Correctness, Correctness versus Reliability, Testing and Debugging, Test Metrics, Software and Hardware testing, Testing and Verification, Defect Management, Execution History, Test-Generation stratagies,Static Testing, Control flow Graph, Dominators and Post Dominators, Program dependence Graph.

UNIT -II

Testing Techniques:

Levels of Testing, Acceptance Testing. Special Tests: Complexity Testing, GUI Testing, Security Testing, Performance, Volume and Stress Testing, Recovery Testing, Requirement Testing, Regression Testing, Smoke Testing, Sanity Testing, Adhoc Testing, State Graph, Object-Oriented Applications Testing, COTS Testing.

UNIT - III

Test Planning:

Test policy, Test strategy, Test plan, Quality plan and Test plan, Quality plan template, Test plan template, Guidelines for developing the Test plan, Test Standards, Building Test data and Test cases, Test scenario, Test cases, Template for Test cases, Test scripts, Test Log Document, Effective Test cases, Test file, Building Test data, Generation of Test data, Roles and Responsibilities in Testing life cycle, Test progress monitoring.

UNIT - IV

Test Metrics and Test Reports:

Test Metrics and Test Reports, Categories of the Product/Project Test Metrics, Estimated, Budgeted, Approved and Actual, Resources Consumed in Testing, Effectiveness of Testing, Defect Density, and Defect Leakage Ratio, Residual Defect Density, Test team efficiency, Test case efficiency, Rework ,MTBF/MTTR, Implementing Measurement Reporting System in an Organization, Test Reports, Project Test Status Report,
Test Reports: Integration Test Report, System Test Report, Acceptance Test Report, Guidelines for writing and using Report, final Test Reporting, Test Status Report.

UNIT - V

Test process Improvement:

The need for Test process Improvement, Test process Maturity, Test process Improvement Model, Test process Improvement Model stages, Graphical representation of Improvements.

Testing Tools:

Introduction, Features of Test Tool, Guidelines for selecting a Tool, Tools and Skills of Tester, Static Testing Tools, Dynamic Testing Tools, Advantages of using Tools, When to use Automated Test Tools.

Text Book:

M G Limaye, “Software Testing, Principles, Techniques, and Tools”, (TMH),2011.

Reference Books:

1. Aditya P. Mathur Foundations of Software Testing. –,2011.
2. Baris Beizer, “Software Testing techniques”, Dreamtech, second edition.
3. Dr.K.V.K.K.Prasad, “Software Testing Tools “, Dreamtech,2007.
4. Edward Kit, “Software Testing in the Real World”, Pearson,2006.
5. Perry, “ Effective Methods of Software Testing”, John Wiley, 3rd edition, 2007.

MCA405 – WEB TECHNOLOGIES

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
Tutorial	**: 1 Period/Week**	**External Marks**	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course, the students will learn to

- Developing the web sites which are secure and dynamic in nature and writing scripts which get executed on server as Well.
- The web page site planning, management and maintenance.
- Developing Internet based applications using Java Technologies.

Course Outcomes:

After the completion of this course, prospective engineers will have the ability to

- Develop web Application Using **HTML** and **XML**.
- Perform Client side validation using **Java Script.**
- Develop server side applications using servlets and JSP.
- Install and run web servers
- Deploy a web based application onto the web server.

SYLLABUS

UNIT - I
HTML:
Introduction, Common tags, HTML Tables and formatting internal linking, Complex HTML forms. Introduction to Scripting Languages, Java script functions, Arrays & Objects, Control structures

UNIT - II
DHTML:
Introduction , CSS, Event model
XML:
Introduction, DTD, Schema, Parsers: DOM and SAX.
Introduction to AJAX

UNIT - III
JDBC:
Database Programming using JDBC, Javax.sql.* package, JDBC Drivers, JDBC applications using select, insert, delete, update, Types of Statements (Statement, Prepared Statement and Callable Statement); ResultSet, ResultSetMetaData, Inserting and updating records,
BDK:
Introduction to Java Beans, Advantages of Java Beans,Introspection,Bound properties, Bean Info Interface, Constrained properties, Persistence, Customizers, Java Beans API, Introduction to EJB.

UNIT - IV

Servlets:

Introduction, Servlet Basics, Types of Servlets and Life cycle,Servlet API Overview; Writing and running Simple Servlet. ServletConfig & ServletContext, Writing Servlet to handle Get and Post Methods, Reading user request ,Writing thread safe Servlets,Concept of cookies, Reading and writing cookies, Need of Session Management, Types of Session management-Using HttpSession Object , Servlet chaining , Servlet using JDBC,

TOMCAAT:

How to configure TOMCAAT, Directory structure for a web Application

UNIT - V

JSP:

Servlet drawbacks, Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC.Introduction to JSP and JSP Basics, Implicit Objects,JSP Tags, Life cycle of JSP, JSP and Java Beans, JSP:sessions and cookies, Error Handling with JSP, JDBC with JSP,Introduction to Struts

Text Book :

Harvey M. Deitel and Paul J. Deitel, **“Internet & World Wide Web How to Program”, 4/e,** Pearson Education.

Reference Books:

1. J. MCAGovern. Adatia,Y. Fain, 2003, J2EE 1.4 Bible, Wiley-dreamtech India Pvt. Ltd, New Delhi.2. H. Schildt, 2002,
2. JAVA Complete Reference, 5th Edition, Tata MCAGraw-Hill, New Delhi.
3. K. Moss, 1999, Java Servlets, Second edition, Tata MCAGraw Hill, New Delhi.
4. D. R.Callaway, 1999, Inside Servlets, Addison Wesley, Boston
5. Joseph O’Neil, 1998, Java Beans from the Ground Up, Tata MCAGraw Hill
6. TomValesky, Enterprise JavaBeans, Addison Wesley. 7. Cay S Horseman & Gary Cornell, Core Java Vol II Advanced Features, Addison Wesley.

MCA4061 – COMPUTER GRAPHICS AND VISION

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn to

- Internal workings of commercial systems for the rendering of digital images from 3D models
- 3D modeling and rendering
- use 3D graphics API's
- undertake creative work and research in 3D graphics

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Appreciate the history and evolution of computer graphics, both hardware and software. Assessed by written homework assignment.
- Implement 2D Graphical algorithms including: line drawing, polygon filling, clipping, and transformations.
- Implement concepts and techniques used in 3D computer graphics, including viewing transformations, hierarchical modeling, color, lighting and texture mapping. Students will be exposed to current computer graphics research areas. Assessed by tests, homework and programming assignments.
- Use a current graphics API (OpenGL). Assessed by programming assignments.

SYLLABUS:

UNIT - I

Introduction: Application areas of Computer Graphics, Overview of graphics systems, Video-display devices, Raster-scan systems, Random scan systems.

