

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU
Course Structure w.e.f. 2017-18 onwards
Master of Computer Applications

MCA I Year -I-Semester

S.No.	Course code	Subject	L	T	P	C
1	17FHS101	Technical Communication Skills	4	-	-	4
2	17FBS101	Probability and Statistics	4	-	-	4
3	17FHS102	Accounting and Financial Management	4	-	-	4
4	17F00101	Mathematical Foundations for Computer Science	4	-	-	4
5	17F00102	Introduction to Problem Solving and Programming	4	-	-	4
6	17FHS103	English Language Communication Skills Lab	-	-	4	2
7	17F00103	Computer Programming Lab	-	-	4	2
8	17F00104	IT Workshop	-	-	4	2
Total			20	-	12	26

MCA I Year-II-Semester

S.No.	Course code	Subject	L	T	P	C
1	17FHS201	Organization Structure and Human Resource Management	4	-	-	4
2	17F00201	Data Structures	4	-	-	4
3	17F00202	Computer Organization	4	-	-	4
4	17F00203	Operations Research	4	-	-	4
5	17F00204	Java Programming	4	-	-	4
6	17F00205	Data Structures Lab	-	-	4	2
7	17F00206	Java Programming Lab	-	-	4	2
8	17FHS202	Advanced Communication Skills Lab	-	-	4	2
Total			20	-	12	26

MCA II Year-I-Semester

S.No.	Course code	Subject	L	T	P	C
1	17F00301	Database Management Systems	4	-	-	4
2	17F00302	Computer Networks	4	-	-	4
3	17F00303	Web Technologies	4	-	-	4
4	17F00304	Software Engineering	4	-	-	4
5	17F00305	Operating Systems	4	-	-	4
6	17F00306	Web Technologies Lab	-	-	4	2
7	17F00307	Operating Systems Lab	-	-	4	2
8	17F00308	Database Management Systems Lab	-	-	4	2
Total			20	-	12	26

MCA II Year-II-Semester

S.No.	Course code	Subject	L	T	P	C
1	17F00401	Object Oriented Analysis and Design	4	-	-	4
2	17F00402	Design and Analysis of Algorithms	4	-	-	4
3	17F00403	Linux Programming	4	-	-	4
4	17F00404 17F00405 17F00406	Elective-I a. Computer Graphics and Multimedia b. Internet of Things c. Artificial Intelligence	4	-	-	4
5	17F00407 17F00408 17F00409	Elective-II a. Big data Analytics b. Scripting Languages c. Distributed Systems	4	-	-	4
6	17F00410	Object Oriented Analysis and Design Lab	-	-	4	2
7	17F00411	Design and Analysis of Algorithms Lab	-	-	4	2
8	17F00412	Linux Programming Lab	-	-	4	2
Total			20	-	12	26

MCA III Year-I-Semester

S.No.	Course code	Subject	L	T	P	C
1	17F00501	Cloud Computing	4	-	-	4
2	17F00502	Software Testing	4	-	-	4
3	17F00503	Fundamentals of Data Science	4	-	-	4
4	17F00504 17F00505 17F00506	Elective-III a. Design Patterns b. Human Computer Interaction c. Python Programming	4	-	-	4
5	17F00507 17F00508 17F00509	Elective-IV a. Software Project management b. Mobile Application Development c. Web services	4	-	-	4
6	17F00510	Map Reduce Programming Lab	-	-	4	2
7	17F00511	Software Testing Lab	-	-	4	2
8	17F00512	R & Analytics Lab	-	-	4	2
Total			20	-	12	26

MCA III Year-II-Semester

S.No	Course code	Subject	C
1	17F00601	Seminar	2
2	17F00602	Dissertation/Thesis (Very Good/Good/Satisfactory/Not-Satisfactory)	8
Total			10

COURSE OBJECTIVES

- 1 To develop awareness in students of the relevance and importance of technical communication and presentation skills.
- 2 To prepare the students for placements
- 3 To sensitize the students to the appropriate use of non-verbal communication
- 4 To train students to use language appropriately for presentations and interviews
- 5 To enhance the documentation skills of the students with emphasis on formal and informal writing

COURSE OUTCOMES

- CO1 Become effective technical communicators
- CO2 Be job-ready and able to face interviews confidently
- CO3 Sensitive use of non-verbal language suitable to different situations in professional life
- CO4 Learn and use keys words, phrases and sentence structures making a mark in interviews and presentation skills
- CO5 Effective writing skills with the ability to use different styles for different situations

UNIT 1: Basics of Technical Communication – Introduction – Objectives & Characteristics of Technical Communication – Importance and need for Technical communication - LSRW Skills – Barriers to effective communication

UNIT II

Informal and Formal Conversation - Verbal and Non-verbal communication –Kinesics, Proxemics, Chronemics, Haptics, Paralanguage

UNIT III

Written communication – Differences between spoken and written communication – Features of effective writing –Advantages and disadvantages of spoken and written communication- Art of condensation- summarizing and paraphrasing

UNIT IV

Presentation Skills – Nature and importance of oral presentation – Defining the purpose – Analyzing the audience - Planning and preparing the presentation, organizing and rehearsing the presentation – Individual and group presentations - Handling stage fright

UNIT V

Interview Skills – The Interview process –Characteristics of the job interview – Pre-interview preparation techniques – Projecting the positive image – Answering Strategies

Text Books:

1. Effective Technical Communication, Ashrif Rizvi, TataMcGrahill, 2011
2. Technical Communication by Meenakshi Raman & Sangeeta Sharma, 3rd Edition, O U Press 2015

References:

1. Communication Skills by Pushpalatha & Sanjay Kumar, Oxford University Press
2. Books on TOEFL/GRE/GMAT/CAT/ IELTS by Barron's/DELTA/Cambridge University Press.2012.
3. Soft Skills for Everyone, Butterfield Jeff, Cengage Publications, 2011.
4. Management Shapers Series by Universities Press (India) Pvt Ltd., Himayatnagar, Hyderabad 2008.
5. Successful Presentations by John Hughes & Andrew Mallett, Oxford.
6. Winning at Interviews by Edgar Thorpe and Showick Thorpe, Pearson

(17FBS101) PROBABILITY AND STATISTICS

Objectives:

- To help the students in getting a thorough understanding of the fundamentals of probability and usage of statistical techniques like testing of hypothesis, ANOVA, Statistical Quality Control, curve fitting and Queuing theory.

UNIT-I

Basic Concepts of Probability - Conditional probability – Baye's theorem. Random variables – Expectation Discrete and continuous – Distribution – Distribution functions. Binomial and Poisson distributions Normal distribution – Related properties.

UNIT-II

Test of hypothesis: Populations and samples- confidence interval of mean from normal distribution – Statistical hypothesis – Null and Alternative Hypothesis –Level of Significance –Test of significance –Test based on the normal distribution –Z-test for means and proportions: small samples –t-test for one sample and two sample problem and paired t-test, F-test and chi-square test (testing of goodness of fit and independence).

UNIT-III

Analysis of variance one way classification and two-way classification. Latin square Design and RBD.

UNIT-IV

Statistical Quality control : Concept of quality of a manufactured Defectives- Causes of variations- Random and assignable – the principle of Schwartz control charts for attribute and Variable quality characteristics- Constructions and operation of X-bar chart, R-chart , P-chart and C-chart.

UNIT-V

Curve fitting: The method of least squares- Inferences based on the least squares estimations-curve fitting regression-Multiple regression-correlation for univariate and bivariate distributions.

TEXT BOOKS:

1. Probability & Statistics for engineers by Dr.J.Ravichandran WILEY-INDIA publishers.
2. Probability & statistics by E.Rukmagadachari &E.keshava Reddy, Pearson publisher.

REFERENCES:

1. Probability & Statistics by T.K.V.Iyengar, B.Krishna Gandhi and S.Ranganatham and M.V.S.S.N.Prasad, S.Chand publications.
2. Mathematical Statistics by B.Rama Bhupal Reddy Research India Publications (DELHI), 2016).
2. Stastical methods by S.P.Gupta, S.Chand Publications.
3. Probability & Statistics for Science and Engineering by G.Shankarrao, Universities Press.
4. Probability & Statistics for Engineering and Sciences by Jay L.Devore, Cengage.
5. Probability & Statistics by R.A.Johnson and Gupta C.B.

Outcomes:

- The student will be able to analyze the problems of engineering & industry using the techniques of testing of hypothesis, ANOVA, Statistical Quality Control, curve fitting and Queuing theory and draw appropriate inferences.

(17FHS102) ACCOUNTING AND FINANCIAL MANAGEMENT

Objective: The objective of the course is to familiarize the student with the fundamentals of Accounting principles and Financial Management for making sound financial decisions.

UNIT- I: Introduction to Accounting: Definition of Accounting- Accounting concepts –Principles- Double entry system of accounting- classification of accounts- Books of accounts – Journal entries- Ledger books – preparation of financial statements and accounts-Trial Balance- Trading account-Profit and Loss account - Balance sheet(Simple problems with adjustments) .

UNIT- II: Cost Accounting and Marginal Costing: Nature- importance- Scope- difference between financial accounting and cost accounting- principles-Absorption costing- Marginal Costing - Concept of Break Even Analysis - Margin of Safety and P/V ratio- Break Even Point-Determination of BEP- Cost-Volume-Profit Analysis – managerial applications of BEP and application of marginal costing techniques (Simple problems).

UNIT- III: Financial Analysis and Interpretations: Funds flow and cash flow statements meaning-importance-statement of changes in working capital - sources and application of funds - Funds Flow and Cash flow analysis-Financial analysis through Ratios–liquidity ratios- solvency ratios – Profitability ratio, Activity ratio (Simple problems).

UNIT- IV: Financial Management: Definition-objectives- finance functions-importance-Profit and wealth maximization- Sources of capital- concept of Leverage and types of Leverage- Over Capitalization and Under Capitalization- Time Value of money -Present value of Money and Future Value of Money.

UNIT- V: Capital Budgeting and Budgeting Techniques: Definition- Features- Significance-methods of evaluation of capital budgeting proposals - Payback Period-Accounting Rate of Return (ARR)- Net Present Value Method (NPV) and Internal Rate of Return (IRR)- (Simple problems).

Learning Outcome: After completion of this course, the student will be able to understand the basic accounting principles, gets exposure to the fundamental concepts, techniques and tools of Financial Management, also enables to prepare and analyze financial statements of business enterprises for taking sound financial decisions.

TEXT BOOKS:

1. M.N.Arora, Accounting for Management, , HPH, 2012.
2. T.S.Reddy and Y.Hari Prasad Reddy, Accounting and Financial Management, Margham Publications.

REFERENCES:

1. Khan M.Y, Jain P.K, Management Accounting, 5th Edition , Tata McGraw Hill, 2012.
2. S.N.Maheshwari, Financial Accounting, 4th Edition,Vikas Publications, 2012.
3. Khan M.Y, Jain P.K, Financial Statement Analysis, PHI, 2009.
4. I.M.Pandey, Financial Management,10th Edition,Vikas Publications, 2011.
5. Financial Management, 7th Edition, TMH, 2011.

(17F00101) MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE

Course Objective

- Apply logical reasoning to solve a variety of problems.
- Understand and apply methods of discrete mathematics such as proofs, counting principles, number theory, logic and set theory to mathematical problems in a creative way.
- To apply the abstract concepts of graph theory in modelling and solving non-trivial problems in different fields of study.

Course Outcomes

- Able to apply mathematical concepts and logical reasoning to solve problems in different fields of Computer science and information technology.
- Able to apply the concepts in courses like Computer Organization, DBMS, Analysis of Algorithms, Theoretical Computer Science, Cryptography, Artificial Intelligence

Unit – I

Sets and Propositions: Introduction, Combination of Sets, Finite and Infinite Sets, Uncountably Infinite Sets, Mathematical Induction, Principle of Inclusion and Exclusion, Multisets, Propositions, Logical Connectives, Conditional and Biconditionals, Well-Formed Formulas, Tautologies, Logical Equivalences.

Relations and Functions: Introduction, Properties of Binary Relations, Closure of Relations.

Unit – II

Groups: Introduction, Groups, Subgroups, Generators and Evaluations of Powers, Cosets and Lagranges’s Theorem, Permutations Groups and Burnside’s Theorem, Codes and Group Codes, Isomorphisms and Automorphisms, Homomorphisms and Normal Subgroups.

Unit – III

Permutations, Combinations, and Discrete Probability: Introduction, the Rules of Sum and Product, Permutations, Combinations, Generation of Permutations and Combinations, Discrete Probability, Conditional Probability.

Recurrence Relations and Recursive Algorithms: Introduction, Recurrence Relations, Linear Recurrence Relations with Constant Coefficients, Homogeneous Solutions, Particular Solutions, Total Solutions.

Unit – IV

Graphs: Introduction, Basic Terminology, Multigraphs and Weighted Graphs, Digraphs and Relations, Representation of Graphs, Operations on Graphs, Paths and Circuits, Graph Traversals, Shortest Paths in Weighted Graphs, Eulerian Paths and Circuits, Hamiltonian Paths and Circuits.

Unit – V

Trees: Trees, Rooted Trees, Path Lengths in Rooted Trees, Prefix Codes, Binary Search Trees, Spanning Trees and Cut sets, Minimum Spanning Trees, Kruskal’s Algorithm, Prim’s Algorithm.

Discrete Numeric Functions: Introduction, Manipulation of Numeric Functions, Asymptotic Behavior of Numeric Functions.

Text Books:

1. C L Liu and D Mohapatra “Elements of Discrete Mathematics”, Tata Mcgraw Hill, 2009.

Reference Books:

1. Discrete and Combinatorial Mathematics, Fifth Edition, R. P. Grimaldi, B.V. Ramana, Pearson
2. Discrete Mathematics Theory and Applications, D.S Malik and M.K. Sen, Cengage Learning

3. J .L.Mott, A.Kandel, T.P .Baker, Discrete Mathematics for Computer Scientists and Mathematicians, second edition 1986, Prentice Hall of India
4. C.L.Liu, Elements of Discrete Mathematics, Second Edition 1985, McGraw-Hill Book Company. Reprinted 2000
5. Discrete Mathematics, Norman L. Biggs, Second Edition, OXFORD Indian Edition.
6. K.H.Rosen, Discrete Mathematics and applications, 5th Edition 2003, TataMcGraw Hillpublishing Company
7. Graph Theory with Applications to Engineering & Computer Science: Narsingh Deo, PHI (2004)
“Discrete Mathematical Structures” Jayant Ganguly, Sanguine.

(17F00102) INTRODUCTION TO PROBLEM SOLVING AND PROGRAMMING

Course Objectives:

- To understand the various steps in Program development.
- To understand the basic concepts in C Programming Language.
- To learn how to write modular and readable C Programs
- To understand the basic concepts such as Abstract Data Types, Linear and Non Linear Data structures.
- To understand the notations used to analyze the Performance of algorithms.
- To understand and analyze various searching and sorting algorithms.

Course Outcomes:

- Able to design the flowchart and algorithm for real world problems
- Able to learn and understand new programming languages
- Able to construct modular and readable programs
- Able to write C programs for real world problems using simple and compound data types
- Adapt programming experience and language knowledge to other programming language contexts
- Employee good programming style, standards and practices during program development

UNIT I

INTRODUCTION OF COMPUTER PROBLEM-SOLVING

Introduction – The Problem-solving Aspect – Top-down Design – Implementation of Algorithms - Program Verification – The Efficiency of Algorithms – The Analysis of Algorithms.