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

2D&3D Geometrical Transformations: Translation, Scaling, Rotation, Reflection and Shear transformations, Matrix representations and homogeneous coordinates, Composite transformations, Transformations between coordinate systems, Cohen-Sutherland line clipping algorithm, Sutherland–Hodgeman polygon clipping algorithm.

UNIT - III

Introduction to Digital Image Processing : Examples of fields that use Digital image processing, Fundamental steps in Digital image processing, Components of image processing system.

Digital Image Fundamentals: A simple image formation model, Image sampling and quantization, Basic relationships between pixels, Color models.

UNIT - IV

Image enhancement in the spatial domain: Basic gray-level transformation, Histogram processing, Enhancement using arithmetic and logic operators, Basic spatial filtering, Smoothing and sharpening spatial filters, Combining the spatial enhancement methods

UNIT - V

Morphological Image Processing: Preliminaries, Dilation, Erosion, Open and Closing, Hit or miss transformation, Basic morphologic algorithms

Image Segmentation: Detection of discontinuous, Edge linking and boundary detection, Thresholding, Region–based segmentation

TEXT BOOKS

1.“Computer Graphics C version”, Donald Hearn and M.Pauline Baker, Pearson Education. (Units I & II)

2.Digital Image Processing, Rafeal C.Gonzalez, Richard E.Woods, Second Edition, Pearson Education/PHI. (Units III,IV & V)

REFERENCE BOOKS

1. Procedural elements for Computer Graphics, David F Rogers, Tata MCA Graw hill, 2nd edition.
2. Computer Graphics, Steven Harrington, TMH
3. Image Processing, Analysis, and Machine Vision, Milan Sonka, Vaclav Hlavac and Roger Boyle, Second Edition, Thomson Learning.
4. Introduction to Digital Image Processing with Matlab, Alasdair MCAAndrew, Thomson Course Technology
5. Computer Vision and Image Processing, Adrian Low, Second Edition, B.S.Publications

MCA4062 – DISTRIBUTED OPERATING SYSTEMS

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- The types of problems which arise and methods used in the design and analysis of systems of interconnected computers.
- Distributed systems and their characteristics, and the developments in distributed systems.
- Distributed algorithms for locking, synchronization and concurrency, scheduling, and replication.
- Investigating problems of timing and event ordering, naming of objects, and distribution of objects
- Inter-process communication in a distributed environment.
- How to fault-tolerance can be enhanced with concurrency control mechanisms and replication of services.
- Distributed file systems.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Describe important characteristics of distributed systems and the salient architectural features of such systems.
- Describe the features and applications of important standard protocols which are used in distributed systems.
- Characterise different implementation paradigms for distributed systems.
- Explain the working of a distributed algorithm.
- Perform simple proofs of system properties, given a formalised description of a system.
- Select an appropriate distributed algorithm to satisfy given design requirements for a distributed system.
- Select an appropriate implementation paradigm to satisfy given design requirements for a distributed system.
- Develop an implementation of a distributed system from a formal or informal description of its function and purpose.

SYLLABUS:

UNIT - I

Introduction

Definition of a DOS, Goals, H/w and S/w Concepts, Client-Server Model

Processes Threads: Introduction to Threads, Threads in Distributed Systems, Clients: User Interfaces, Client-Side Software for Distribution Transparency; Servers: General Design Issues, Object Servers; Code Migration: Approaches to Code Migration, Migration and Local Resources, Migration in Heterogeneous Systems; Software Agents: Software Agents in Distributed Systems, Agent Technology.

UNIT - II

Naming Systems:

Naming Entities: Names, Identifiers, and Addresses, Name Resolution, The Implementation of a Name Space, Example: DNS, X.500 Locating Mobile Entities**:** Naming versus Locating Entities, Simple Solutions, Home-Based Approaches, Hierarchical Approaches Clock synchronization, logical clocks, global state, election algorithms, mutual exclusion.

UNIT - III

Consistency and Replication:

Introduction, Data-Centric Consistency Models, Client-Centric Consistency Models, Distribution Protocols, Consistency Protocols.

Fault Tolerance:

Introduction to Fault Tolerance, Process Resilience, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit.

UNIT - IV

Distributed File System

Sun Network File System, Coda File System, Plan~9, XFS and SFS, Scalable Security. Distributed Shared memory: Introduction, Bus based multi processors, Ring based multiprocessors, Switched multiprocessors - NUMA comparison of shared memory systems.

UNIT - V

Distributed Object Based System

CORBA, Distributed Com, Globe and Comparison of CORBA, DCOM.

Distributed Document-Based System and Coordinate Based System

The World Wide Web, Lotus Notes, Comparison of WWW and Lotus Notes.

TEXT BOOK

Distributed Systems, Principles and paradigms, 2/e Tanenbaum,Maarten Vansteen, Pearson education,2009.

REFERENCE BOOKS

1. Andrew S.Tanenbaum: Distributed Operating System, Prentice Hall International Inc. 1995,MCAGrawHill.,2007.
2. Distributed Operating Systems & Algorithm Analysis, Chow, Johnson,PEA.
3. Distributed Systems Concepts and Design 4/e , George coulouris, Dollimore ,Kindberg ,PEA
4. Distributed Operating Systems ,Pradeep K.Sinha ,PHI,2001.

MCA4063 – PRINCIPLES OF PROGRAMMING LANGUAGES

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about:

- Fundamental concepts that underlies in most programming languages.
- The tradeoff between language design and implementation
- The language implementation: how are programs parsed and translated by a compiler?
- How to program in imperative, functional, object-oriented programming languages.
- To provide an introduction to formalisms for specifying syntax and semantics of programming languages, including an introduction to the theory of formal languages,
- To provide an exposure to core concepts and principles in contemporary programming languages
- To explore various important programming methodologies, such as functional programming, logic programming, programming with abstract data types, and object-oriented programming.

Course Outcomes:

At the end of this course student will be able to:

- Knowledge of, and ability to use, language features used in current programming languages.
- An ability to program in different language paradigms and evaluate their relative benefits.
- An understanding of the key concepts in the implementation of common features of programming languages.
- Implement recursive programs in a functional language to manipulate data structures such as lists and trees

- Structure a labeled tree data structure, such as an abstraction syntax tree, using datatypes.
- Implement a recursive program such as an interpreter, pretty-printer, parser, and typechecker for functional, imperative or object-oriented languages.

- Implement parameter-passing mechanisms such call-by-value, call-by-result, call-by-value-result, call-by-reference, call-by-name and call-by-need and be able to reason about the behavior of programs that use those conventions.
- Implement static and dynamic scoping rules in programming languages

SYLLABUS

UNIT - I

Preliminary Concepts: Reasons for studying, Concepts of programming languages, Programming domains, Language Evaluation Criteria, Influences on Language design, Language categories, Programming Paradigms– Imperative, Object Oriented, functional Programming , Logic Programming. Programming Language Implementation–Compilation and Virtual Machines, programming environments.

UNIT - II

Syntax and Semantics: General Problem of describing Syntax and Semantics, Formal methods of describing syntax - BNF, EBNF for common programming languages features, Parse trees, Ambiguous grammars, Attribute grammars, Denotational semantics and Axiomatic semantics for common programming language features. Names, Variable, concept of binding, type checking, Strong typing, Type compatibility, Named constants, Variable initialization. Data types: Introduction, Primitive, Character, User defined, Array, Associative, Record, Union, Pointer and Reference types, Design and Implementation issues related to these types.

UNIT - III

Expressions and Statements: Arithmetic, Relational and Boolean expressions, Short circuit evaluation, Mixed mode assignment, Assignment Statements. Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, Guarded commands. **Subprograms and Blocks**: Fundamentals of sub-programs, Scope and lifetime of variable, Static and Dynamic scope, Design issues of subprograms and operations, Local referencing environments, Parameter passing methods, Overloaded sub-programs, Generic sub-programs, Parameters that are sub-program names, Design issues for functions, User defined overloaded operators, Co-routines.

UNIT - IV

Abstract Data types: Abstractions and encapsulation, introduction to data abstraction, Design issues, Language examples, C++ parameterized ADT.
Object oriented programming in C++,Java,
Concurrency: Subprogram level concurrency, Semaphores, Monitors, Message passing, Java threads, C# threads.

UNIT - V :

Exception handling : Exceptions, Exception Propagation, Exception handling in Ada, C++ and Java.
Functional Programming Languages: Introduction, Fundamentals of FPL, LISP, ML, Haskell, Application of Functional Programming Languages and comparison of functional and imperative Languages.
Logic Programming Language : Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

TEXT BOOK

Robert .W. Sebesta , “Concepts of Programming Languages”, 4/e, Addison Wesley

REFERENCES

1. Ghezzi , “Programming languages”, 3/e, John Wiley
2. Pratt and Zelkowitz, “Programming Languages Design and Implementation”, Fourth Edition, PHI/Pearson Education
3. Watt, Wiley, “Programming languages”, Dreamtech
4. Henry Winston and Paul Horn, “LISP Patric”, Pearson Education,2000.
5. Clocksin,”Programming in PROLOG”, Springer,1997.

MCA4064 – SOFTWARE DESIGN METHODOLOGIES

Lecture : 4 Periods/week	**Internal Marks : 40**
	External Marks : 60
Credits : 3	**External Examination : 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Software Design methodology
- Software Architectural styles
- The basic characteristics of Software Design processes;
- The elements of Software Designs;
- The factors that affect Software Design processes and outcomes.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Evaluate various Architectural Styles
- Create new Architectural styles for specific purposes
- Perform the Software Design analytically.

SYLLABUS:

UNIT - I

Basic concepts of Design:
Introduction, Characteristics of Design activities, Essential elements of Designs.
Design Quality:
Software Quality models: Hierarchical models, Relational models,
The effect of Design on software quality, Efficiency, Correctness and Reliability, Portability, Maintainability, Reusability, Interoperability, Quality attributes of software Design, Witt, Baker and Merritt's Design objectives, Parnas and Weiss's requirements of good Designs, Quality of development process
Design Principles:
Basic rules of software Design: Causes of difficulties, Vehicles to overcome difficulties, Basic rules of software Design
Design processes: The context of Design in Software development process, Generic Design process, Descriptive models, Structure of software Design methods

UNIT - II

Software Architecture:
The notion of Architecture: Architecture in the discipline of buildings, Architecture in the discipline of computer hardware, General notion of architecture. The notion of software architecture, Prescriptive models, Descriptive models, Multiple view models, Roles of architecture in software Design. Software architectural style, Introductory examples, the notion of software architectural style.
Description of Software Architectures:
The Visual Notation: Active and Passive elements, Data and control Relationships, Decomposition/Composition of Architectural elements

UNIT - III

Typical Architectural Styles:

Data flow: General data flow styles, The pipe- and filter sub-style, The batch sequential processing ,sub-style Independent components: the general independent components style, the event-based implicit invocation systems sub-style.

Call and return**:**

The general call and return style, the layered systems sub-style, Data Abstraction: the abstract data type and object-oriented sub-style

Data-centered style, Virtual machine Architecture

Using Styles in Design:

Choices of styles, Combinations of styles: Hierarchical heterogeneous styles, Simultaneously heterogeneous styles, Locationally heterogeneous styles

UNIT - IV

Architectural Design space:

Theory of Design spaces: Structure of Design spaces, Solving Design synthesis and analysis problems ;Design space of architectural elements: Behavior features, Static features, Design space of architectural styles,Characteristic features of architectural styles, Classification of styles

Scenario-Based Analysis and Evaluation:

The concept of scenario,Scenarios for evaluating modifiability, Scenarios for evaluating Performance, Scenarios for evaluating reusability.

UNIT - V

Analysis and Evaluation of Modifiability: **SAAM Method**:

The input and output, the process (Activities in SAAM Analysis)

Quality Trade- Off Analysis: ATAM Method

ATAM analysis process, ATAM analysis activities

Model-Based Analysis: HASARD Method

Representation of quality models, construction of quality models.

TEXT BOOK

Software Design Methodology: From Principles to Architectural Styles , Hong zhu, Elsevier,2006.

REFERENCE BOOKS

1. Software Architecture: Perspectives on an Emerging discipline, Shaw, M.,Garlan, PEA, 2010.
2. Software Architecture in Practice, Bass, L., Clements P,Kazman, PEA,2003
3. Evaluating Software Architectures: Methods and Case Studies, Clements, Kazman, Klien, PEA, 2002
4. Tutorial on Software Design Techniques, Freeman, Wasserman, A.I.(Es), IEEE, 1980
5. Design and Use of Software Architectures- Adopting and Evolving a product – Line Approach, Bosch, J., ACM Press , Addison Wesley, 2000
6. Software Architecture and Design, Bernard Witt, Baker, Merritt, Von Nostrand Reinhold, NY, 1994.

MCA452 – OBJECT ORIENTED ANALYSIS & DESIGN USING UML LAB.

Lab/Practical	**: 4 Period/Week**	**Internal Marks**	**: 25**
		External Marks	**: 50**
Credits	**: 2**	**External Examination**	**: 3Hrs**

Course Educational Objectives

In this course, the students will learn to

- Concept and techniques necessary to effectively use system requirements to drive the development of a robust design model.
- Applying UML to fundamental OOAD concepts.
- Forward engineering
- Reverse engineering

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Capture and communicate analysis and design decisions.
- Use object oriented technologies.
- Perform reverse and forward engineering.
- Implement any object oriented design with object oriented programming languages like C++, JAVA etc.
- Manage the complexity of artifacts.