FUNDAMENTAL ALGORITHMS – Introduction – Exchanging the Values of Two Variables – Counting – Summation of a Set of Numbers – Factorial Computation – Sine Functional Computation – Generation of the Fibonacci Sequence – Reversing the Digits of an Integer – Base Conversion – Character to Number Conversion.

UNIT II

FACTORING METHODS

Finding the Square Root of a Number – The Smallest Divisor of an Integer – The Greatest Common Divisor of Two Integers – Generating Prime Numbers – Computing the Prime Factors of an Integer – Generation of Pseudo-random Numbers – Raising a Number to a Large Power – Computing the nth Fibonacci Number.

UNIT III

OVERVIEW OF C LANGUAGE

Features – Components – Structure – Process of Executing a ‘ C ’ Program - Data Types – Variables – Constants – Operators - Type Modifiers – Expressions – Type Definitions using typedef – Control Statements – Conditional Statements – Loops – Infinite Loops – Nested Loops – Break Statement – Continue Statement – exit() Function – goto Statement – Introduction to Arrays – One-dimensional Array – Strings – Two-dimensional Array

UNIT IV

FUNCTIONS - Introduction to Functions – Function Declaration and Prototypes – Definition – Storage Classes – Scope and Lifetime of Declaration – Passing Parameters of Functions – Command Line Arguments – Recursion in Function.

STRUCTURES – Definition – Bit Fields – Giving Values to Members – Structure Initialization – Comparison of Structures Variables – Arrays of Structures – Array within Structures – Structures within Structures – Passing Structures to Functions – Structure Pointers.

UNIONS – Definition and Declaration – Accessing a Union Member – Union of Structures – Initialization of a Union Variable – Use of Union – Use of User-defined Type Declarations.

UNIT V

POINTERS – Introduction to Pointers – Pointer Notation – Declaration and Initialization – Accessing a Variable through a Pointer – Difference between Array and Pointer – Pointer Expressions – Pointers and One-dimensional Arrays – malloc Library Function – calloc Library Function – Pointers and Multi-dimensional Arrays – Arrays of Pointers – Pointer to Pointers – Pointers and Functions – Functions with a Variable Number of Arguments.

FILE HANDLING IN 'C' – File – Defining and Opening a File – Closing a File – Input/Output Operations on Files – Functions for Random Access to Files – Example Programs.

TEXT BOOKS:

1. R. G. Dromey, How to Solve it by Computer, Pearson Education, 2007.
2. ISRD Group, Programming and Problem Solving Using C, Tata McGraw-Hill.

REFERENCE BOOKS:

1. Herbert Schildt, Osborne, C- The Complete Reference, Mcgraw Hill, Inc.
2. Brian W. Kerningham and Dennis Ritchie, C Programming Language (ANSI C), Pearson Edition.
3. B.S. Gottfried, Programming with C, Schaum Series, TMH.
4. Alfred V. Aho, Foundations of Computer Science(C Edition).

COURSE OBJECTIVES

- 1 To facilitate computer-aided multi-media instruction enabling individualized and independent language learning
- 2 To sensitise the students to the nuances of English speech sounds, word accent, intonation and rhythm
- 3 To provide opportunities for practice in using English in day to day situations
- 4 To improve the fluency in spoken English and neutralize mother tongue influence
- 5 To train students to use language appropriately for debate, group discussion and public speaking

COURSE OUTCOMES

- CO1 Better Understanding of nuances of language through audio- visual experience and be independent learners
- CO2 The significance of paralinguistic features will be understood by the students and they will try to be intelligible.
- CO3 Become good at Inter-personal skills
- CO4 Achieve neutral accent and be free from mother tongue influence
- CO5 Being an active participant in debates and group discussion, showing ability to express agreement, argument to summarize ideas to elicit the views of others and present own ideas;

UNIT- I

Phonetics – Introduction to Sounds of Speech – Vowels – Consonants – Phonetic Transcription & Orthographic Transcription

UNIT – II

Syllabification – Word Stress – Rules of word stress – Intonation – Falling tone and Rising tone

UNIT – III

Situational Dialogues – Role-play – Expressions in various situations – Self Introduction – Introducing others – Greetings – Apologies – Requests – Giving directions -Social and Professional etiquettes – Telephone Etiquettes

UNIT – IV

JAM – Describing Pictures, Photographs, Products, and Process – Talking about Wishes-Information Transfer.

UNIT – V

Debates - Group Discussions-1

MINIMUM REQUIREMENT FOR ELCS LAB:

The English Language Lab shall have two parts:

1. Computer Assisted Language Learning (CALL) Lab:
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.
2. The Communication Skills Lab with movable chairs and audio-visual aids with a P.A. system, Projector, a digital stereo-audio & video system and camcorder etc.

System Requirement (Hardware component):

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

- i) P – IV Processor
 - a) Speed – 2.8 GHZ
 - b) RAM – 512 MB Minimum
 - c) Hard Disk – 80 GB
- ii) Headphones of High quality

SUGGESTED SOFTWARE:

1. Walden Infotech English Language Communication Skills.
2. Clarity Pronunciation Power – Part I (Sky Pronunciation)
3. Clarity Pronunciation Power – part II
4. LES by British council
5. TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
6. *DELTA's key to the Next Generation TOEFL Test: Advanced Skills Practice.*
7. Lingua TOEFL CBT Insider, by Dreamtech
8. English Pronunciation in Use (Elementary, Intermediate, Advanced) CUP
9. Cambridge Advanced Learners' English Dictionary with CD.

REFERENCE BOOKS:

1. A Textbook of English Phonetics for Indian Students 2nd Ed T. Balasubramanian. (Macmillian), 2012.
2. A Course in Phonetics and Spoken English, [Dhamija Sethi](#), Prentice-Hall of India Pvt.Ltd
3. Speaking English Effectively, 2nd Edition Krishna Mohan & NP Singh, 2011. (Mcmillan).
4. A Hand book for English Laboratories, E.Suresh kumar, P.Sreehari, Foundation Books,2011
5. English Pronunciation in Use. Intermediate & Advanced, Hancock, M. 2009. CUP
6. Basics of Communication in English, Soundararaj, Francis. 2012.. *New Delhi: Macmillan*
7. Spoken English (CIEFL) in 3 volumes with 6 cassettes, OUP.
8. English Pronouncing Dictionary, Daniel Jones Current Edition with CD.Cambridge, 17th edition, 2011.

(17F00103) COMPUTER PROGRAMMING LAB

Course Objective

- To work with the compound data types
- To explore dynamic memory allocation concepts
- Able to design the flowchart and algorithm for real world problems
- Able to write C programs for real world problems using simple and compound data types
- Employee good programming style, standards and practices during program development

Course Outcomes

- Able to have fundamental concept.
- Able to write, compile and debug programs in C language.
- Able to formulate problems and implement algorithms in C.
- Able to effectively choose programming components that efficiently solve computing problems in real-world.
- Able to use different data types in a computer program.
- Able to design programs involving decision structures, loops and functions.

- Week-1**
- 1) Write a C program to make the following exchange between the variables a-> b -> c->d -> a
 - 2) Write a C program to carry out the arithmetic operations addition, subtraction, multiplication, and division between two variables
 - 3) Write a C program for printing prime numbers between 1 and n.

- Week-2**
- 1) Write a C program to construct a multiplication table for a given number.
 - 2) Write a program to reverse the digit of a given integer.
 - 3) Write a C program to find the sum of individual digits of a positive integer.
 - 4) Write a C program to calculate the factorial of a given number

- Week-3**
- 1) Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
 - 2) Write a program to calculate tax, given the following conditions:
 - a) If income is less than 1,50,000 then no tax.
 - b) If taxable income is in the range 1,50,001 – 300,000 then charge 10% tax
 - c) If taxable income is in the range 3,00,001 – 500,000 then charge 20% tax
 - d) If taxable income is above 5,00,001 then charge 30% tax

- Week-4**
- 1) Write a program to print the calendar for a month given the first Week- day of the month.

Input the first day of the month (Sun=0,Mon=1,Tue=2,Wed=3,.....) :: 3

Total number of days in the month : 31

Expected output

<i>Sun</i>	<i>Mon</i>	<i>Tue</i>	<i>Wed</i>	<i>Thu</i>	<i>Fri</i>	<i>Sat</i>
-	-	-	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
25	26	27	28	29	30	31

- 2) Write a C program to find the roots of a quadratic equation

- Week-5**
- 1) Write a program to print the Pascal triangle for a given number

- 2) Write a C program to find the GCD (greatest common divisor) of two given integers
 - 3) Write a C program to construct a pyramid of numbers.
 - 4) Write C code to define a function `cash_disperse`, which takes an amount as its input, and returns the number of 1000, 500, 100, 50, 20, 10, 5, 2, 1 rupee denomination that make up the given amount
- Week-6**
- 1) Write C code to reverse the contents of the array. For example, [1,2,3,4,5] should become [5,4,3,2,1]
 - 2) Write a C program that uses functions to perform the following:
 - i) Addition of Two Matrices
 - ii) Multiplication of Two Matrices
 - 3) Write a program that will search and find out the position where the given key element exist in a user chosen array and print it as output.
- Week-7**
- 1) Write C code to compute the frequency table of survey responses given by 20 users. The survey responses range from 1 to 5 and are stored in an array. For example, 10 responses are stored in the array [1,1,5,2,3,3,5,5,2,2]. The frequency table will be as shown below:
 - a. 1 = 2
 - b. 2 = 3
 - c. 3 = 2
 - d. 4 = 0
 - e. 5 = 3
 - 2) Write a program to define a function to sort an array of integers in ascending order by using exchange sort.
- Week-8**
- 1) Write a C program to check whether a given string is a palindrome or not, without using any built-in functions.
 - 2) Write a C program to determine if the given string is a palindrome or not by using string functions.
 - 3) Write a function that accepts a string and delete the first character.
 - 4) Write a function that accepts a string and delete all the leading spaces.
- Week-9**
- Write a program to accept a string from user and display number of vowels, consonants, digits and special characters present in each of the words of the given string.
- Week-10**
- 1) Write a C program to define a union and structure both having exactly the same numbers using the `sizeof` operators print the `sizeof` structure variables as well as union variable
 - 2) Declare a structure *time* that has three fields *hr*, *min*, *secs*. Create two variables, *start_time* and *end_time*. Input there values from the user. Then while *start_time* is not equal to *end_time* display GOOD DAY on screen.
- Week-11**
- 1) Write a program to read in an array of names and to sort them in alphabetical order. Use sort function that receives pointers to the functions `strcmp`, and `swap`, sort in turn should call these functions via the pointers.
 - 2) Write a program to read and display values of an integer array. Allocate space dynamically for the array using the `malloc()`.
 - 3) Write a program to calculate area of a triangle using function that has the input parameters as pointers as sides of the triangle.
- Week-12**
- 1) Two text files are given with the names `text1` and `text2`. These files have several lines of text. Write a program to merge (first line of `text1` followed by first line of `text2` and so on until both the files reach the end of the file) the lines of `text1` and `text2` and write the merged text to a new file `text3`.
 - 2) Write a program to split a given text file into `n` parts. Name each part as the name of the original file followed by `.part<n>` where `n` is the sequence number of the part file.

Reference Books:

1. Computer Science, A Structured Programming Approach Using C by Behrouz A. Forouzan& Richard F. Gilberg, Third Edition, Cengage Learning
2. C Programming A Problem-Solving Approach, Behrouz A. Forouzan& E.V. Prasad, F. Gilberg, Third Edition, Cengage Learning
3. Programming with C RemaTheraja, Oxford
4. “C Test Your Skills”, Kamthane, Pearson Education
5. Programming in C: A Practical Approach, Ajay Mittal, Pearson
6. Problem solving with C, M.T.Somasekhara, PHI
7. C Programming with problem solving, J.A. Jones & K. Harrow,Dreamtech Press
8. Programming withc, Byron S Gottfried, Jitender Kumar Chhabra, TMH, 2011

(17F00104) IT WORKSHOP

Course Objectives:

- To provide Technical training to the students on Productivity tools like Word processors, Spreadsheets, Presentations
- To make the students know about the internal parts of a computer, assembling a computer from the parts, preparing a computer for use by installing the operating system
- To learn about Networking of computers and use Internet facility for Browsing and Searching

Preparing your Computer

Task 1: Identify the internal parts of a computer of a computer, and its peripherals. Represent the same in the form of diagrams including Block diagram.

Task 2: Disassemble and assemble the PC back to working condition. Students should be able to trouble shoot the computer and identify working and non-working parts. Student should identify the problem correctly by various methods available. Students should record the process of assembling and trouble shooting a computer.

Task 3: Student should install Linux on the computer. Student may install another operating system (including proprietary software) and make the system dual boot or multi boot. Students should record the entire installation process.

Task 4: Students should record the various features that are supported by the operating system installed and submit it.

Networking and Internet

Task 5: Students should connect two computers directly using a cable or wireless connectivity and share information. Students should connect two or more computers using a switch/hub and share information. Crimping activity, logical configuration etc should be done by the student. The entire process has to be documented.

Task 6: Student should access the Internet for Browsing. Students should search the Internet for required information. Students should be able to create e-mail account and send email. If Intranet mailing facility is supported in the organization, then students should share the information using it. If the operating system supports sending messages to multiple users (LINUX supports it) in the same network, then it should be done by the student. Students are expected to submit the information about different browsers available, their features and search process in different languages.

Task 7: Students should download freely available Antivirus software, install it and use it to check for threats to the computer being used. Students should submit information about the features of the antivirus used, installation process, about virus definitions, virus engine etc.

Productivity tools

Task 8: Word Processor: Students should be able to create documents using the word processor tool. Some of the tasks that are to be performed are inserting and deleting the characters, words and lines, Alignment of the lines, Inserting header and Footer, changing the font, changing the colour, including images and tables in the word file, making page setup, copy and paste block of text, images, tables etc, linking the images which are present in other directory, formatting paragraphs, spell checking, etc. Students should be able to prepare project cover pages etc at the end of the task. Students should submit a user manual of the word processor considered.

Task 9: Spreadsheet: Students should be able to create, open, save the application documents and format them as per the requirement. Some of the tasks that may be practiced are Managing the worksheet environment, creating cell data, inserting and deleting cell data, format cells, adjust the cell size, applying formulas and functions, preparing charts, sorting cells. Students

should submit a user manual of the Spreadsheet application considered.

Task 10: Presentations : creating, opening, saving and running the presentations; Selecting the style for slides, formatting the slides with different fonts, colours; creating charts and tables, inserting and deleting text, graphics and animations; bulleting and numbering; hyperlinking, running the slide show, setting the timing for slide show. Students should submit a user manual of the Presentation tool considered.

References:

1. "Introduction to Computers", Peter Norton, Mc Graw Hill
2. "LaTeX Companion" – Leslie Lamport, PHI/Pearson.
3. "MOS study guide for word, Excel, Powerpoint & Outlook Exams", Joan Lambert, Joyce Cox, PHI.
4. "Introduction to Information Technology", ITL Education Solutions limited, Pearson Education.
5. "Networking your computers and devices", Rusen, PHI
6. "Trouble shooting, Maintaining & Repairing PCs", Bigelows, TMH.

(17FHS201) ORGANIZATION STRUCTURE AND HUMAN RESOURCE MANAGEMENT

Objective: The main aim of this course is to equip the student with the basic understanding about the concepts of Organizational design and Structure, Management, and the fundamental knowledge of Human Resource Management.