SYLLABUS

1. Library Information System.
2. University Model
3. ATM Transactions
4. Cell Phone Networking System
5. Hospital Management System

MCA453 –WEB TECHNOLOGES LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Period/Week**	**External Marks**	**: 50**
Credits	**: 2**	**External Examination**	**: 3 Hrs**

Course Educational Objectives

In this course, the students will learn to

- Developing the web sites which are secure and dynamic in nature and writing scripts which get executed on server as Well.
- The web page site planning, management and maintenance.
- Developing Internet based applications using Java Technologies.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Develop web Application Using **HTML** and **XML**.
- Perform Client side validation using **Java Script.**
- Develop server side applications using servlets and JSP.
- Install and run web servers
- Deploy a web based application onto the web server.

SYLLABUS

1. Write a HTML program to create a Table.
2. Write a HTML program to create Lists.
3. Write a HTML program using Frames
4. Write a HTML program to develop a web page to fill student information
5. Write a HTML program that use CSS.
6. Write a HTML program to reverse a given number.
7. Write a Java script program to find that a given number is prime or not.
8. Write a Java script program to find that a given number is Armstrong or not.
9. Write a Java script program to find the factorial of a number using recursion.
10. Write a Java script program by using Java script Objects.
11. Write a HTML program that handles the events.
12. Write an XML program using DTDs
13. Develop a web page to implement online book stores using XML
14. Write a Java program to retrieve data from data base using Type-1 and Type-4 drivers.
15. Write Java program by using Prepared Statements and Callable Statements.
16. Write a Java program using forward only and bi-directional Resultsets
17. Write an example program using BDK.
18. Write a simple Servlet program using Generic and HTTP Servlets.
19. Write a Servlet program that handles the user request by using doGet () and doPost () methods.
20. Write a Servlet program using Config and Context parameters.
21. Write a Servlet program to implement Session Tracking.
22. Write a Servlet program that uses JDBC.
23. Write a simple JSP program to display Date.
24. A) Write a JSP program by using Implicit objects.
    B) Write a JSP program to handle Exceptions.
25. Write a JSP program using JDBC.
26. Write a JSP program using Include, Forward requests.
27. Write a JSP program using useBean.
28. Write a web application program using struts frame work
29. Develop an application using EJB.

MCA501 – ARTIFICIAL INTELLIGENCE

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn to

- The basic knowledge representation, problem solving, and learning methods of Artificial Intelligence
- Assess the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, and learning methods in solving particular engineering problems
- Develop intelligent systems by assembling solutions to concrete computational problems
- produce highly competent computer scientists, knowledge and software specialists and systems analyst who are able to develop, maintain, and utilize intelligent systems in e-learning, e-commerce, tele-medicine, automation, and bio-technology industries.
- Produce leaders, critical thinkers and technopreneurs in artificial intelligence for the knowledge economy.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Good understanding of fundamental concepts, Prepare students to strengthen their basic knowledge and competencies in AI.
- Acquisition and mastery of knowledge in specialized area which, is designed to achieve the breadth of knowledge through carefully selected core subjects which pertains to the fields of specialization.
- Acquire the Basic Knowledge about the developing projects include Artificial Intelligent, Neural Networks, Expert System, Natural Language Processing, Machine Learning, Pattern Recognition, Computer Vision, Agent Technology, Speech Processing etc.
- Acquisition of analytical capabilities and problem solving skills. Students would need to apply their analytical and problem solving skills when dealing with their Final Year Project
- Predict the behavior of forward-chaining and backward-chaining rule-based systems
- Predict the behavior and estimate the cost in time and space of various heuristic and optimal search methods (depth-first, breadth-first, hill-climbing, branch-and-bound, and A*), and choose the appropriate method for particular problems
- Develop small rule-based and search-based expert systems, predict performance characteristics, and describe the role of rule-chaining and search in intelligent-system engineering
- Use rules and frames to represent behavioral, classification, and causal knowledge
- Apply basic machine learning methods such as identification trees, neural nets, and genetic algorithms
- Predict the behavior of the basic machine-learning methods, and choose the appropriate method for particular problems
- Modify and extend simple implementations of the subject's representations and methods
- Develop small learning systems, predict performance characteristics, and describe the role of learning in intelligent-system engineering.

SYLLABUS

UNIT - I

Introduction:
The AI Problems, The underlying Assumption
Problems, Problem Spaces, and Search:
Defining the problem as a State Space Search, Production Systems, Problem Characteristics, Production System Characteristics, Issues in the Design of Search Programs.
Heuristic Search Techniques:
Generate- and-Test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means-ends Analysis.

UNIT - II

Knowledge Representation Issues:
Representations and Mappings, Approaches to Knowledge Representation, Issues in Knowledge Representation.
Using Predicate Logic:
Representing Simple Facts in Logic, Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution.
Representing Knowledge Using Rules:
Procedural versus Declarative Knowledge**,** Logic Programming, Forward Versus Backward Reasoning, Matching.

UNIT - III

Symbolic reasoning Under Uncertainty:
Introduction to Nonmonotonic Reasoning, Logics for Nonmonotonic Reasoning, Implementation Issues, Augmenting a problem-solver,
Depth-First search, Breadth-First search.
Statistical Reasoning:
Probability and Baye's Theorem, Certainly Factors and Rule-based Systems, Bayesian Networks.
Weak Slot-and-Filler Structures: Semantic Nets, Frames.
Strong Slot-and Filler Structures: Conceptual Dependency, Scripts.

UNIT - IV

Game Playing:
The Minimax search Procedure, Adding Alpha-beta Cutoffs.
Planning:
An Example Domain: The Blocks World, Components of a Planning System,
Goal Stack Planning.
Natural language Processing, **Learning.**

UNIT - V

Expert Systems, Perception and Action
Introduction to Neural Networks:
Biological Neuron structure,Basic Artificial Neuron Models.
Types of Neural Networks, Applications of Neural Networks

TEXT BOOK

ElaineRich KevinKnight, Shivashankar B Nair "Artificial Intelligence", 3rd Edition, TMH, 2010.

REFERENCE BOOKS

1. Stuart Russell, Peter Norvig, "Artificial Intelligence – A Modern Approach", Second Edition, Pearson Education / Prentice Hall of India, 2008.

2. George F.Luger, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Pearson Education / PHI, 2010.

MCA502 – BUSINESS INTELLIGENCE

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Data Warehouse and OLAP Technology for Data Mining
- Data Preparation
- Data Mining Primitives, Languages, and System Architectures
- Concept Description: Characterization and Comparison
- Mining Association Rules in Large Databases (compressed)
- Classification and Prediction
- Cluster Analysis
- Mining Complex Types of Data
- Data Mining Applications and Trends in Data Mining

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Define knowledge discovery and data mining
- Recognize the key areas and issues in data mining
- Apply the techniques of clustering, classification, association finding, feature selection and visualization to real world data
- Determine whether a real world problem has a data mining solution
- Apply evaluation metrics to select data mining techniques

SYLLABUS

UNIT - I

Introduction

Fundamentals of Data Mining, Data Mining functionalities, Classification of Data Mining Systems, Data mining applications, Data Warehouse and OLAP Technology, Multidimensional data Model, Data warehouse architecture.

UNIT - II

Data preprocessing: Data cleaning, Data Integration and Tranformation, Data Reduction, Discretization and concept Hierarchy generation, Data Mining primitives, Data Generalization and Summerization, Basic Characterization, attribute relevants analysis, Mining descriptive statistical measures, Data Mining query Languages.

UNIT - III

Association Rule Mining & Market Basket Analysis, Efficient and scalable Frequent Item Set Mining methods.(Apriory and FP growth), Mining various kinds of Association rules

UNIT - IV

Classification and Prediction, Classification by Decision tree induction, Bayesian classification, Prediction: Linear regression, Non-Linear regression.

UNIT - V

Cluster analysis: Types of Data in Cluster analysis, Categorization of clustering methods, Partitioning methods, Outlier analysis, Text Mining, Web Mining.

TEXT BOOK

Data Mining ,Concepts and Techniques ,Jiawei Han, Micheline Kamber,Harcourt India,2006,Elsevier.

REFERENCE BOOKS

1. Data Mining, Introductory & advanced Topics, Margaret H Dunham, Pearson,2006.
2. Data Mining Techniques, Arun K Pujari, University Press,2007.
3. Data warehousing Fundamentals, Paulraj Ponnaiah, Wiley,2006.
4. The Data Warehouse Life Cycle Tool kit, Ralph Kimball, Wiley.

MCA503– MOBILE APPLICATION DEVELOPMENT

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Mobile Applications development using Android
- App User interface designing
- Testing the mobile Apps.
- Distribute and market mobile Apps

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Appreciate the Mobility landscape
- Familiarize with Mobile apps development aspects
- Design and developmobile apps, using Android as development platform, with key focus on user experience design, native data handling and background tasks and notifications.
- Appreciation of nuances such as native hardware play, location awareness, graphics, and multimedia.
- Perform testing, signing, packaging and distribution of mobile apps

SYLLABUS

Unit 1:
Getting started with Mobility
Mobility landscape, Mobile platforms, Mobile appsdevelopment, Overview of Android platform, setting up the mobile app development environment along with an emulator, a case study on Mobile app development

Unit II:
Building blocks of mobile apps
App user interface designing – mobile UI resources (Layout,UI elements,Draw-able, Menu), Activity- states and life cycle, interaction amongst activities.

App functionality beyond user interface - Threads,Async task,Services – states and lifecycle, Notifications, Broadcastreceivers,Telephony and SMS APIs

Native data handling – on-device file I/O, shared preferences, mobile databases such as SQLite, and enterprise data access (via Internet/Intranet)

Unit III:
Sprucing up mobile apps
Graphics and animation – custom views, canvas, animation APIs, multimedia – audio/video playback and record, location awareness, and native hardware access (sensors such as accelerometer and gyroscope)

Unit IV:

Testing mobile apps

Debugging mobile apps, White box testing, Black box testing, and test automation of mobile apps, JUnit for Android, Robotium, MonkeyTalk

Unit V:

Taking apps to Market

Versioning, signing and packaging mobile apps, distributing apps on mobile market place

TEXT BOOK

Anubhav Pradhan, Anil V Deshpande, ‘Mobile Applications Development’ Edition 1

REFERENCE BOOKS

1. Barry Burd ‘Android Applications Development all in one for Dummies”,e dition 1
2. “Teach Your self Android Application Development in 24 hours” Edition 1, SAMS.

MCA5041 – DATABASE TUNING

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Database internals and their impact on performance.
- The recurring principles underlying database tuning.
- Principled approach to database tuning problems. This includes both discussion of case studies and hands-on experiments.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Understand the parameters that impact the performance of a database system.
- Monitor performance relevant parameters.
- Interpret performance parameters correctly and pinpoint bottlenecks.
- Propose effective solutions to performance problems.
- Discuss tuning issues related to common underlying components of all database systems.
- iscuss and use techniques for query evaluation and optimization

SYLLABUS

UNIT - I

Basic Principles : The Power of Principles, Five Basic Principles, Basic Principles and Knowledge, Tuning The Guts , Locking and Concurrency Control, Logging and the Recovery Subsystem, Operating System Considerations, Hardware Tuning.

UNIT - II

Index Tuning, Types of Queries, Key Types, Data Structures, Sparse Versus Dense Indexes, To Cluster or Not to Cluster, Joins, Foreign key Constraints, and Indexes, Avoid Indexes on small Tables.

UNIT - III

Tuning Relational Systems, Table Schema and Normalization**,** Clustering Two tables**,** Aggregate Maintenance, Record Layout, Query Tuning, Triggers. Communicating with the outside Client-server Mechanisms, Objects, application Tools, and Performance, Tuning the application Interface, Bulk Loading Data, Accessing Multiple Databases.

UNIT - IV

Troubleshooting: Introduction, How to gather Information: The Tools, Queries from Hell, Are DBMS Subsystems Working Satisfactorily, Is the DBMS Getting All It Needs.

UNIT - V

Transaction Chopping : Assumptions, Correct Choppings, Finding the Finest Chopping, Optimal Chopping Algorithm, Application to Typical Database Systems, Related Work.
Time Series, Especially for Finance: Setting up a Time Series Database, FAME,S-Plus, SAS, KDB, Oracle-8i Time Series, features you want for Time Series, Time Series Data Mining.
Understanding access Plans: Data Access Operators, Query structure Operators, Auxiliary Operators.
Configuration Parameters: Oracle, SQL Server, DB2 UDB.

TEXT BOOK

Dennis Shasha and Philippe Bonnet "Database Tuning, Principles, Experiments and Troubleshooting Techniques", Morgan Kaufmann, Elsevier.

REFERENCE BOOKS

3. Thomas Connoly and Carlolyn Begg,"Database Systems, A Practical Approach to Design, Implementation and Management", Third Edition, Pearson Education.
4. M.Tamer Ozsu, Patrick Valduriez and S.Sridhar "Principles of Distributed Database Systems", Pearson Education.

MCA5042 – E-COMMERCE

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Importance of information systems for business and management:
- Evaluate the role of the major types of information systems in a business environment and their relationship to each other;
- Assess the impact of the Internet and Internet technology on business-electronic commerce and electronic business;
- Identify the major management challenges to building and using information systems and learn how to find appropriate solutions to those challenges.
- Define an infrastructure and describe its components
- Learn the core activities in the systems development process;
- Cultivate skills and experience in the development and implementation of information systems projects.