UNIT- I: Organization Design and Structure: Organisation- meaning- definition-Formal and Informal Organization- Organisation as a system and process- Concept of Organisational design and Structure- - types of organizational structures - Mechanistic and Organistic structures- Division of labour – Departmentation- Span of Management- Delegation of Authority- Centralisation and Decentralisation.

UNIT-II: Introduction to Management: Meaning-definition- importance- evolution of Management thought- Scientific Management-Henry Fayol’s Principles of Management
-Functions of Management- Planning- Organising-Directing- Staffing- and Controlling

UNIT-III: Human Resource Management: Meaning - definition- functions- evolution of HRM - Human Resource Planning(HRP)-Meaning-definition-Steps in HRP- Job Analysis process and methods- Employee Recruitment –Meaning- definition- Sources of Recruitment-internal and external sources- Methods of Recruitment- Factors affecting recruitment-Selection-meaning- definition-process of selection-different tests used for selection- Employee Induction -Placement.

UNIT-IV: Employee Training and Development: Meaning- importance-need- objectives and policies-principles- training methods- On-the-job and Off-the-job training methods- Career planning- Definition, succession planning, elements of career development programmes -steps in career development system-advantages and limitations.

UNIT-V: Performance Appraisal: Meaning- need- purpose- methods of performance Appraisal-essentials of a good performance appraisal- Benefits of performance appraisal system- Wage and Salary Administration-Concept-Meaning- objectives- Principles.

Learning outcome: After completion of the course, the student will be able to understand various aspects of organizational structure, fundamental concepts of management and Human Resource Management.

TEXT BOOKS:

1. James A.F. Stoner, R.Edward Freeman and Daniel R.Gilbert, jr, Management PHI-India.
2. Subbarao.P, Human Resource Management HPH.

REFERENCES:

1. Prasad.L.M , Principles & Practice of Management , 7e, S.Chand.
2. Industrial Business Management, Martand T Telsang, S.Chand.
3. Human Resources Management, Dr L.M.Prasad, S.Chand.
4. Dynamic Personnel Administration, Rudrabasavaraj MN, Himalaya.
5. Personnel Management, Mamoria & Gankar, HPH, 2009.

(17F00201) DATA STRUCTURES

Course Objective

- To develop skills to design and analyze linear and non linear data structures.
- Develop algorithms for manipulating linked lists, stacks, queues, trees and graphs.
- Develop recursive algorithms as they apply to trees and graphs.
- To get acquaintance with frequently used data structures in Software Engineering and Programming practices.
- To Strengthen the ability to identify and apply the suitable data structure for the given real world problem
- To develop a base for advanced computer science study.

UNIT I

INTRODUCTION TO DATA STRUCTURES: - Basic concepts Overview: System Life Cycle Pointers and Dynamic Memory Allocation. Pointers: Dynamic Memory Allocation, Pointers Can Be Dangerous. Algorithm Specification: Introduction, Recursive Algorithms, Data Abstraction. Performance Analysis: Space Complexity, Time Complexity, Asymptotic Notation. Practical Complexities: Performance Measurement, Clocking, Generating Test Data.

UNIT II

ARRAYS:-

The Abstract Data Type: Arrays in C, Dynamically Allocated Arrays: One-dimensional Arrays, Two-dimensional Arrays.

STACKS & QUEUES: Stacks: Stacks Using Dynamic Arrays.

Queues: Circular Queues Using Dynamic Arrays: A Mazing Problem, Evaluation of Expressions: Expressions, Evaluating Postfix Expressions, Multiple Stacks and Queues.

UNIT III

LINKED LISTS: Singly Linked Lists and Chains, Representing Chains in C Linked Stacks and Queues.

Polynomials: Polynomial Representation, Adding Polynomials, Erasing Polynomials, Circular List Representation of Polynomials.

Additional List Operations: Operations for Chains, Operations for Circularly Linked Lists, Equivalence Classes. Sparse Matrices: Sparse Matrix Representation, Sparse Matrix Input, Sparse Matrix Output, Erasing a Sparse Matrix, Doubly Linked Lists.

UNIT IV

TREES & GRAPHS: - TREES: Introduction, Binary Trees: The Abstract Data Type, Properties of Binary Trees, Binary Tree Representations. Binary Tree Traversals, Additional Binary Tree Operations, Threaded Binary Trees, Heaps, Binary Search Trees, Selection Trees, Forests.

GRAPHS: The Graph Abstract Data Type, Introduction, Elementary Graph Operations, Minimum Cost Spanning Trees, Shortest Paths and Transitive Closure.

UNIT V

SORTING & SEARCHING: Introduction, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Sorting on Several Keys, List and Table Sorts, External Sorting.

Searching: Linear Search, Binary Search, Interpolation Search, Fibonacci Search.

Text Books :

1. “ Fundamentals of Data Structures in C”, Horowitz, Sahni & Anderson-Freed - University Press.
2. “Data Structures Using C”, Reema Thareja- Oxford Higher Education.

REFERENCE BOOKS:

1. Data Structures and Algorithms Using C++ by Ananda Rao Akepogu and Radhika Raju Palagiri
2. Classic Data Structure by D. Samanta, Eastern Economy Edition.
3. Data Structures and Algorithms Made Easy by Narasimha Karumanchi, Second Edition, Written in C/C++, CareerMonk Publications, Hyderabad
4. ADTs, Data Structures and Problem Solving with C++, Larry Nyhoff, Pearson
5. Data Structures using C++, D.S.Malik, 2nd Edition, Cengage Learning
6. Data Structures through C++, Yashavant P.Kanetkar, BPB Publication
7. Data Structures using C and C++, Yedidyah Langsam.Moshe J.Augenstein Aaron M.Tenenbaum, 2nd Edition,PHI
8. Data Structures using C & C++, Rajesh K.Shukla, Wiley-India

(17F00202) COMPUTER ORGANIZATION

Course Objectives:

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(17F00203) OPERATIONS RESEARCH

Course Objectives:

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

Prerequisites:

- Probability and Statistics

UNIT I

OR Models, LP & applications, Simplex Method, M-Method, 2- Phase Method, Special cases in Simplex Method, Sensitivity Analysis

UNIT II

Transportation Model: Definition of the Transportation Model, Nontraditional Transportation Models, The Transportation Algorithm, The Assignment Model, The Transshipment Mode.

Network Model: Scope and Definition of Network Models, Minimal Spanning Tree Algorithm, Shortest-Route Problem, Maximal flow model, CPM and PERT.

UNIT III

Advanced Linear Programming: Simplex Method Fundamentals, Revised Simplex Method, Bounded-Variables Algorithm, Parametric Linear Programming.

Integer Linear Programming: Illustrative Applications, Integer Programming Algorithms - Branch-and-Bound (B&B) Algorithm, Cutting-Plane Algorithm, Computational Considerations in ILP.

UNIT IV

Heuristic Programming – Greedy Heuristic, Meta Heuristic – Tabu Search Algorithm, Simulated Annealing Algorithm, Genetic Algorithm, Application of Metaheuristics to Integer Linear Programs, Constraint Programming.

UNIT V

Travelling Salesperson Problem: Example Application Of TSP, TSP Mathematical Model, Exact TSP Algorithms.

Deterministic Dynamic Programming: Recursive Nature of Computations in DP, Forward and Backward Recursion, Selected DP Applications, Problem of Dimensionality.

TEXT BOOKS :

1. Operations Research An Introduction, By Hamdy A.Taha, Pearson 9th Edition.

REFERENCE BOOKS:

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

(17F00204) JAVA PROGRAMMING

Course Objectives:

- Study the syntax, semantics and features of Java Programming Language
- Study the Object Oriented Programming Concepts of Java Programming language
- Learn the method of creating Multi-threaded programs and handle exceptions
- Learn Java features to create GUI applications & perform event handling

Course Outcomes:

- Use object oriented approach for solving problems and implementing them
- Ability to write Efficient programs that handle exceptions
- Create user friendly interface

Unit - I :

The Java Language, The key attributes of object oriented programming language, JDK, simple program, Java keywords, identifiers in java, the java class libraries, introducing data types and operators, program control structures

Unit – II:

Introducing classes, objects, and methods, Arrays, multidimensional arrays, strings, a closer look at methods and classes, Inheritance

Unit – III :

Interface fundamentals, creating and implementing an interface, using interface references, implementing multiple interfaces, constants in interfaces, interfaces can be extended, nested interfaces, final thoughts on interface, packages, Exception handling

Unit – IV :

Byte streams and character streams, byte and character stream classes, using byte streams for reading and writing, reading and writing binary data, random access files, using character streams for file i/o, Multi threaded programming, Applet basics, a complete applet skeleton, applet initialization and termination, requesting repainting, using the status window, passing parameters to applets

Unit – V :

Swings – the origin and design philosophy of swing, components and containers, layout managers, event handling, using a push button, jtextfield, jlabel and image icon, the swing buttons, jtext field, jscrollpane, jlist, jcombobox, trees, jtable, an overview of jmenubar, jmenu and jmenuitem, creating a main menu, showmessagedialog, showconfirmdialog, showinputdialog, showoptiondialog, jdialo, create a modeless dialog

Text Books :

1. “Java Fundamentals A Comprehensive Introduction” Herbert Schildt and Dale Skrien, Mc Graw Hill.
2. “Java – How to Program”, Paul Deitel, Harvey Deitel, PHI

Reference Books :

1. “Programming with Java” T.V.Suresh Kumar, B.Eswara Reddy, P.Raghavan Pearson Edition.
2. “Core Java”, Nageswar Rao, Wiley Publishers.
3. “Thinking in Java”, Bruce Eckel, Pearson Education.
4. “Programing In java”, Malhotra, Oxford University Press
5. “Head First Java”, Kathy Sierra, Bert Bates, O’Reilly
6. “SCJP – Sun Certified Programmer for Java Study guide” – Kathy Sierra, Bert Bates, McGrawHill
7. “Java in Nutshell”, David Flanagan, O’Reilly
8. “Core Java : Volume I – Fundamentals, Cay S. Horstmann, Gary Cornell, The Sun Micro Systems Press

(17F00205) DATA STRUCTURES LAB

Course Objectives:

- To write and execute programs in C to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.
- To write and execute write programs in C to implement various sorting and searching methods
- Exemplify and implement how abstract data types such as stack, queue and linked list can be implemented to manage the memory using static and dynamic allocations
- Understand and distinguish the conceptual and applicative differences in trees, binary trees, and binary search trees
- Examine and analyze why self balancing trees are necessary in real world dynamic applications
- Develop and compare the comparison-based search algorithms and sorting algorithms

Week 1

- Write a Program to Implement Stack Operations by using Array and Linked Lists.
- Write a Program to Implement the Operations of Double Linked Lists

Week 2

- Write a C program that uses stack operations to convert a given infix expression into its postfix
- Write a Program to Implement Queue Operations by using Array and Linked Lists.

Week 3

Write a Program to Implement Circular Queue Operations by using Array and Linked Lists.

Week 4

Write a Program to Sort the set of elements by using
i). Quick Sort ii). Heap Sort. iii). Merge Sort

Week 5

Write a Program to Implement the Binary Search Tree Operations.

Week 6

Write a Program to Perform the Tree Traversal Techniques by using the Iterative Method

Week 7

Write C programs for implementing the following graph traversal algorithms:
a)Depth first traversal b)Breadth first traversal

Week 8

Write a Program to Implement All functions of a Dictionary by using Hashing

Week 9

Write a Program to Implement Skip List Operations.

Week 10

Write a Program to Implement Insertion, Deletion and Search Operations on SPLAY Trees.

Week 11

Write a program to Implement Insertion and Deletion Operations on AVL Trees

Week 12

Write a Program to Implement Insertion and Deletion Operations on B – Trees

Note: Use Classes and Objects to implement the above programs.

Reference Books:

1. Data Structures and Algorithms Using C++ by Ananda Rao Akepogu and Radhika Raju Palagiri.
2. Object Oriented Programming with ANSI & Turbo C++, Ashok N.Kamthane, Pearson Education
3. Data Structures using C++, D.S.Malik, 2nd Edition, Cengage Learning
4. Data Structures through C++, Yashavant P.Kanetkar, BPB Publication
5. Data Structures using C and C++, Yedidyah Langsam.Moshe J.Augenstein Aaron M.Tenenbaum, 2nd Edition,PHI
6. Data Structures using C & C++, Rajesh K.Shukla, Wiley-India
7. ADTs, Data Structures and Problem Solving with C++, Larry Nyhoff, Pearson

(17F00206) JAVA PROGRAMMING LAB

Course Objectives:

- To introduce java compiler and eclipse platform
- To impart hand on experience with java programming

Note:

1. **IDEs are not mandatory, encourage the use of Eclipse or Netbean platform**
2. **The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed**

Week-1:

1. Use Eclipse or Netbean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with java program to find prime numbers between 1 to n.

Week-2:

1. Write a Java program that prints all real and imaginary solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula.
2. Write a Java program for sorting a given list of names in ascending order
3. Write a java program to accept a string from user and display number of vowels, consonants, digits and special characters present in each of the words of the given text.

Week -3:

1. Write a java program to make rolling a pair of dice 10,000 times and counts the number of times doubles of are rolled for each different pair of doubles.
Hint: Math.random()
2. Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.
3. Write a java program to read the time intervals (HH:MM) and to compare system time if the system time between your time intervals print correct time and exit else try again to repute the same thing. By using StringTokenizer class

Week-4:

1. Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part<n> where n is the sequence number of the part file.
2. Write java program to create a super class called Figure that receives the dimensions of two dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and second is Triangle. Each of the sub class overridden area() so that it returns the area of a rectangle and a triangle respectively.
3. Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds

Week-5:

1. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication
2. Write a java program to find and replace pattern in given file,
3. Use inheritance to create an exception super class called EexceptionA and exception sub class ExceptionB and ExceptionC, where ExceptionB inherits from ExceptionA and ExceptionC inherits from ExceptionB. Write a java program to demonstrate that the catch

block for type ExceptionA catches exception of type ExceptionB and ExceptionC

Week-6:

1. Write a java program to convert an ArrayList to an Array.
2. Write a Java Program for waving a Flag using Applets and Threads
3. Write a Java Program for Bouncing Ball (The ball while moving down has to increase the size and decrease the size while moving up)

Week-7:

1. Write a Java Program for stack operation using Buttons and JOptionPane input and Message dialog box.
2. Write a Java Program to Addition, Division, Multiplication and subtraction using JOptionPane dialog Box and Textfields.

Week-8:

1. Write a Java Program for the blinking eyes and mouth should open while blinking.
2. Implement a Java Program to add a new ball each time the user clicks the mouse. Provided a maximum of 20 balls randomly choose a color for each ball.

Week-9:

1. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Jtable component
2. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

Week-10:

1. Write a Java Program to implement the opening of a door while opening man should present before hut and closing man should disappear.
2. Write a Java code by using JTextField to read decimal value and converting a decimal number into binary number then print the binary value in another JTextField

Week-11:

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

Week-12:

1. Write a java program establish a JDBC connection, create a table student with properties name, register number, mark1, mark2, mark3. Insert the values into the table by using the java and display the information of the students at front end.

Text Books :

1. Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI
2. Java The Complete Reference” by Herbert Schildt, TMH, 8th Edition

Reference Books :

1. Introduction to Java programming, Sixth edition, Y.Daniel Liang, Pearson Education
2. Programming in java Sachine
3. Big Java, 2nd edition, Cay Horstmann, Wiley Student Edition, Wiley India Private Limited.
4. Introduction to Programming with Java, J.Dean & R.Dean, McGraw Hill education.
5. Java Programming, D S Malik, cengage learning, India Edition

1. INTRODUCTION

The introduction of the Advanced Communication Skills Lab is considered essential at 3rd year level. At this stage, the students need to prepare themselves for their careers which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalised context.

The proposed course should be a laboratory course to enable students to use ‘good’ English and perform the following:

- Gathering ideas and information to organise ideas relevantly and coherently.
- Engaging in debates.
- Participating in group discussions.
- Facing interviews.
- Writing project/research reports/technical reports.
- Making oral presentations.
- Writing formal letters.
- Transferring information from non-verbal to verbal texts and vice-versa.
- Taking part in social and professional communication.

2. OBJECTIVES:

This Lab focuses on using multi-media instruction for language development to meet the following targets:

- To improve the students’ fluency in English, through a well-developed vocabulary and enable them to listen to English spoken at normal conversational speed by educated English speakers and respond appropriately in different socio-cultural and professional contexts.
- Further, they would be required to communicate their ideas relevantly and coherently in writing.
- To prepare all the students for their placements.

3. SYLLABUS:

The following course content to conduct the activities is prescribed for the Advanced Communication Skills (ACS) Lab:

UNIT-I: COMMUNICATIVE COMPETENCY

1. Reading Comprehension
2. Listening comprehension
3. Vocabulary for competitive purpose
4. Spotting errors

UNIT-II: TECHNICAL WRITING

1. Report writing
2. Curriculum vitae

3. E-mail writing
4. Abstract & Synopsis Writing
5. Reviewing (Book/Film)

UNIT-III: PRESENTATIONAL SKILLS

1. Oral presentation
2. Power point presentation
3. Poster presentation
4. Stage dynamics
5. Body Language

UNIT-IV: CORPORATE SKILLS

1. Telephonic skills
2. Net Etiquettes
3. SMART Goal setting
4. Time Management
5. Negotiation Skills

UNIT-V: GETTING READY FOR JOB

1. Group discussions-II
2. Interview skills
3. Answering Strategies
4. Mock Interviews

4. LEARNING OUTCOMES:

- Accomplishment of sound vocabulary and its proper use contextually
- Flair in Writing and felicity in written expression.
- Effective Speaking Abilities
- Enhanced job prospects.

5. MINIMUM REQUIREMENT:

The Advanced Communication Skills (ACS) Laboratory shall have the following infra-structural facilities to accommodate at least 60 students in the lab:

- Spacious room with appropriate acoustics.
- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- P – IV Processor, Hard Disk – 80 GB, RAM–512 MB Minimum, Speed – 2.8 GHZ
- T. V, a digital stereo & Camcorder
- Headphones of High quality

6. SUGGESTED SOFTWARE:

The software consisting of the prescribed topics elaborated above should be procured and used.

1. K-VAN SOLUTIONS-Advanced communication lab
2. DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.
3. TOEFL & GRE(KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
4. Train2success.com

7. BOOKS RECOMMENDED:

1. Objective English for Competitive Exams, Hari Mohana Prasad, 4th edition, Tata Mc Graw Hill.
2. Technical Communication by Meenakshi Raman & Sangeeta Sharma, O U Press 2009.
3. Books on TOEFL/GRE/GMAT/CAT/IELTS by Barron's/DELTA/Cambridge University Press.2012.
4. Soft Skills for Everyone, Butterfield Jeff, Cengage Publications, 2011.
5. Practice Psychometric Tests: How to familiarize yourself with genuine recruitment tests, 2012.
6. Management Shapers Series by Universities Press (India) Pvt Ltd., Himayatnagar, Hyderabad 2008.
7. Handbook for Technical Writing by David A McMurrey & Joanne Buckely CENGAGE Learning 2008.
8. English for Technical Communication for Engineering Students, Aysha Vishwamohan, Tata Mc Graw-Hill 2009.
9. Word Power Made Handy, Shalini Verma, S Chand Publications, 2011.
10. Effective Technical Communication, Ashrif Rizvi, TataMcGrahill, 2011.

(17F00301) DATABASE MANAGEMENT SYSTEMS

Objectives:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- To understand the relational database design principles.
- To become familiar with the basic issues of transaction processing and concurrency control.
- To become familiar with database storage structures and access techniques.

Outcomes:

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

UNIT-I

Introduction-Database System Applications, Purpose of Database Systems, View of Data - Data Abstraction, Instances and Schemas, Data Models, Database Languages - DDL, DML, Database Architecture, Database Users and Administrators, History of Data base Systems.

Introduction to Data base design , ER diagrams, Beyond ER Design, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises. Relational Model: Introduction to the Relational Model - Integrity Constraints over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design, Introduction to Views Destroying/ altering Tables and Views.

UNIT-II

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

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

UNIT-III

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

UNIT-IV

Transaction Management - Transaction Concept - Transaction State - Implementation of Atomicity and Durability - Concurrent - Executions - Serializability - Recoverability - Implementation of Isolation - Testing for serializability.

Concurrency Control - Lock - Based Protocols - Timestamp Based Protocols - Validation - Based Protocols - Multiple Granularity.

Recovery System-Failure Classification-Storage Structure-Recovery and Atomicity - Log - Based Recovery - Recovery with Concurrent Transactions - Buffer Management - Failure with loss of nonvolatile storage - Advance Recovery systems - Remote Backup systems.

UNIT-V

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

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

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

TEXT BOOKS:

1. Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, McGrawHill Education, 3rd Edition, 2003.
2. Data base System Concepts, A.Silberschatz, H.F. Korth, S.Sudarshan, McGraw Hill, VI edition, 2006.

REFERENCES:

1. Database Systems, 6th edition, Ramez Elmasri, Shamkat B. Navathe, Pearson Education, 2013.
2. Database Systems Concepts, Peter Rob & Carlos Coronel, Cengage Learning, 2008.
3. Introduction to Database Systems, C.J. Date, Pearson Education.
4. Database Management Systems, G.K. Gupta, McGrawHill Education.

(17F00302) COMPUTER NETWORKS

Course Objectives:

- Study the evolution of computer networks and future directions.
- Study the concepts of computer networks from layered perspective.
- Study the issues open for research in computer networks.

Course Outcomes:

- Ability to choose the transmission media depending on the requirements.
- Ability to design new protocols for computer network.
- Ability to configure a computer network logically.

Unit I

Introduction: Networks, Network Types, Internet History, Standards and Administration, Network Models: Protocol Layering, TCP/IP Protocol Suite, The ISO Model.

The Physical layer: Data and Signals, Transmission impairment, Data rate limits, Performance, Transmission media: Introduction, Guided Media, Unguided Media, Switching: Introduction, Circuit Switched Networks, Packet switching.

Unit II

The Data Link Layer: Introduction, Link layer addressing, Error detection and Correction: Cyclic codes, Checksum, Forward error correction, Data link control: DLC Services, Data link layer protocols, HDLC, Point to Point Protocol, Media Access control: Random Access, Controlled Access, Channelization, Connecting devices and virtual LANs: Connecting Devices.

Unit III

The Network Layer: Network layer design issues, Routing algorithms, Congestion control algorithms, Quality of service, Internetworking, The network layer in the Internet: IPV4 Addresses, IPV6, Internet Control protocol, OSPF, BGP, IP, ICMPv4, IGMP.

Unit IV

The Transport Layer: The Transport Service, Elements of Transport Protocols, Congestion Control, The internet transport protocols: UDP, TCP, Performance problems in computer networks, Network performance measurement.

Unit V

The Application Layer: Introduction, Client Server Programming, WWW and HTTP, FTP, e-mail, TELNET, Secure Shell, Domain Name System, SNMP.

Text Books:

1. “Data communications and networking”, Behrouz A. Forouzan, Mc Graw Hill Education, 5th edition, 2012.
2. “Computer Networks”, Andrew S. Tanenbaum, Wetherall, Pearson, 5th edition, 2010.

References:

1. Data Communication and Networks, Bhushan Trivedi, Oxford
2. “Internetworking with TCP/IP – Principles, protocols, and architecture- Volume 1, Douglas E. Comer, 5th edition, PHI
3. “Computer Networks”, 5E, Peterson, Davie, Elsevier.
4. “Introduction to Computer Networks and Cyber Security”, Chawan- Hwa Wu, Irwin, CRC Publications.

“Computer Networks and Internets with Internet Applications”, Comer.

(17F00303) WEB TECHNOLOGIES

Course Objectives:

- Learn the fundamentals of HTML and JavaScript
- Learn to communicate over a network using java
- Learn do design server side programs and access them from client side

Course Outcomes:

- Ability to design websites and do client side validations
- Share information over a network
- Ability to write server side programs

Unit I

Fundamentals: Introduction to the Web, Web servers and Clients, Resources, URL and its Anatomy, Message Format, Persistent and Non-persistent connections, Web Caching, Proxy, Java and the Net, Java Network Classes and Interfaces, Looking up Internet Address, Client/Server programs, Socket programming, e-mail client, POP3 programs, Remote method invocation, Example.

Unit II

HTML: HTML and its Flavors, HTML basics, Elements, Attributes and Tags, Basic Tags, Advanced Tags, Frames, Images, Meta tag, Planning of Web page, Model and Structure for a Website, Designing Web pages, Multimedia content.

Cascading style sheets: Advantages, Adding CSS, Browser compatibility, CSS and page layout, Selectors.

Unit III

JavaScript: Introduction, Variables, Literals, Operators, Control structure, Conditional statements, Arrays, Functions, Objects, Predefined objects, Object hierarchy, Accessing objects, Events, Event handlers, Multiple windows and Frames, Form object and Element, Advanced JavaScript and HTML, Data entry and Validation, Tables and Forms, DHTML with javascript.

Unit IV

Server side programming: Internet programming paradigm, Sever-side programming, Languages for CGI, Applications, Server environment, Environment variables, CGI building blocks, CGI scripting using C, Shell script, Writing CGI program, CGI security, Alternatives and Enhancement to CGI, Server-side Java, Advantages over Applets, Servlet alternatives, Servlet strengths, Servlet architecture, Servlet life cycle, Generic and HTTP Servelet, First servlet, Passing parameters to servlets, Retrieving parameters, Server-side include, Cookies, Fileters, Problems with servlet, Security issues, JSP and HTTP, JSP Engines, How JSP works, JSP and Servlet, Anatomy of a JSP page, JSP syntax, JSP components.

Unit – V

Sever side programming: continued: Beans, Session tracking, Users passing control and data between pages, Sharing session and Application data, Database connectivity, JDBC drivers, Basic steps, Loading a driver, Making a connection, Execute and SQL statement, SQL statements, Retrieving the result, Getting database information, Scrollable and updatable resultset, Result set metadata, Introduction to

JavaBeans, Bean builder, Advantages of Java Beans, BDK introspection, Properties, BeanInfo interface, Persistence, Customizer, JavaBeans API, EJB, Introduction to Struts Framework.

Text Books:

1. “Web Technologies”, Uttam K. Roy, , Oxford Higher Education., 1st edition, 10th impression, 2015

References

1. “Java How to program”, Paul deitel, Harvey deital, PHI
2. “Introduction to Java Programming”, Y.Daniel Liang, 6th Edition, Pearson Education, 2007
3. “The J2EE Tutorial”, Stephanie Bodoff et al, 2nd Edition, Pearson Education, 2004.
4. “Web Technologies”, Roy, Oxford University Press
5. “Web Technologies” Srinivasan, Pearson Education, 2012
6. “Java EE 5 for Beginners”, Ivan Bayross, Sharanam Shah, Cynthia Bayrossand Vaishali shai,SPD.
7. “Programming the Worldwide Web”, Robert W.Sebesta, 7th edition, 2009, Pearson Education.

(17F00304) SOFTWARE ENGINEERING

Course Objectives

- To understand the software life cycle models.
- To understand the software requirements and SRS document.
- To understand the importance of modeling and modeling languages.
- To design and develop correct and robust software products.
- To understand the quality control and how to ensure good quality software.
- To understand the planning and estimation of software projects.
- To understand the implementation issues, validation and verification procedures.
- To understand the maintenance of software

Course Outcomes

- Define and develop a software project from requirement gathering to implementation.
- Ability to code and test the software
- Ability to plan, Estimate and Maintain software systems

Unit I:

Software and Software Engineering: The Nature of Software, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths

Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology, Product and Process.

Agile Development: Agility, Agility and the Cost of Change, Agile Process, Extreme Programming, Other Agile Process Models

Unit II:

Understanding Requirements: Requirements Engineering, Establishing the groundwork, Eliciting Requirements, Developing Use Cases, Building the requirements model, Negotiating Requirements, Validating Requirements.

Requirements Modeling (Scenarios, Information and Analysis Classes): Requirements Analysis, Scenario-Based Modeling, UML Models that Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling.

Requirements Modeling (Flow, Behavior, Patterns and WEBAPPS): Requirements Modeling Strategies, Flow-Oriented Modeling, Creating a Behavioral Model, Patterns for Requirements Modeling, Requirements Modeling for WebApps.

Unit III:

Design Concepts: Design with Context of Software Engineering, The Design Process, Design Concepts, The Design Model.

Architectural Design: Software Architecture, Architecture Genres, Architecture Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Mapping Using Data Flow.

Component-Level Design: Component, Designing Class-Based Components, Conducting Component-level Design, Component Level Design for WebApps, Designing Traditional Components, Component-Based Development.

Unit IV:

User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebApp Interface Design, Design Evaluation.

WebApp Design: WebApp Design Quality, Design Goal, A Design Pyramid for WebApps, WebApp Interface Design, Aesthetic Design, Content Design, Architecture Design, Navigation Design, Component-Level Design, Object-Oriented Hypermedia Design Method(OOHMD).

Unit V:

Software Testing Strategies: A strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System Testing, The Art of Debugging.

Testing Conventional Applications: Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, basic Path testing, Control Structure Testing, Black-Box Testing, Model-based Testing, Testing for Specialized Environments, Architectures and Applications, Patterns for Software Testing. **Testing Object-Oriented Applications:** Broadening the View of Testing, Testing with OOA and OOD Models, Object-Oriented Testing Strategies, Object-Oriented Testing Methods, Testing Methods Applicable at the Class level, Interclass Test-Case Design.

Textbook:

1. “Software engineering A practitioner’s Approach”, Roger S. Pressman, McGraw Hill International Education, Seventh Edition, 2016.

Reference Textbooks:

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

(17F00305) OPERATING SYSTEMS

Course Objectives:

- To make the students understand the basic operating system concepts such as processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection.
- To get acquaintance with the class of abstractions afford by general purpose operating systems that aid the development of user applications.

Course Outcomes:

- Able to use operating systems effectively.
- Write System and application programs to exploit operating system functionality.
- Add functionality to the exiting operating systems
- Design new operating systems

UNIT I

Operating Systems Overview: Operating system functions, Operating system structure, operating systems Operations, protection and security, Computing Environments, Open- Source Operating Systems

System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

Processes: Process concept, process Scheduling, Operations on processes, Inter process Communication, Examples of IPC systems.

UNIT II

Threads: overview, Multicore Programming, Multithreading Models, Thread Libraries, Implicit Threading, Threading Issues.

Process Synchronization: The critical-section problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic problems of synchronization, Monitors, Synchronization examples, Alternative approaches.

CPU Scheduling: Scheduling-Criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Real-Time CPU Scheduling, Algorithm Evaluation.

UNIT III

Memory Management: Swapping, contiguous memory allocation, segmentation, paging, structure of the page table.