Course Outcomes

Upon completion of this course, students will be able to:

- Understand the basic concepts and technologies used in the field of management information systems;.
- Have the knowledge of the different types of management information systems;.
- Understand the process of developing and implementing information systems;
- Be aware of the ethical,social,and security issues of information systems;
- Understand the role of information systems in organizations, the strategic management processes, and the implications for the management;
- Develop an understanding of how various information systems work together to accomplish the information objectives of an organization;
- Learn about the importance of managing organizational change associated with information systems implementation;
- Use the application software skills such as analyzing spreadsheets, creating database, and web browsing, that they have learned in other courses to apply to real-world business problems.

SYLLABUS

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, Supplychain Management.

UNIT - IV

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

Advertising and Marketing - Information based marketing, Advertising on Internet, on-line marketing process, market research.

UNIT - V

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

Multimedia - key multimedia concepts, Digital Video and electronic Commerce, Desktop video 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 – Thomson.
5. E-Commerce – Business, Technology, Society, Kenneth C.Taudon, Carol Guyerico

MCA5043 – OBJECT ORIENTED SOFTWARE ENGINEERING

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Software requirements elicitation and specification
- Domain modeling to help understand and communicate domain knowledge
- Deriving use cases from requirements
- Specifying actor-system interaction behavior
- Object interaction modeling
- Assigning responsibilities to objects
- Deriving design class diagram
- Deriving an implementation order
- Deriving code from diagrams

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Employ formal methods to produce effective software designs as solutions to specific tasks
- Develop structured sets of simple user-defined classes using Object-Oriented principles to achieve overall programming goals.
- Develop error identification and testing strategies for code development.
- Employ group working skills - including general organization, planning and time management, inter-group negotiation, etc.
- Translate a specification into a design, and then realize that design practically, all using an appropriate software engineering methodology.
- Reflect on the appropriateness of different software engineering methodologies in different circumstances.
- Demonstrate knowledge of the wider software engineering context, software engineering processes and their applicability.

SYLLABUS

UNIT - I

Introduction to Classical software Engineering: Historical, Economic and Maintenance aspects. Introduction to OO Paradigm: Different phases in structured paradigm and OO Paradigm. Software Process and different life cycle models and corresponding strengths and weaknesses.

UNIT - II

Planning and Estimation: Estimation of Duration and Cost , COCOMO components of software. Project Management plan, one case Study. Tools for step wised refinement : Cost - Benefit analysis, Introduction to software metrics and CASE tools. Taxonomy and scope of CASE tools. Introduction to testing with focus on Utility, Reliability, Robustness, Performance, Correctness.

UNIT - III

Modules to objects: Cohesion and Coupling, Data Encapsulation and Information hiding aspects of Objects. Inheritance, polymorphism and Dynamic Binding aspects. Cohesion and coupling of objects. Reusability, Portability and Interoperability aspects.

UNIT – IV

Requirement phase: Rapid Prototyping method, Specification phase , Specification Document, Formal methods of developing specification document, Examples of other semi - formal methods of using Finite-State-Machines, Petri nets Analysis phase: Use case Modeling, Class Modeling , Dynamic Modeling, Testing during OO Analysis Design phase: Data oriented design, Object Oriented design, Formal techniques for detailed design. Onecase study. Challenges in design phase.

UNIT - V

IIM Phases: Implementation, Integration and maintenance phases, OOSE aspects in these phases

<u>TEXT BOOKS</u>

1. Object oriented and Classical Software Engineering, 7/e, Stephen R. Schach, TMH

2. Object oriented and classical software Engineering, Timothy Lethbridge, Robert , Laganiere, TMH

<u>REFERENCE BOOKS</u>

1. Component-based software engineering: 7th international symposium, CBSE 2004, Ivica Crnkovic,Springer

MCA5044 – OPEN SOURCE SOFTWARE

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives:

In this course student will learn about:

- An overview of the historical and modern context and operation of free and open source software (FOSS) communities and associated software projects.
- How to participate in a FOSS project in order to contribute to and improve aspects of the software that they feel are wrong.
- Learn some important FOSS tools and techniques for contributing to projects and how to set up their own FOSS projects.

Course Outcomes:

At the end of this course student will be able to:

- Install and run open-source operating systems
- Gather information about Free and Open Source Software projects from software releases and from sites on the internet.
- Build and modify one or more Free and Open Source Software packages.
- Use a version control system and interface with version control systems used by development communities.
- Interact with Free and Open Source Software development projects.

SYLLABUS

UNIT - I

Open Source Software: Definitions & History- Definitions of terms, A Brief History of Software

Where Open Source Is Successful –Analytical Framework, Open Source in widespread successful use, Examples of Open Source Systems

Open Source: The Good, the Bad, and the Ugly-What is Good about Open source, Open Source is Not enough by itself, How Choosing Open Source Is More difficult for You, What Others Say about Open Source.

UNIT - II

Five Open Source Opportunities – Introduction, Directory Services, Email, Groupware and Collaboration, Complex Web Publishing, Manage User Desktops, Other Possibilities

Operating Systems – Contents of the Operating systems, Linux Distribution Vendors, Enterprise Distribution Vendors, Community-Supported Distribution Vendors, International Alternatives

UNIT - III

Open Source Server Applications – Infrastructure Services, Web Services, Database Servers, Mail Servers, System Management

Open Source Desktop Applications – Introduction, Graphical Desktops, Web Browsers, The Office Suite, Mail and Calendar Clients, Personal Software

UNIT - IV

How Open Source Software is Developed – Methodology, Languages Used to Develop Open source Products, Cross-Platform Code

Application Architecture –Types of Systems, Tiered Design, Managing Performance and Scalability, Interoperability, Development Platform Choices

UNIT - V

Managing System Implementations – Implementation Roles, Open Source Impact on Team Issues, Implementation Process, Implementations Principles, Key Documents, Migration, Interacting with Open Source Community, Support**.**

The Cost of Open Source Systems- Total Cost of Ownership, Types of Costs, Scenarios

Licensing – Types of Licenses, Licenses in Use, Mixing Open and Closed Code, Dual Licensing, Other Intellectual Property Issues

TEXTBOOK

Paul Kavanagh,"Open Source Software Implementation and Management", Elsevier Digital Press.

REFERENCE BOOKS

1. Joseph Feller and Brian Fitzgerald," Understanding Open Source Software Development", Addison Wesley Professional 2002.
2. Karl Fogel, "Producing Open Source Software", O'reilly-2006.