Virtual memory: demand paging, page-replacement, Allocation of frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory

Deadlocks: System Model, deadlock characterization, Methods of handling Deadlocks, Deadlock prevention, Detection and Avoidance, Recovery from deadlock.

UNIT IV

Mass-storage structure: Overview of Mass-storage structure, Disk structure, Disk attachment, Disk scheduling, Swap-space management, RAID structure, Stable-storage implementation.

File system Interface: The concept of a file, Access Methods, Directory and Disk structure, File system mounting, File sharing, Protection.

File system Implementation: File-system structure, File-system Implementation, Directory Implementation, Allocation Methods, Free-Space management.

UNIT V

I/O systems: I/O Hardware, Application I/O interface, Kernel I/O subsystem, Transforming I/O requests to Hardware operations.

Protection: Goals of Protection, Principles of Protection, Domain of protection, Access Matrix, Implementation of Access Matrix, Access control, Revocation of Access Rights, Capability- Based systems, Language – Based Protection

Security: The Security problem, Program threats, System and Network threats, Cryptography as a security tool, User authentication, Implementing security defenses, Firewalling to protect systems and networks, Computer–security classifications.

Text Books:

1. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Wiley , Eight Edition, 2014.

Reference Books:

1. Operating systems by A K Sharma, Universities Press,
2. Operating Systems, S.Haldar, A.A.Aravind, Pearson Education.
3. Modern Operating Systems, Andrew S Tanenbaum, Second Edition, PHI.
4. Operating Systems, A.S.Godbole, Second Edition, TMH.
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems, G.Nutt, N.Chaki and S.Neogy, Third Edition, Pearson Education.
7. Operating Systems, R.Elmasri, A,G.Carrick and D.Levine, Mc Graw Hill.
8. Principles of Operating Systems, B.L.Stuart, Cengage learning, India Edition.
9. Operating System Desgin, Douglas Comer, CRC Press, 2nd Edition.

(17F00306) WEB TECHNOLOGIES LAB

Course Outcomes:

- To create a fully functional website with MVC architecture. To develop an online Book store using we can sell books (Ex: amazon .com).

Course Outcomes:

- Ability to apply object oriented concepts for programming and its use.
- Practical WEB Development using java by using JDBC and ODBC connectivity.
- Implementation of servlets and PHP connectivity by using MYSQL applications.
- Learning how to use PHP in different operating systems with different editors like eclipse and net beans.
- Acquire skills to develop final project by acquired knowledge during curriculum.

Week-1:

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

1) **HOME PAGE:**

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

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

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

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

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

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart

CSE ECE EEE CIVIL	Description of the Web Site
----------------------------	-----------------------------

Fig 1.1

2) LOGIN PAGE:









This page looks like below:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Login : <input type="text"/> Password: <input type="password"/> <input type="button" value="Submit"/> <input type="button" value="Reset"/>			

3) CATALOGUE PAGE:

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

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	   	Book : XML Bible Author : Winston Publication : Wiely Book : AI Author : S.Russel Publication : Princeton hall Book : Java 2 Author : Watson Publication : BPB publications Book : HTML in 24 hours Author : Sam Peter Publication : Sam publication	\$ 40.5 \$ 63 \$ 35.5 \$ 50	   

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

Week-2:

4) CART PAGE:

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

The cart page should look like this:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Book name	Price	Quantity	Amount
	Java 2	\$35.5	2	\$70
	XML bible	\$40.5	1	\$40.5
	Total amount -			\$130.5

5) REGISTRATION PAGE:

Create a “*registration form*” with the following fields

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

WEEK 3:

VALIDATION:

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

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

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

Week-4:

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

- 1) Use different font, styles:

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

For example:

```
<HTML>
<HEAD>
<style type="text/css">
B.headline {color:red, font-size:22px, font-family:arial, text-
decoration:underline}
</style>

</HEAD>

<BODY>
<b>This is normal bold</b><br>
Selector {cursor:value}

For example:

<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
```

```

<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>

<b class="headline">This is headline style bold</b>
</BODY>

</HTML>

```

2) Set a background image for both the page and single elements on the page. You can define the background image for the page like this:

```

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

```

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

4) Define styles for links as

```

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

```

Example:

```

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

```

5) Work with layers:

For example:

LAYER 1 ON TOP:

```

<div style="position:relative, font-size:50px, z-index:2,">LAYER 1</div><div style="position:relative,
top:-50, left:5, color:red, font-size:80px, z-
index:1">LAYER 2</div>

```

LAYER 2 ON TOP:

```

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

```

6) Add a customized cursor:

```

Selector { cursor:value }

```

For example:

```

<html>
<head>
<style type="text/css">
.xlink { cursor:crosshair }
.hlink { cursor:help }
</style>
</head>

```



```

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>

```

Week-5:

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

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

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

Display the XML file as follows.

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

Use XML schemas XSL and CSS for the above purpose.

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

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

Week-6:

VISUAL BEANS:

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

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

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

Week-7:

- 1) Install IIS web server and APACHE.

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

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

Access the pages by using the urls : <http://localhost:4040/rama/books.html> (for tomcat)

<http://localhost:8080/books.html> (for Apache)

Week-8:

User Authentication :

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

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

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

Use init-parameters to do this.

Week-9:

Install a database(Mysql or Oracle).

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

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

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

Week-10:

Write a PHP which does the following job:

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

Week-11:

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

Week-12:

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

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

Modify your catalogue and cart PHP pages to achieve the above mentioned functionality using sessions.

(17F00307) OPERATING SYSTEMS LAB

Course Objectives:

- To understand the design aspects of operating system
- To solve various synchronization problems

Course out comes:

- Ensure the development of applied skills in operating systems related areas.
- Able to write software routines modules or implementing various concepts of operating system.

1. Simulate the following CPU scheduling algorithms
a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate all file allocation strategies
a) Sequential b) Indexed c) Linked
3. Simulate MVT and MFT
4. Simulate all File Organization Techniques
a) Single level directory b) Two level c) Hierarchical d) DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention
7. Simulate all page replacement algorithms
a) FIFO b) LRU c) LFU Etc. ...
8. Simulate Paging Technique of memory management
9. Control the number of ports opened by the operating system with
a) Semaphore b) monitors
10. Simulate how parent and child processes use shared memory and address space
11. Simulate sleeping barber problem
12. Simulate dining philosopher's problem
13. Simulate producer and consumer problem using threads (use java)
14. Simulate little's formula to predict next burst time of a process for SJF scheduling algorithm.
15. Develop a code to detect a cycle in wait-for graph
16. Develop a code to convert virtual address to physical address
17. Simulate how operating system allocates frame to process
18. Simulate the prediction of deadlock in operating system when all the processes announce their resource requirement in advance.

Reference Books :

1. "Operating System Concepts", Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Eighth edition, John Wiley.
2. "Operating Systems: Internals and Design Principles", Stallings, Sixth Edition--2009, Pearson Education
3. "Modern Operating Systems", Andrew S Tanenbaum, Second Edition, PHI.
4. "Operating Systems", S.Haldar, A.A.Aravind, Pearson Education.
5. "Principles of Operating Systems", B.L.Stuart, Cengage learning, India Edition.2013-2014
6. "Operating Systems", A.S.Godbole, Second Edition, TMH.
7. "An Introduction to Operating Systems", P.C.P. Bhatt, PHI.

(17F00308) DATABASE MANAGEMENT SYSTEMS LAB

Course Objectives:

- To create a database and query it using SQL, design forms and generate reports.
- Understand the significance of integrity constraints, referential integrity constraints, triggers, assertions.

Course Outcomes:

- Design databases
- Retrieve information from data bases
- Use procedures to program the data access and manipulation
- Create user interfaces and generate reports

List of Experiments:

1. Practice session: Students should be allowed to choose appropriate DBMS software, install it, configure it and start working on it. Create sample tables, execute some queries, use SQLPLUS features, use PL/SQL features like cursors on sample database. Students should be permitted to practice appropriate User interface creation tool and Report generation tool.
2. A college consists of number of employees working in different departments. In this context, create two tables **employee** and **department**. Employee consists of columns empno, empname, basic, hra, da, deductions, gross, net, date-of-birth. The calculation of hra,da are as per the rules of the college. Initially only empno, empname, basic have valid values. Other values are to be computed and updated later. Department contains deptno, deptname, and description columns. Deptno is the primary key in department table and referential integrity constraint exists between employee and department tables. Perform the following operations on the the database:

- Create tables department and employee with required constraints.
- Initially only the few columns (essential) are to be added. Add the remaining columns separately by using appropriate SQL command
- Basic column should not be null
- Add constraint that basic should not be less than 5000.
- Calculate hra,da,gross and net by using PL/SQL program.
- Whenever salary is updated and its value becomes less than 5000 a trigger has to be raised preventing the operation.
- The assertions are: hra should not be less than 10% of basic and da should not be less than 50% of basic.
- The percentage of hra and da are to be stored separately.
- When the da becomes more than 100%, a message has to be generated and with user permission da has to be merged with basic.
- Empno should be unique and has to be generated automatically.
- If the employee is going to retire in a particular month, automatically a message has to be generated.
- The default value for date-of-birth is 1 jan, 1970.
- When the employees called daily-wagers are to be added the constraint that salary should be greater than or equal to 5000 should be dropped.
- Display the information of the employees and departments with description of the fields.
- Display the average salary of all the departments.
- Display the average salary department wise.
- Display the maximum salary of each department and also all departments put together.
- Commit the changes whenever required and rollback if necessary.
- Use substitution variables to insert values repeatedly.
- Assume some of the employees have given wrong information about date-of-birth. Update the corresponding tables to change the value.
- Find the employees whose salary is between 5000 and 10000 but not exactly 7500.
- Find the employees whose name contains 'en'.
- Try to delete a particular deptno. What happens if there are employees in it and if there are no employees.
- Create alias for columns and use them in queries.
- List the employees according to ascending order of salary.
- List the employees according to ascending order of salary in each department.
- Use '&&' wherever necessary
- Amount 6000 has to be deducted as CM relief fund in a particular month which has to be accepted as input from the user. Whenever the salary becomes negative it has to be maintained as 1000 and the deduction amount for those employees is reduced appropriately.
- The retirement age is 60 years. Display the retirement day of all the employees.
- If salary of all the employees is increased by 10% every year, what is the salary of all the employees at retirement time.
- Find the employees who are born in leap year.
- Find the employees who are born on feb 29.
- Find the departments where the salary of atleast one employee is more than 20000.
- Find the departments where the salary of all the employees is less than 20000.

- On first January of every year a bonus of 10% has to be given to all the employees. The amount has to be deducted equally in the next 5 months. Write procedures for it.
- As a designer identify the views that may have to be supported and create views.
- As a designer identify the PL/SQL procedures necessary and create them using cursors.

Use appropriate Visual programming tools like oracle forms and reports, visual basic etc to create user interface screens and generate reports.

Note: As a designer identify other operations that may be required and add to the above list. The above operations are not in order. Order them appropriately. Use SQL or PL/SQL depending on the requirement.

3. Students may be divided into batches and the following experiments may be given to them to better understand the DBMS concepts. Students should gather the required information, draw ER diagrams, map them to tables, normalize, create tables, triggers, procedures, execute queries, create user interfaces, and generate reports.

- Student information system
- APSRTC reservation system
- Hostel management
- Library management
- Indian Railways reservation
- Super market management
- Postal system
- Banking system
- Courier system
- Publishing house system

References:

1. “Oracle Database 11g PL/SQL Programming”, M.Mc Laughlin, TMH.
2. “Learning Oracle SQL and PL/SQL”, Rajeeb C. Chatterjee, PHI.
3. “Introduction to SQL”, Rick F.Vander Lans, Pearson education.
4. “Oracle PL/SQL”, B.Rosenzweig and E.Silvestrova, Pearson education.

(17F00401) OBJECT ORIENTED ANALYSIS & DESIGN

Course Objectives

- To understand how to solve complex problems
- Analyze and design solutions to problems using object oriented approach
- Study the notations of Unified Modeling Language

Course Outcomes:

- Ability to find solutions to the complex problems using object oriented approach
- Represent classes, responsibilities and states using UML notation
- Identify classes and responsibilities of the problem domain

Unit-I

Introduction: The Structure of Complex systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems, Evolution of Object Model, Foundation of Object Model, Elements of Object Model, Applying the Object Model.

Unit-II

Classes and Objects: Nature of object, Relationships among objects, Nature of a Class, Relationship among Classes, Interplay of Classes and Objects, Identifying Classes and Objects, Importance of Proper Classification, Identifying Classes and Objects, Key abstractions and Mechanisms.

Unit-III

Introduction to UML: Why model, Conceptual model of UML, Architecture, Classes, Relationships, Common Mechanisms, Class diagrams, Object diagrams.

Unit-IV

Structural Modeling: Package Diagram, Composite Structure Diagram, Component Diagram, Deployment Diagram, Profile Diagram.

Unit-V

Behavioral Modeling: Use Case Diagram, Activity Diagrams, State Machine Diagrams, Sequence Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram.

Text Books:

1. "Object- Oriented Analysis And Design with Applications", Grady BOOCH, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston, PEARSON, 3rd edition, 2013.
2. "The Unified Modeling Language User Guide", Grady Booch, James Rumbaugh, Ivar Jacobson, PEARSON 12th Impression, 2012.
3. <http://www.omg.org/>

References:

1. "Object-oriented analysis and design using UML", Mahesh P. Matha, PHI
2. "Head first object-oriented analysis and design", Brett D. McLaughlin, Gary Pollice, Dave West, O'Reilly
3. "Object-oriented analysis and design with the Unified process", John W. Satzinger, Robert B. Jackson, Stephen D. Burd, Cengage Learning
4. "The Unified modeling language Reference manual", James Rumbaugh, Ivar Jacobson, Grady Booch, Addison-Wesley

(17F00402) DESIGN AND ANALYSIS OF ALGORITHMS

Course Objectives:

- To know the importance of the complexity of a given algorithm.
- To study various algorithm design techniques.
- To utilize data structures and/or algorithmic design techniques in solving new problems.
- To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.
- To study some techniques for solving hard problems.

Course Outcomes:

- Analyze the complexity of the algorithms
- Use techniques divide and conquer, greedy, dynamic programming, backtracking, branch and bound to solve the problems.
- Identify and analyze criteria and specifications appropriate to new problems, and choose the appropriate algorithmic design technique for their solution.
- Able to prove that a certain problem is NP-Complete.

UNIT I

Introduction: What is an Algorithm, Algorithm specification, Performance analysis.

Divide and Conquer: General method, Binary Search, Finding the maximum and minimum, Merge sort, Quick Sort, Selection sort, Strassen's matrix multiplication.

UNIT II

Greedy Method: General method, Knapsack problem, Job Scheduling with Deadlines, Minimum cost Spanning Trees, Optimal storage on tapes, Single-source shortest paths.

Dynamic programming: General Method, Multistage graphs, All-pairs shortest paths, Optimal binary search trees, 0/1 knapsack, The traveling sales person problem.

UNIT III

Basic Traversal and Search Techniques: Techniques for binary trees, Techniques for Graphs, Connected components and Spanning trees, Bi-connected components and DFS

Back tracking: General Method, 8 – queens problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles, Knapsack Problem.

UNIT IV

Branch and Bound: The method, Travelling salesperson, 0/1 Knapsack problem, Efficiency Considerations.

Lower Bound Theory: Comparison trees, Lower bounds through reductions – Multiplying triangular matrices, inverting a lower triangular matrix, computing the transitive closure.

UNIT V

NP – Hard and NP – Complete Problems: NP Hardness, NP Completeness, Consequences of being in P, Cook's Theorem, Reduction Source Problems, Reductions: Reductions for some known problems

Text Books:

1. “Fundamentals of Computer Algorithms”, Ellis Horowitz, S. Satraj Sahani and Rajasekhran, 2nd edition, University Press.2014,
2. “Design and Analysis of Algorithms”, Parag Himanshu Dave, Himanshu Bhalchandra Dave, Pearson Education, Second Edition, 2009.

Reference Books:

1. “Introduction to Algorithms”, second edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd./ Pearson Education.
2. “Introduction to Design and Analysis of Algorithms A strategic approach”, R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc Graw Hill.
3. “Data structures and Algorithm Analysis in C++”, Allen Weiss, Second edition, Pearson education.
4. “Design and Analysis of algorithms”, Aho, Ullman and Hopcroft,Pearson education.
5. “Algorithms” – Richard Johnson baugh and Marcus Schaefer, Pearson Education

(17F00403) LINUX PROGRAMMING

Course Objectives:

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

Course Outcomes:

- Able to describe and use the UNIX operating system.
- Able to describe and use the fundamental UNIX system tools and utilities.
- Able to describe and write shell scripts in order to perform basic shell programming.
- Able to describe and understand the UNIX file system.

Prerequisites

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

UNIT I

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

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

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

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

UNIT II

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

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

UNIT III

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

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

UNIT IV

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

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

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

UNIT V

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

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(17F00404) COMPUTER GRAPHICS AND MULTIMEDIA
Elective-I

OBJECTIVES:

- To develop, design and implement two and three dimensional graphical structures
- To enable students to acquire knowledge Multimedia compression and animations
- To learn Creation, Management and Transmission of Multimedia objects.

OUTCOMES:

- Ability to develop programs to control the content, structure and appearance of objects.
- Ability to design, organize and produce multimedia projects of all kinds

UNIT I

2D PRIMITIVES:

Elements of pictures created in computer graphics – Graphics input primitives and devices Drawing primitives in open GL and Basic open GL programming - open GL basic Graphics primitives – Output primitives – Line, Circle and Ellipse drawing algorithms – Attributes of output primitives.

UNIT II

2D GEOMETRIC TRANSFORMATIONS:

2D Viewing – Window-Viewport Transformation - Two dimensional Geometric transformations – Line, Polygon, Curve and Text clipping algorithms.

UNIT III

MULTIMEDIA BASICS

Introduction and definitions – applications – elements – Animations – Compression – Types of Compressions: Lossless – Lossy – Video compression – Image Compression – Audio compression – Data and file format standards – Multimedia data structures: KD Trees –R trees.

UNIT IV

MULTIMEDIA:

Where to use multimedia, Text: The power of meaning, About fonts and faces, Images: Before you start to create, Making still images, colour, Sound: The power of sound, Digital audio, MIDI Audio, MIDI Vs Digital audio, Multimedia system sounds, Audio File formats, Animation, Video: Using video, How video works and is displayed, Digital video containers

UNIT V

MULTIMEDIA AUTHORING AND APPLICATIONS Creating interactive multimedia – Multimedia Authoring Systems – Multimedia Authoring Software Applications – Video On demand – Virtual Reality – Augmented Reality – Content based retrieval in digital libraries.

TEXT BOOKS:

1. Donald D. Hearn, M. Pauline Baker and Warren Carithers, “Computer Graphics with OpenGL”, Fourth Edition, Pearson Education, 2010.
2. Ze-Nian Li and Mark S.Drew, “Fundamentals of Multimedia”, First Edition, Pearson Education, 2007
3. Multimedia: Making It Work”, , Tay Vaughan, 8th Edition, 2011, Tata McGrawHill Edition

REFERENCE BOOKS:

1. F.S.Hill, “Computer Graphics using OPENGL”, Second edition, Pearson Education, 2003.
2. Prabhat K Andleigh, Kiran Thakrar, “Multimedia systems design”, First Edition, PHI, 2007.

(17F00405) INTERNET OF THINGS

Elective-I

Objectives

- Makes clear view over physical computing, ubiquitous computing, or the Internet of Things, it's a hot topic in technology.
- It discusses design concepts that will make IOT products eye-catching and appealing.

Outcomes

- Ability to combine sensors, servos, robotics, Arduino chips, and more with various or the Internet, to create interactive, cutting-edge devices.
- Better idea of the overview of necessary steps to take the idea of IOT concept through production

UNIT 1

Introduction - Internet of Things – **Design Principles for Connected Devices** – Web Thinking for Connected Devices – **Internet Principles** – IP – TCP – IP Protocol Suite – UDP – IP Address – MAC Address – TCP and UDP Ports – Application Layer Protocols.

UNIT 2

Prototyping – Prototypes and Production – Cloud – Open Source vs Closed Source – Tapping into the Community – **Prototyping Embedded Devices** – Electronics – Embedded Computing Basics – Arduino – Raspberry Pi – Beagle Bone Black – Electronic Imp.

UNIT 3

Prototyping thePhysicalDesign – Laser Cutting – 3D Printing – CNC Milling – Repurposing and Recycling – **Prototyping Online Components** – New API – Real Time Reactions – Other Protocols.

UNIT 4

Techniques for writing Embedded Code – Memory Management – Performance and Battery life – Libraries – Debugging – **Business Models** – Models – Funding an Internet of Things Startup.

UNIT 5

Moving to Manufacture – Designing Kits – Designing Printed Circuit Boards – Manufacturing Printed Circuit Boards – Mass Producing the case and other Fixtures – Scaling up Software – **Ethics** – Characterizing the Internet of Things – Control – Environment – Solutions.

Text Books:

1. Adrian McEwen and HakinCassimally, “Designing The Internet of Things” Wiley Publications , 2015

Reference Books:

1. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014.
2. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013
3. CunoPfister, “Getting Started with the Internet of Things”, O’Reilly Media, 2011

(17F00406) ARTIFICIAL INTELLIGENCE
Elective-I

Course Objective:

- To learn the difference between optimal reasoning Vs human like reasoning
- To understand the notions of state space representation, exhaustive search, heuristic search along with the time and space complexities
- To learn different knowledge representation techniques
- To understand the applications of AI namely, Game Playing, Theorem Proving, Expert Systems, Machine Learning and Natural Language Processing

Learning Outcome:

- Possess the ability to formulate an efficient problem space for a problem expressed in English
- Possess the ability to select a search algorithm for a problem and characterize its time and space complexities.
- Possess the skill for representing knowledge using the appropriate technique
- Possess the ability to apply AI techniques to solve problems of Game Playing, Expert Systems and Machine Learning.

Unit – I

Foundations of AI: What is AI, History of AI, Strong and weak AI, The State of the Art.

Intelligent Agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

Unit – II

Solving Problems by Searching: Problem – Solving Agents, Example Problems, Searching for Solutions, uniformed search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.

Unit – III

Knowledge Representation: Ontological Engineering, Categories and Objects, Events, Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information, The Internet Shopping World.

Unit – IV

Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, The Theory of Learning, Regression and Classification with Learner Models, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning.

Unit – V

Learning Probabilistic Models: Statistical Learning, Learning with Complete data, Learning with Hidden variables: The EM Algorithm.

Text Books :

1. “Artificial Intelligence A Modern Approach”, Stuart J. Russell & Peter Norvig – Pearson.
2. “Artificial Intelligence”, Elaine Rich, Kevin Knight & Shivashankar B Nair – McGraw Hill Education.

(17F00407) BIG DATA ANALYTICS
Elective-II

Objectives:

- To learn to analyze the big data using intelligent techniques.
- To understand the various search methods and visualization techniques.
- To learn to various techniques for mining data stream.
- To understand the applications using Map Reduce Concepts.

Outcomes:

- On completion of this course the student will able to
- Analyze the big data analytics techniques for useful business application.
- Design efficient algorithms for mining the data from large volumes.
- Analyze the HADOOP and Map Reduce technologies associated with big data analytics.
- Explore on big data applications using Pig and Hive.

UNIT-I

Introduction to Big Data

Introduction to Big Data Platform – Challenges of Conventional System – Intelligent data analysis – Nature of Data – Analytic Processes and Tool – Analysis vs Reporting – Modern Data Analytic Tool – Statistical Concepts: Sampling Distributions – Re-Sampling – Statistical Inference – Prediction Error.

UNIT- II

Mining Data Streams

Introduction To Stream Concepts – Stream Data Model and Architecture - Stream Computing – Sampling Data in a Stream – Filtering Stream – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window – Real time Analytics Platform(RTAP) Applications – Case Studies – Real Time Sentiment Analysis, Stock Market Predictions.

UNIT – III

Hadoop

History of Hadoop- The Hadoop Distributed File System – Components of Hadoop – Analyzing the Data with Hadoop – Scaling Out – Hadoop Streaming – Design of HDFS- Java interfaces to HDFSBasics- Developing a Map Reduce Application – How Map Reduce Works – Anatomy of a Map Reduce Job run – Failures – Job Scheduling – Shuffle and Sort – Task Execution – Map Reduce Types and Formats – Map Reduce Features.

UNIT – IV

Hadoop Environment

Setting up a Hadoop Cluster – Cluster specification – Cluster Setup and Installation –Hadoop Configuration – Security in Hadoop – Administering Hadoop – HDFS – Monitoring – Maintenance – Hadoop Benchmarks – Hadoop in the Cloud.

UNIT –V

Frameworks

Applications on Big Data Using Pig and Hive – Data Processing operators in Pig – Hive Services – HiveQL – Querying Data in Hive – fundamentals of HBase and Zookeeper – IBM Info Sphere Big Insights and Streams. Visualization - Visual data analysis techniques, interaction techniques; Systems and applications.

Text Books:

1. Michael Berthold, David J.Hand, Intelligent Data Analysis, Spingers, 2007.
2. Tom White, Hadoop: The Definitive Guide Third Edition, O'reilly Media, 2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, Understanding Big Data : Analytics for Enterprise Class Hadoop and Streaming Data, McGrawHill Publishing, 2012.
4. AnandRajaraman and Jeffrey David Ullman, Mining of Massive Datasets Cambridge University Press, 2012.

Reference Books:

1. Bill Franks, Taming the big Data tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & sons, 2012.
2. Glenn J. Myatt, Making Sense of Data , John Wiley & Sons, 2007 Pete Warden, Big Data Glossary, O'Reilly, 2011.
3. Jiawei Han, MichelineKamber, Data Mining Concepts and Techniques, Second Edition.
4. Elsevier, Reprinted 2008. Da Ruan, Guoqing Chen, Etienne E.Kerre, Geert Wets, Intelligent Data Mining, Springer, 2007.
5. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Harness the Power of Big Data the IBM Big Data Platform, Tata McGraw Hill Publications, 2012.
6. Michael Minelli (Author), Michele Chambers (Author), AmbigaDhirraj (Author), Big Data, BigSnalytics.

(17F00408) SCRIPTING LANGUAGES
Elective-II

Course Objectives:

The primary objective of the course is to learn web programming by designing and developing a web based project and also learn basic User Interface Principles.

Course Outcomes:

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

- Ability to build interactive, data-driven sites and solve real time problems using PHP
- Ability to create dynamic web pages using the PHP scripting language and MySQL database.
- Understand the concepts of object-oriented programming as used in Python: classes, subclasses, inheritance, and overriding. Understand the basics of OO design.

Unit-1

Essential PHP: Enter PHP, Getting PHP, Creating your Development Environment, Creating and running first PHP page, Mixing HTML and PHP, Printing some Text, Using PHP “here” documents, command-line PHP, comments, variables, storing data, Interpolating Strings, creating variables and constants and data types.

Operators and flow controls: PHP’s Math Operators, working with the assignment operator, Increment and decrement values, String operators, Bitwise, Execution operator, Operator precedence, If statement, comparison operator, logical operators, else, elif, ternary, switch, for, while, do-while loops, foreach, terminating, skipping iterations.

Strings and Arrays: String Functions, Converting to and from Strings, Formatting Text Strings, Arrays, Array PHP Functions, Sorting Arrays, handling Multi dimensional arrays, splitting and merging arrays.

Unit-2

Creating Functions: Creating in Functions in PHP, Passing Functions some data, Passing Arrays to functions, Passing by reference, default arguments, returning data from functions, returning arrays and lists, variable scope in php, accessing global data, static variables, conditional functions, nesting functions, creating include files, returning errors from functions.

Reading data in web Pages: Setting up web pages to communicate with PHP, Handling of Text fields, Text areas, Check boxes, Radio buttons, List boxes, Password controls, Hidden controls, Image Maps, File Uploads, buttons.

PHP browser Handling Power: Using PHP’s Server Variables, HTTP Headers, User’s browser type, Redirecting browsers with HTTP Headers, Dumping a form’s data all at once, Handling form data with custom arrays, putting all it into the page, performing data validation, checking entered data, requiring numbers and text, Persisting user data, client side data validation, handling html tags in user input.

Unit-3

Object Oriented Programming: Creation of class and Objects, setting access to Properties and methods, Using constructors to Initialize Objects, using Destructors to clean up after objects, Inheritance, Method Overriding, overloading, autoloading classes .

Advanced Object Oriented Programming: Creating Static Methods, static members and

Inheritance, abstract classes, interfaces, Supporting Object Iteration, Comparing Objects, creating class constants, final key word, cloning objects.

Working with Database: what is database?, Essential sql, creating MySQL database, creating table, Accessing the database in php, Updating, Inserting, deleting, sorting data.

Sessions, Cookies, and FTP: Setting cookie and reading cookie, setting cookie expiration, deleting cookies, Working with FTP, Downloading and Uploading and deleting files with FTP, creating and removing directories with FTP, Sending E-mail, sending advanced email, adding attachments to the email, sorting data.

Unit-4

Introduction to PYTHON: Getting Started with Python Programming, Detecting and correcting Syntax errors.

Software Development, data types, and Expressions: Software Development Process, case study: Income tax calculator, Strings, Assignments, and comments, Numeric data types and character sets, Expressions, Using functions and modules, Control statements.

Design with Functions: Functions as abstraction Mechanism, Problem solving with top-down design, recursive function, File system, Managing a program's namespace, Higher Order Functions.

Unit-5

String and Text Files: Accessing characters and substrings in strings, data encryption, strings and number Systems, String Methods, Text Files, Case Study : Text Analysis.

List and Dictionaries: List, Defining simple Functions, case study: generating sentences, dictionaries, case study: nondirective psychology.

Design with classes: Getting inside objects and classes, case study: Playing the game of craps, data modelling examples, Structuring classes with Inheritance and Polymorphism.

Text Books:

1. The Complete Reference PHP by Steven Holzner, M H HILL Education, Indian Edition, 2008.
2. Fundamentals of PYTHON By Kenneth A. and Lambert and B.L Juneja, Cengage Learning, 2012.

Reference Books:

1. Core Python application and programming by Wesley J.Chun, Pearson, 3rd edition.
2. Introduction to computing and programming in python by Mark. J. Guzdial and Barbara Ericson, Pearson, 4th edition.

(17F00409) DISTRIBUTED SYSTEMS
Elective-II

Course Objectives:

The student should be made to:

- Understand the issues involved in studying process and resource management.
- Understand in detail the system level and support required for distributed system.
- Introduce the idea of peer to peer services and file system.
- Understand foundations of Distributed Systems.

Course Outcomes:

Student should be able to:

- Design process and resource management systems.
- Apply remote method invocation and objects.
- Apply network virtualization.
- Discuss trends in Distributed Systems.