MCA5051 – BIGDATA ANALYTICS

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- The fundamental concepts of cloud and Apache Hadoop (Big data framework).
- The HDFS file system, MapReduce frameworks
- Hadoop tools like Hive, and Hbase.

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Apply HDFS file structure and Mapreduce frame works to solve complex problems
- Use Relational data in Hadoop environment
- Use Hadoop tools like Hive, and Hbase.

SYLLABUS

UNITI:
What is cloud computing. Cloud computing for end users / system administrators / software developers / corporate customers. Infrastructure as service (IAAS), virtualization. Platform as service (PAAS).Type-1, Type-2 PAAS.
Software as Service SAAS.
A brief history of Hadoop. Apachehadoop and the HadoopEcoSystem.Linux refresher; VmWare Installation of Hadoop.

UNIT II:
The design of HDFS. HDFS concepts. Command line interface to HDFS.
HadoopFile systems. Interfaces. Java Interface to Hadoop. Anatomy of a file read. Anatomy of a file write. Replica placement and Coherency Model. Parallel copying with distcp, Keeping an HDFS cluster balanced.

UNITIII:
Introduction. Analyzing data with unix tools. Analyzing data with hadoop. Java MapReduce classes (new API). Data flow, combiner functions, Running a distributed MapReduce Job.
Configuration API.Setting up the development environment. Managing configuration. Writing a unit test with MRUnit.Running a job in local job runner.Running on a cluster.Launching a job.The MapReduceWebUI.

UNIT IV:
Classic Mapreduce.Job submission.Job Initialization.Task Assignment. Task execution.Progress and status updates. Job Completion. Shuffle and sort on Map and reducer side. Configuration tuning.
MapReduce Types. Input formats. Output formats ,Sorting. Map side and Reduce side joins.

UNIT V:
The Hive Shell. Hive services. Hive clients. The meta store. Comparison with traditional databases.HiveQl.
Hbasics.Concepts.Implementation.Java and Mapreduce clients.Loading data, web queries.

TEXT BOOKS:

1. Tom White, Hadoop: The Definitive Guide, 3rd Edition, O'Reilly Publications, 2012
2. Landis &Blacharski , Cloud Computing made easy , 1st edition,Virtual Global, Inc,
3. Unit 2: VMWare : https://ccp.cloudera.com/display/SUPPORT/Downloads

WEB REFERENCES:

1. http://www.cloudera.com/content/cloudera-content/cloudera-docs/HadoopTutorial/CDH4/Hadoop-Tutorial.html

MCA5052 – DATABASE ADMINISTRATION

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- The basics of Database Administration.
- How to maintain a database quickly & accurately.
- Design and manage the Database Server to solve the issues related to the Database Server.
- Analyze the DBMS requirements for specific scenarios.
- Latest emerging trends in Database design and implementation of object-relational databases

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Analyse and model requirements and constraints for the purposes of Install, configure, optimize and tune the performance of a DBMS;
- Design and implement plans for security, back-up and recovery measures;
- Manage the Database storage structures;
- Create and administer user accounts;
- Monitor, troubleshoot, and maintain a database.

SYLLABUS

UNIT - I

Introduction: Database Architecture, DBMS Architecture and Data independance,DBA roles and responsibilities, Logical Database layouts, Physical Database layouts,Hardware Configurations and considerations, Overview of physical and logical storage structures.

UNIT - II

Schema Management,User Management and Database Security, Database creation, Connectivity and User Management, Creating and modifying user accounts, Creating and using Roles, Granting and revoking previliges, Managing user groups with profiles, Managing user and Security, Profiles, Managing previliges.

UNIT - III

Transaction Management, Managing multiple Databases, Managing Rollback statements, Database security and auditing, Introduction to Network administration, Network responsibilities for DBA, Network configuration, Managing large Databases, managing Distributed Databases, Configuring, Client-Server and Network computing. Oracle background processors, Overview of Oracle Net Futures.

UNIT - IV

Backup and recovery : Overview, Database backup restoration and recovery. Types of failures in oracle environment.

Defining backup and recovery strategies : Optimal backup and recovery procedures, Testing the backup and recovery plan.

UNIT - V

Introduction to performance tuning: Improving Database performance, Brief overview of Tuning methodology, An approach to oracle performance, Tuning, Optimizing, Oracle query processing, Query optimization and Oracle cost based Optimizer, The role of DBA to improve SQL processing.

TEXT BOOK

Kevin Loney , "Oracle DBA Handook", Oracle press

REFERENCE BOOKS

1. Sam R Alapati,"Expert Oracle database administration", Apress.
2. Jennick,Carol,MCAcullough Dieter,and Gerrit, Jan Linker , "Oracle DBA Bible"
3. Loney Kevin, "Oracle Database The complete reference", MCAGrahill
4. Bob Brayela,Biju Thomas, "Oracle DBA fundamentals" BPV publications

MCA5053 – SOFTWARE PROJECT MANAGEMENT

Lecture	**: 4 Periods/week**	**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits	**: 3**	**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Define and highlight importance of software project management.
- Software project management concepts, techniques and issues related to implementation.
- Describe the software project management activities
- Train software project managers and other individuals involved in software project planning and tracking
- Implementation of the software project management process

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Upon the completion, participant will be able to:
- Develop a project management plan (PMP).
- Track project execution through collecting artifacts and metrics according to procedures described in PMP.
- Revise PMP.

SYLLABUS

UNIT - I

Conventional Software Management: Waterfall model, Conventional Software Management performance.
Evolution of Software Economics: Software economics, Pragmatic software cost estimation.
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 - II

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.

UNIT - III

Model based software architectures: A Management perspective and technical perspective.
Work Flows of the process: Software process workflows, Iteration workflows
Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.

UNIT - IV

Iterative Process Planning: Work breakdown structures, Planning guidelines, Cost and Schedule estimating, Iteration planning process, Pragmatic planning
Project Organization 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 seven core Metrics, Management indicators, Quality indicators, Life cycle exceptions, Pragmatic Software Metrics, Metrics automation
Tailoring the Process: Process discriminants.
Future Software Project Management: Modern Project Profile, Next generation Software economics, Modern process transitions.

TEXT BOOK

Walker Royce, "Software Project Management", Pearson Education, 2012.

REFERENCE BOOKS

1. Walker Royce, Bob Hughes and Mike Cotterell, "Software Project Management", Tata MCAGraw-Hill Edition.
2. Joel Henry, "Software Project Management" , Pearson Education.
3. Pankaj Jalote, "Software Project Management in practice", Pearson Education 2005

MCA5054 – UNIX ADMINISTRATION

Lecture : 4 Periods/week		**Internal Marks**	**: 40**
		External Marks	**: 60**
Credits : 3		**External Examination**	**: 3 Hrs.**

Course Educational Objectives

In this course, the students will learn about

- Various key aspects of Unix administration
- Process controlling
- Users creation
- Adding new disks
- System logs

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Create new users
- Administrate various users with various permissions.
- Backup a system
- Implement system logs

SYLLABUS

UNIT - I

Booting and shutting Down: Bootstrapping, Booting Pcs, Booting in Single User mode, Startup Scripts Rebooting and Shutting down.