UNIT I

INTRODUCTION

Examples of Distributed Systems – Trends in Distributed Systems – Focus on resource sharing – Challenges. Case study: World Wide Web.

UNIT II

COMMUNICATION IN DISTRIBUTED SYSTEM

System Model – Inter process Communication - the API for internet protocols – External data representation and Multicast communication. Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation And Objects: Remote Invocation – Introduction - Request-reply protocols - Remote procedure call - Remote method invocation. Case study: Java RMI - Group communication - Publish-subscribe systems - Message queues - Shared memory approaches - Distributed objects - Case study: Enterprise Java Beans -from objects to components.

UNIT III

PEER TO PEER SERVICES AND FILE SYSTEM

Peer-to-peer Systems – Introduction - Napster and its legacy - Peer-to-peer – Middleware - Routing overlays. Overlay case studies: Pastry, Tapestry- Distributed File Systems – Introduction - File service architecture – Andrew File system. File System: Features-File model -File accessing models - File sharing semantics Naming: Identifiers, Addresses, Name Resolution – Name Space Implementation – Name Caches – LDAP.

UNIT IV

SYNCHRONIZATION AND REPLICATION

Introduction - Clocks, events and process states - Synchronizing physical clocks- Logical

time and logical clocks - Global states – Coordination and Agreement – Introduction - Distributed mutual exclusion – Elections – Transactions and Concurrency Control– Transactions -Nested transactions – Locks – Optimistic concurrency control - Timestamp ordering – Atomic Commit protocols -Distributed deadlocks – Replication – Case study – Coda.

UNIT V

PROCESS & RESOURCE MANAGEMENT

Process Management: Process Migration: Features, Mechanism - Threads: Models, Issues, Implementation. Resource Management: Introduction- Features of Scheduling Algorithms – Task Assignment Approach – Load Balancing Approach – Load Sharing Approach.

TEXT BOOK:

1. George Coulouris, Jean Dollimore and Tim Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012.

(17F00410) OBJECT ORIENTED ANALYSIS AND DESIGN LAB

Course Objectives:

- Practice the notation for representing various UML diagrams
- Analyze and design the problem by representing using UML diagrams
- Become familiar with all phases of OOAD

Course Outcomes:

- Find solutions to the problems using object oriented approach
- Represent using UML notation and interact with the customer to refine the UML diagrams

UML diagrams to be developed are:

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

Description for an ATM System

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

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

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

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

4. A customer must be able to make a balance inquiry of any account linked to the card.
5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

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

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

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

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

List of Tasks for which students have to design all UML diagrams:

1. Banking system
2. Online bookshop system
3. University Systems
4. Library management system
5. Hospital management system
6. Result processing system

(17F00411) DESIGN AND ANALYSIS OF ALGORITHMS LAB

Course Objectives

- Implement the various algorithms that are being studied in Design and Analysis of Algorithms subject in C++/Java.

Note: You may develop programs using java or C++

1. Write a program that implements Prim's algorithm to generate minimum cost spanning tree.
2. Write a program that implements Kruskal's algorithm to generate minimum cost spanning tree.
3. Write a program to implement Huffman's algorithm for text compression.
4. Write a program to implement Dijkstra's algorithm for Single source shortest path problem.
5. Write a program to implement Floyd's algorithm for the All pairs shortest path problem.
6. Write a program to implement greedy algorithm for job sequencing with deadlines.
7. Write programs for the implementation of bfs and dfs for a given graph.
8. Write a program to find Minimum Cost Binary Search Tree.
9. Write a program to implement Dynamic Programming algorithm for 0/1 Knapsack problem.
10. Write a program to implement the Backtracking algorithm for the sum of subsets problem.
11. Write programs to implement backtracking algorithms for
 - a) N-queens problem
 - b) The Hamiltonian cycles problem
 - c) The m-colourings graph problem

TEXT BOOKS

1. Data structures and Algorithms in java, 3rd edition, A. Drozdek, Cengage Learning.
2. Data structures with Java, J.R. Hubbard, 2nd edition, Schaum's Outlines, TMH.
3. Data structures and algorithms in Java, 2nd Edition, R. Lafore, Pearson Education.
4. Data Structures using Java, D.S. Malik and P.S. Nair, Cengage Learning.
5. Data structures, Algorithms and Applications in java, 2nd Edition, S. Sahani, Universities Press.
6. Data structures, Algorithms and Applications in C++, 2nd Edition, S. Sahani, Universities Press.
7. Data structures and Algorithm Analysis in C++, 2nd Edition, M.A. Weiss, Pearson education.
8. Design and Analysis of Algorithms, P.H. Dave and H.B. Dave, Pearson education.
9. Data structures and java collections frame work, W.J. Collins, Mc Graw Hill.

(17F00412) LINUX PROGRAMMING LAB.

Course Objectives:

- To implement some standard Unix utilities using system calls.
- To develop shell scripts to solve problems.
- To produce programs in C for network-based applications.
- To implement CPU scheduling algorithms, file allocation methods and page replacement algorithms in C.

Course Outcomes:

After completion of the course students will be able to

- Work confidently in Unix/Linux environment
- Write shell scripts to automate various tasks
- Master the basics of linux administration

Note: Use Bash for Shell scripts.

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a C program that makes a copy of a file using standard I/O and system calls.
11. Implement in C the following Unix commands using System calls
a). cat b) mv
12. Write a C program to list files in a directory.
13. Write a C program to emulate the Unix ls -l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C program that redirects standard output to a file.Ex: ls > f1.
16. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
17. Write a C program to create a Zombie process.

18. Write a C program that illustrates how an orphan is created.
19. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls -l | sort
20. Write C programs that illustrate communication between two unrelated processes using named pipe.
21. Write a C program in which a parent writes a message to a pipe and the child reads the message.
22. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
23. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (22)) and displays them.
24. Write a C programs to transfer a large amount of data between processes,using
 - a) a pipe
 - b)a FIFO
 - c)a message queue.
25. Write a C program to allow cooperating processes to lock a resource for exclusive use, using:
 - a)Semaphores
 - b)flock or lockf system calls.
26. Write a C program that illustrates suspending and resuming processes using signals.
27. Write a C program that implements a producer-consumer system with two processes. (using Semaphores).
28. Write client and server programs(using c) for interaction between server and client processes using Unix Domain sockets.
29. Write client and server programs(using c) for interaction between server and client processes using Internet Domain sockets.
30. Write C programs that illustrate two processes communicating using shared memory.

TEXT BOOKS:

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

(17F00501) CLOUD COMPUTING

Objectives

- To introduce the basis of Cloud Computing
- To educate the cloud working function
- To allow computer system resources to be used in an efficient manner
- Makes the environment to the cloud.

Outcomes

On successful completion of the course, students will be able to:

- Understand the concepts of cloud computing and its related techniques.
- Provide a pleasant and effective user interface.

UNIT-I

Introduction to cloud computing – The Evolution of cloud computing – Hardware Evolution- Internet Software Evolution – Server Virtualization – Web Services Deliver from the cloud– Communication-as-a-service–Infrastructure-as-a-service–Monitoring-as-a-service–Platform- as-a-Service - Software-as-a-service – Building Cloud Network.

UNIT-II

Federation in the cloud – presence in the cloud – Privacy and its Relation to cloud-Based Information Systems– Security in the cloud – Common Standards in the cloud-End-User Access to the cloud Computing.

UNIT-III

Introduction – Advancing towards a Utility Model – Evolving IT infrastructure – Evolving Software Applications – Continuum of Utilities- Standards and Working Groups- Standards Bodies and Working Groups- Service Oriented Architecture- Business Process Execution Language- Interoperability Standards for Data Center Management – Utility Computing Technology- Virtualization – Hyper Threading – Blade Servers- Automated Provisioning- Policy Based Automation- Application Management – Evaluating Utility Management Technology – Virtual Test and development Environment – Data Center Challenges and Solutions – Automating the Data Center.

UNIT-IV

Software Utility Application Architecture – Characteristics of a SaaS – Software Utility Applications – Cost Versus Value – Software Application Services Framework – Common Enablers – Conceptual view to Reality – Business profits – Implementing Database System for Multitenant Architecture.

UNIT-V

Other Design Consideration – Design of a Web Services Metering Interface – Application Monitoring Implementation – A Design for an update and Notification Policy – Transforming to Software as a Service – Application Transformation Program – Business Model Scenarios – Virtual Services for Organizations – The Future.

Text Books:

1. Guy Bunker and Darren Thomson, Delivering utility Computing, John Wiley & Sons Ltd, 2012.

References Books:

1. John W. Rittinghouse and Ames F. Ransome, Cloud Computing Implementation , Management and security, CRC press & Francis Group, Boca Raton London New York. 2010.
2. Alfredo Mendroza, Utility Computing Technologies, Standards, and Strategies Artech House INC, 2007.

(17F00502) FUNDAMENTALS OF DATA SCIENCE

Objectives

The course gives you a set of practical skills for handling data that comes in a variety of formats and sizes, such as texts, spatial and time series data. These skills cover the data analysis lifecycle from initial access and acquisition, modeling, transformation, integration, querying, application of statistical learning and data mining methods, and presentation of results. This includes data wrangling, the process of converting raw data into a more useful form that can be subsequently analysed.

Outcomes

- Understand business intelligence and business and data analytics.
- To understand the business data analysis through the powerful tools of data application.
- Understand the methods of data mining.
- Apply basic tools (plots, graphs, summary statistics) to carry out EDA.
- Understand the key elements of a data science project
- Identify the appropriate data science technique and/or algorithm to use for the major data science tasks.

UNIT - I

Introduction, What Is Statistical Learning?, Why Estimate f ?, How Do We Estimate f ?, The Trade-Off Between Prediction Accuracy and Model Interpretability, Supervised Versus Unsupervised Learning, Regression Versus Classification Problems, Assessing Model Accuracy, Measuring the Quality of Fit, The Bias-Variance Trade-of, The Classification Setting, Introduction to R, Basic Commands, Graphics, Indexing Data, Loading Data, Additional Graphical and Numerical Summaries.

UNIT – II

Linear Regression, Simple Linear Regression, Multiple Linear Regression, Other Considerations in the Regression Model, Comparison of Linear Regression with K-Nearest Neighbours, Linear Regression.

UNIT-III

Classification, Logistic Regression, Linear Discriminant Analysis, A Comparison of Classification Methods, Logistic Regression, LDA, QDA, and KNN.

UNIT- IV

Programming for basic computational methods such as Eigen values and Eigen vectors, sparse matrices, QR and SVD, Interpolation by divided differences.

Data Wrangling: Data Acquisition, Data Formats, Imputation, The split-apply-combine paradigm.

UNIT-V

Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

Data Warehouse: Basic Concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization by Attribute-Oriented Induction.

Text Books:

1. Gareth James Daniela Witten Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, February 11, 2013, web link: www.statlearning.com.
2. Mark Gardener, Beginning R The statistical Programming Language, Wiley, 2015.
3. Han , Kamber, and J Pei, Data Mining Concepts and Techniques, 3rd edition, Morgan Kaufman, 2012.

References:

1. Sinan Ozdemir, Principles of Data Science, Packt Publishing Ltd Dec 2016.
2. Joel Grus, Data Science from Scratch, Oreilly media, 2015.

(17F00503) SOFTWARE TESTING

Course Objectives:

- Fundamentals for various testing methodologies.
- Describe the principles and procedures for designing test cases.
- Provide supports to debugging methods.
- Acts as the reference for software testing techniques and strategies.

Course Outcomes:

- Understand the basic testing procedures.
- Able to support in generating test cases and test suites.
- Able to test the applications manually by applying different testing methods and automation tools.
- Apply tools to resolve the problems in Real time environment.

UNIT I

Introduction: Purpose of Testing, Dichotomies, Model for Testing, Consequences of Bugs, Taxonomy of Bugs.

Flow graphs and Path testing: Basics Concepts of Path Testing, Predicates, Path Predicates and Achievable Paths, Path Sensitizing, Path Instrumentation, Application of Path Testing.

UNIT II

Transaction Flow Testing: Transaction Flows, Transaction Flow Testing Techniques.

Dataflow testing: Basics of Dataflow Testing, Strategies in Dataflow Testing, Application of Dataflow Testing.

UNIT III

Domain Testing: Domains and Paths, Nice & Ugly Domains, Domain testing, Domains and Interfaces Testing, Domain and Interface Testing, Domains and Testability.

UNIT IV

Paths, Path products and Regular expressions: Path Products & Path Expression, Reduction Procedure, Applications, Regular Expressions & Flow Anomaly Detection.

Logic Based Testing: Overview, Decision Tables, Path Expressions, KV Charts, Specifications.

UNIT V:

State, State Graphs and Transition Testing: State Graphs, Good & Bad State Graphs, State Testing, Testability Tips.

Graph Matrices and Application: Motivational Overview, Matrix of Graph, Relations, Power of a Matrix, Node Reduction Algorithm, Building Tools.

Text Books:

1. Software testing techniques – Boris Beizer, Dreamtech, second edition.

Reference Books :

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing- Yogesh Singh, Camebridge
3. Software Testing, 3rd edition, P.C. Jorgensen, Aurbach Publications (Dist.by SPD).
4. Software Testing, N.Chauhan, Oxford University Press.
5. Introduction to Software Testing, P.Ammann & J.Offutt, Cambridge Univ. Press.
6. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.
7. Software Testing Concepts and Tools, P.Nageswara Rao, dreamtech Press
8. Win Runner in simple steps by Hakeem Shittu,2007 Genixpress.
9. Foundations of Software Testing, D.Graham & Others, Cengage Learning.

(17F00504) DESIGN PATTERNS

Elective-III

Course Objectives:

- To understand design patterns and their underlying object oriented concepts.
- To understand implementation of design patterns and providing solutions to real world software design problems.
- To understand patterns with each other and understanding the consequences of combining patterns on the overall quality of a system.

Course Outcomes:

- Know the underlying object oriented principles of design patterns.
- Understand the context in which the pattern can be applied.
- Understand how the application of a pattern affects the system quality and its tradeoffs.

UNIT-I

Introduction to Design Patterns

Design Pattern Definition, Design Patterns in Small Talk MVC, Describing Design Patterns, Catalog of Design Patterns, Organizing the Catalog, Solving of Design Problems using Design Patterns, Selection of a Design Pattern, Use of Design Patterns.

UNIT-II

Designing A Document Editor: A Case Study

Design problems, Document structure, Formatting, Embellishing the User Interface, Supporting Multiple Look and Feel standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation.

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

UNIT-III

Structural Patterns-1: Adapter, Bridge, Composite.

Structural Patterns-2: Decorator, Façade, Flyweight, Proxy, Discuss of Structural Patterns.

UNIT-IV

Behavioral Patterns-1: Chain of Responsibility, Command, Interpreter, Iterator.

Behavioral Patterns-2: Mediator, Memento, Observer.

UNIT-V

Behavioral Patterns-2(cont'd): State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns.

What to Expect from Design Patterns, A Brief History, The Pattern Community An Invitation, A Parting Thought.

TEXT BOOK :

1. Design Patterns By Erich Gamma, Pearson Education

REFERENCE BOOKS:

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

(17F00505) HUMAN COMPUTER INTERACTION
Elective-III

Course Objectives:

- Gain an overview of Human-Computer Interaction (HCI), with an understanding of user interface design in general, and alternatives to traditional "keyboard and mouse" computing
- Become familiar with the vocabulary associated with sensory and cognitive systems as relevant to task performance by humans
- Be able to apply models from cognitive psychology to predicting user performance in various human-computer interaction tasks and recognize the limits of human performance as they apply to computer operation
- Be familiar with a variety of both conventional and non-traditional user interface paradigms

Course Outcomes:

- Find innovative ways of interacting with computers
- Help the disabled by designing non-traditional ways of interacting
- Use cognitive psychology in the design of devices for interaction

UNIT- I

Introduction: Importance of user Interface – definition, importance of good design, Benefits of good design, A brief history of Screen design.