UNIT - II

Controlling Processes: Components of a process, Life cycle of a process, Signals, Process states.
The File System: Path names, Mounting and unmounting files, Organization of the file tree, File types, File attributes.

UNIT - III

Adding New Users: The /etc/passwd file, Adding users, Removing users, Disabling logins, Configuration of hardwired terminals, Special characters and Terminal driver, How to unwedge a terminal.

UNIT - IV

Adding a Disk: Disk Interfaces, An overview of the disk installation procedure, Periodic Processes.
Backups: Motherhood and apple pie, Backup devices and media, Restoring from dumps, Using other archiving programs.

UNIT - V

Syslog and Log Files: Logging Polices, Finding Log Files, Files not to manage, Syslog.
Drivers and the Kernel: Kernel Types, Configuring a Solaris Kernel, Linux Kernel,Adding Device Drivers, Device Files, Naming Conventions for devices.

TEXT BOOK

E. Nemeth, G. Snyder, S. Seebass and T.R.Hein, “UNIX System Administration Handbook”, Pearson Education, 3rd ED.

REFERENCE BOOKS

1. Goodheart B. Cox J, “The Magic Garden Explained”, Prentice Hall of India.
2. Leffler S.J., MCAkusick M.K., Karels M.J. and Quarterman J.S.,“The Design and Implementation of the 4.3 BSD Unix Operating System”, Addison Wesley.
3. Behrouz A.Forouzan , Richard Gilberg, “ Unix & Shell programming “, Thomson Asia, 2003

MCA551 – BUSINESS INTELLIGENCE LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Period/Week**	**External Marks**	**: 50**
Credits	**: 2**	**External Examination**	**: 3 Hrs**

Course Educational Objectives

In this course, the students will learn about

- Data Warehouse and OLAP Technology for Data Mining
- Data Preparation
- Data Mining Primitives, Languages, and System Architectures
- Concept Description: Characterization and Comparison
- Mining Association Rules in Large Databases (compressed)
- Classification and Prediction
- Cluster Analysis
- Mining Complex Types of Data
- Data Mining Applications and Trends in Data Mining

Course Outcomes

After the completion of this course, prospective engineers will have the ability to

- Define knowledge discovery and data mining
- Recognize the key areas and issues in data mining
- Apply the techniques of clustering, classification, association finding, feature selection and visualization to real world data
- Determine whether a real world problem has a data mining solution
- Apply evaluation metrics to select data mining techniques

1. Creation and Usage of ARFF files.
2. Develop Weka application to preprocess the Data.
3. Develop Weka application for attribute selection using Filters.
4. Develop Weka application to perform association Mining and categorical Data.
5. Develop Weka applications for various classification algorithms.
6. Develop Weka applications for various clustering algorithms.
7. Develop Weka application to access the data from database.
8. Develop Weka application to visualize the Data in Graphs
9. Develop a Clementine stream to access the data from database.
10. Develop a Clementine stream to access the data from various sources.
11. Develop a Clementine stream for various record options.
12. Develop a Clementine stream to visualize user input Data on Graphs.
13. Develop a Clementine stream to perform Clustering using various algorithms.
14. Develop a Clementine stream to perform Classification using various algorithms.
15. Develop a Clementine stream for various aggregations

MCA553 –MOBILE APPLICATIONS DEVELOPMENT LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Period/Week**	**External Marks**	**: 50**
Credits : 2		**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course, the students will learn about

- Mobile Applications development using Android
- App User interface designing
- Testing the mobile Apps.
- Distribute and market mobile Apps

Course Outcomes:

After the completion of this course, prospective engineers will have the ability to

- Appreciate the Mobility landscape
- Familiarize with Mobile apps development aspects
- Design and developmobile apps, using Android as development platform, with key focus on user experience design, native data handling and background tasks and notifications.
- Appreciation of nuances such as native hardware play, location awareness, graphics, and multimedia.
- Perform testing, signing, packaging and distribution of mobile apps

SYLLABUS

Students should implement (and learn to use the tools to accomplish this task) the following during Practical hours: (illustrative only)

1. Understand the app idea and design user interface/wireframes of mobile app
2. Set up the mobile app development environment
3. Develop and debug mobile app components – User interface, services, notifications, broadcast receivers, data components
4. Using emulator to deploy and run mobile apps
5. Testing mobile app - unit testing, black box testing and test automation

Infrastructure Requirements

HARDWARE / SOFTWARE REQUIREMENTS

Machine:

Pentium P4, 2.8 GHz or higher

2 GB (or higher) RAM, 40 GB (or higher) HD

Windows XP with SP2 (or higher)

REFERENCE BOOKS

1 .Anubhav Pradhan, Anil V Deshpande, 'Mobile Applications Development' Edition 1

2. Barry Burd 'Android Applications Development all in one for Dummies",e dition 1

3. "Teach Your self Android Application Development in 24 hours" Edition 1, SAMS.

MCA554 –TESTING TOOLS LAB

		Internal Marks	**: 25**
Lab/Practical	**: 4 Period/Week**	**External Marks**	**: 50**
Credits	**: 2**	**External Examination**	**: 3 Hrs**

Course Educational Objectives:

In this course, the students will learn about

- Usage of testing tools to perform functional Testing
- Usage of testing tools to perform load Testing /Stress Testing
- Usage of testing tools like Win Runner,Load runner and Test Director

Course Outcomes:

After the completion of this course, prospective engineers will have the ability to

- Use testing tools to perform functional Testing
- Use testing tools to perform load Testing /Stress Testing
- Use testing tools like Win Runner, Load Runner and Test Director

SYLLABUS

1. To learn to use the testing tools to carry out the functional testing, load/stress testing.
2. To learn to use the following (or similar) automated testing tools to automate testing:
   a) Win Runner/QTP for functional testing.
   b) Load Runner for Load/Stress testing.
   c) Test Director for test management.
3. The student should take up the case study of Unified Library application which is mentioned In the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
4. Student has to take up another case study of his/her own interest and do the same whatever Mentioned in first problem. Some of the ideas regarding case studies are given in REFERENCES Which were mentioned in theory syllabus can be referred for some idea.

REFERENCE BOOKS:

1. Software Testing Concepts and Tools, P.Nageswar Rao, dreamtech Press.
2. Software Testing Tools, Dr. K.V.K.K.. Prasad, dreamtech Press.
3. Software Testing with Visual Studio Team System 2008,S.Subashini, N.Satheesh Kumar,SPD.