The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics

Web user Interface - popularity, characteristics- Principles of user interface.

UNIT- II

Design process – Understanding how people interact with computers, importance of human characteristics human consideration, Human interaction speeds, understanding business functions.

Screen Designing: Design goals – Screen meaning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

UNIT- III

System menus: Structures of Menus, Functions of Menus, Content of Menus, Kinds of Graphical menus
Windows: Window characteristics, Components of a window, Window presentation styles, Types of windows, Window management

UNIT- IV

Controls: Characteristics of device based controls, Selecting the proper device based controls, Operable controls, Text Entry/Read-only controls, Selection controls, Combination Entry/selection controls, Selecting the proper controls.

UNIT- V

Graphics: Icons, Multimedia, Color-what is it, Color uses, Color and Human vision, Choosing colors

Testing: The purpose and importance of usability testing, Scope of testing, Prototypes, Kinds of Tests, Developing and conducting the test.

Text books:

1. “The essential guide to user interface design”, Wilbert O Galitz., Wiley, 2nd edition, 2013.

References:

1. “Designing the user interface”, 3rd Edition Ben Shneidermann, Pearson Education Asia.
2. “Human –Computer Interaction”, D.R.Olsen, Cengage Learning.
3. “Human – Computer Interaction”, I.Scott Mackenzie, Elsevier Publishers.
4. “Interaction Design”, Prece, Rogers, Sharps, Wiley Dreamtech.
5. “User Interface Design”, Soren Lauesen , Pearson Education.
6. “Human –Computer Interaction”, Smith - Atakan, Cengage Learning.

(17F00506) PYTHON PROGRAMMING
Elective-III

OBJECTIVES:

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to identify Python object types.

OUTCOMES:

- Making Software easily right out of the box.
- Experience with an interpreted Language.
- To build software for real needs.
- Prior Introduction to testing software

UNIT – I:

Introduction: History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

UNIT – II:

Types, Operators and Expressions: Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations Control Flow- if, if-elif-else, for, while, break, continue, pass.

UNIT – III:

Data Structures Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences. Comprehensions.

UNIT – IV:

Functions - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions(Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

Modules: Creating modules, import statement, from. Import statement, name spacing,

Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages

UNIT – V:

Object Oriented Programming OOP in Python: Classes, 'self variable', Methods, Constructor Method, Inheritance, Overriding Methods, Datahiding,

Error and Exceptions: Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions

Brief Tour of the Standard Library - Operating System Interface - String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multithreading, GUI Programming, Turtle Graphics

Testing: Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

TEXT BOOKS

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

Reference Books:

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage

(17F00507) SOFTWARE PROJECT MANAGEMENT
Elective-IV

Course Objectives:

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

- Understanding the specific roles within a software organization as related to project and process management
- Describe the principles, techniques, methods & tools for model-based management of software projects, assurance of product quality and process adherence (quality assurance), as well as experience-based creation & improvement of models (process management).
- Understanding the basic infrastructure competences (e.g., process modeling and measurement)
- Understanding the basic steps of project planning, project management, quality assurance, and process management and their relationships

Course Outcomes:

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

UNIT I

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

UNIT II

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

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

UNIT III

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

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

UNIT IV

Work Flows of the process: Software process workflows, Inter Trans workflows. Checkpoints of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments. Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Interaction planning process, Pragmatic planning.

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

Process Automation: Automation Building Blocks, The Project Environment

UNIT V

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

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

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

Text Books:

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

Reference Books :

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

(17F00508) MOBILE APPLICATION DEVELOPMENT
Elective-IV

Objectives:

- To understand fundamentals of android operating systems.
- Illustrate the various components, layouts and views in creating android applications
- To understand fundamentals of android programming.

Out comes:

- Create data sharing with different applications and sending and intercepting SMS.
- Develop applications using services and publishing android applications.
- To demonstrate their skills of using Android software development tools

Unit 1:

Basics of Mobile Applications Development:

Tools: Eclipse ADT, Android Studio.

Understanding the Role of Android Application Components, Understanding the Utility of Android API, Overview of the Android Project Files, Understanding Activities, Role of the Android Manifest File, Creating the User Interface, Commonly Used Layouts and Controls, Event Handling, Displaying Messages Through Toast, Creating and Starting an Activity, Using the Edit Text Control, Choosing Options with Checkbox, Choosing Mutually Exclusive Items Using Radio Buttons

Unit 2: Building Blocks for Android Application Design:

Introduction to Layouts, Linear Layout, Relative Layout, Absolute Layout, Using Image View, Frame Layout, Table Layout, Grid Layout, Adapting to Screen orientation.

Utilizing Resources and Media Resources, Creating Values Resources, Using Drawable Resources, Switching States with Toggle Buttons, Creating an Images Switcher Application, Scrolling Through Scroll View, playing Audio, Playing Video, Displaying Progress with Progress Bar, Using Assets

Unit 3: Using Selection widgets and Debugging:

Using List View, Using the Spinner control, Using the GridView Control, Creating an Image Gallery Using the ViewPager Control, Using the Debugging Tool: Dalvik Debug Monitor Service(DDMS), Debugging Application, Using the Debug Perspective.

Displaying And Fetching Information Using Dialogs and Fragments: What Are Dialogs?, Selecting the Date and Time in One Application, Fragments, Creating Fragments with java Code, Creating Special Fragments

Unit 4: Building Menus:Creating Interface Menus and Action Bars, Menus and Their Types, Creating Menus Through XML, Creating Menus Through Coding, Applying a Context Menu to a List View, Using the Action Bar, Replacing a Menu with the Action Bar, Creating a Tabbed Action Bar, Creating a Drop-Down List Action Bar

Unit 5 Storing Data & Communicating with SMS and Emails:

Using the SQLiteOpenHelperclasss, Accessing Databases with the ADB, Creating a Data Entry Form. Understanding Broadcast Receivers, Using the Notification System, Sending SMS Messages with Java Code, Receiving SMS Messages, Sending Email, Working With Telephony Manager.

Text Books

1. Android Programming by B.M Harwani, Pearson Education, 2013.

References Text Books:

1. Android application Development for Java Programmers, James C Sheusi, Cengage Learning
2. Android In Action by w.Frank Ableson, Robi Sen, Chris King, C. Enrique Ortiz., Dreamtech.
3. Professional Android 4 applications development, Reto Meier, Wiley India, 2012.
4. Beginning Android 4 applications development, Wei- Meng Lee, Wiley India,2013
5. PawPrints Learning Technologies, Beginning Android Development: Create Your Own Android Apps Today, 2014.
6. Erik Hellman, Android Programming: Pushing the Limits, John Wiley and sons ltd, 2014.
7. Neil Smyth, Android Studio Development Essentials.
8. Joseph Anuzzi,Jr, Lauren Darcey, Introduction to Android Application Development, Addison-Wesley, Fourth Edition.

(17F00509) DATA WAREHOUSING & MINING

Elective-IV

Course Objectives:

- To know the basic concepts and principles of data warehousing and data mining
- Learn pre-processing techniques and data mining functionalities
- Learn and create multidimensional models for data warehousing
- Study and evaluate performance of Frequent Item sets and Association Rules
- Understand and Compare different types of classification and clustering algorithms

Course Outcomes:

- Understand the basic concepts of data warehouse and data Mining
- Apply pre-processing techniques for data cleansing
- Analyze and evaluate performance of algorithms for Association Rules
- Analyze Classification and Clustering algorithms

UNIT I

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

UNIT II

Data Warehouse and OLAP Technology for Data Mining: Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Further Development of Data Cube Technology, From Data Warehousing to Data Mining. Data Cube Computation and Data Generalization: Efficient Methods for Data Cube Computation, Further Development of Data Cube and OLAP Technology, Attribute-Oriented Induction.

UNIT III

Mining Frequent Patterns, Associations and Correlations: Basic Concepts, Efficient and Scalable Frequent Itemset Mining Methods, Mining various kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining, Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Back propagation, Support Vector Machines, Associative Classification, Lazy Learners, Other Classification Methods, Prediction, Accuracy and Error measures, Evaluating the accuracy of a Classifier or a Predictor, Ensemble Methods

UNIT IV

Cluster Analysis Introduction :Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Clustering High-Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis.

UNIT V

Mining Streams, Time Series and Sequence Data: Mining Data Streams, Mining Time-Series Data, Mining Sequence Patterns in Transactional Databases, Mining Sequence Patterns in Biological Data, Graph Mining, Social Network Analysis and Multi relational Data Mining, Mining Object, Spatial, Multimedia, Text and Web Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Spatial Data Mining, Multimedia Data Mining, Text Mining, Mining the World Wide Web.

TEXT BOOKS:

1. Data Mining: Concepts and Techniques, Jiawei Han and Micheline Kamber, Morgan Kaufmann Publishers, Elsevier, Second Edition, 2006.
2. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education.

REFERENCES:

1. Data Mining Techniques, Arun K Pujari, Second Edition, Universities Press.
2. Data Warehousing in the Real World, Sam Aanhory & Dennis Murray Pearson EdnAsia.
3. Insight into Data Mining, K.P.Soman, S.Diwakar, V.Ajay, PHI, 2008.

(17F00510) CLOUD COMPUTING LAB

Course Objectives:

- The student should be made to:
- Be exposed to tool kits for cloud environment.
- Learn to run virtual machines of different configuration.

Course Outcomes:

The student should be able to
Design and Implement applications on the Cloud.
Use the cloud tool kits.

Programs on SaaS

- 1 Create an word document of your class time table and store locally and on the cloud with doc,and pdf format . (use www.zoho.com and docs.google.com)
- 2 Create a spread sheet which contains employee salary information and calculate gross and total sal using the formula
DA=10% OF BASIC
HRA=30% OF BASIC
PF=10% OF BASIC IF BASIC<=3000
12% OF BASIC IF BASIC>3000
TAX=10% OF BASIC IF BASIC<=1500
=11% OF BASIC IF BASIC>1500 AND BASIC<=2500
=12% OF BASIC IF BASIC>2500
(use www.zoho.com and docs.google.com)
NET_SALARY=BASIC_SALARY+DA+HRA-PF-TAX
- 3 Prepare a ppt on cloud computing –introduction , models, services ,and architecture
Ppt should contain explanations, images and at least 20 pages
(use www.zoho.com and docs.google.com)
- 4 Create your resume in a neat format using google and zoho cloud

Programs on PaaS

- 1 Write a Google app engine program to generate n even numbers and deploy it to google cloud
- 2 Google app engine program multiply two matrices
- 3 Google app engine program to validate user ; create a database login(username, password) in mysql and deploy to cloud
- 4 Write a Google app engine program to display nth largest no from the given list
of numbers and deploy it into google cloud
- 5 Google app engine program to validate the user
Use mysql to store user info and deploy on to the cloud
- 6 Implement Prog 1-5 using Microsoft Azure

CASE STUDY- cloud computing

Sr. No.	Title of Experiment	Aim of the Experiment	Demonstration Equipments/ Components to be required	Type of Experiment/ Demonstration (Lab/Classroom)
1	Case Study of Amazon	To understand the services of Amazon elastic cloud.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
2	Case Study of Azure	To understand the services of Microsoft azure.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
3	Case Study of Hadoop	To understand the services of hadoop.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
4	Case Study of Aneka	To understand the services of aneka elastic cloud.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
5	Case Study of Google Apps	To understand the services of google apps engine.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
6	Google apps business solution for data access and data upload	To understand the business solution application of Google apps.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
7	Control panel software manager Application of hypervisors	To understand the application of hypervisors.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.

(17F00511) DATA SCIENCE LAB

Objectives:

Data Science is the study of the generalizable extraction of knowledge from data. Being a data scientist requires an integrated skill set spanning mathematics, statistics, machine learning, databases and other branches of computer science along with a good understanding of the craft of problem formulation to engineer effective solutions. This course will introduce students to this rapidly growing field and equip them with some of its basic principles and tools as well as its general mindset. Students will learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration, exploratory data analysis, predictive modeling, descriptive modeling, data product creation, evaluation, and effective communication.

Outcomes:

- Demonstrate mastery of a body of knowledge that includes recent developments in computer science and information technology;
- Understand and use appropriate and relevant, fundamental and applied mathematical and statistical knowledge, methodologies and modern computational tools;
- Recognise and use research principles and methods applicable to data science.
- Extract an interpretation of data using exploratory data analysis
- Visualise and plot graphical representations of data.

Task1: Basic Statistics, Visualization, and Hypothesis Tests

1. Reload data sets into the R statistical package
2. Perform summary statistics on the data
3. Remove outliers from the data
4. Plot the data using R
5. Plot the data using lattice and ggplot
6. Test a hypothesis about the data

Task 2: Linear Regression

1. Use the R -Studio environment to code OLS models
2. Review the methodology to validate the model and predict the dependent variable for a set of given independent variables
3. Use R graphics functions to visualize the results generated with the model

Task 3: Logistic Regression

1. Use R -Studio environment to code Logistic Regression models
2. Review the methodology to validate the model and predict the dependent variable for a set of given independent variables
3. Use R graphics functions to visualize the results generated with the model

Task 4: Hadoop, HDFS, MapReduce and Pig Purpose

1. Run Hadoop and Hadoop fs and collect help information
2. Run a shell script to perform a word count activity
3. Run a MapReduce job to produce similar output
4. Investigate the UI for MapReduce/HDFS components to track system behavior
5. Run "Pig" statements to execute the same tasks done with MapReduce

REFERENCES

- R Commands - Quick Reference
- Surviving LINUX - Quick Reference
- Hadoop Commands
- HDFS Commands

(17F00512) SOFTWARE TESTING LAB

Course Objectives:

To learn to use the following (or Similar) automated testing tools to automate testing:

- Win Runner/QTP for functional testing.
 - Load Runner for Load/Stress testing.
 - Test Director for test management.
 - JUnit, HTMLUnit, CPPUnit.
- To study state-of-art tools for software testing and Middleware technologies

Course Outcomes:

- Test the software applications using standard tools available in the market

1. Write programs in C Language to demonstrate the working of the following constructs:
 - i) do...while ii) while....do iii) if...else iv) switch v) for
2. A program written in C language for Matrix Multiplication fails. Introspect the causes for its failure and write down the possible reasons for its failure.
3. Consider ATM System and Study its system specifications and report the various bugs.
4. Write the test cases for Banking application.
5. Create test plan document for Library Management System.
6. Create test cases for Railway Reservation.
7. Create test plan document for Online Shopping.

Working with Tool's:

Understand the Automation Testing Approach, Benefits, Workflow, Commands and Perform Testing on one application using the following Tool's.

1. Win runner Tool for Testing.
2. Load runner Tool for Performance Testing.
3. Selenium Tool for Web Testing.
4. Bugzilla Tool for Bug Tracking.

5. Test Director Tool for Test Management.
6. Test Link Tool for Open Source Testing.

References:

1. M G Limaye, “Software Testing – Principles, Techniques and Tools”, Tata McGraw Hill, 2009.
2. Edward Kit, “Software Testing in the Real World - Improving the Process”, Pearson Education, 2004.
3. William E. Perry, “Effective methods for software testing”, 2nd Edition, John Wiley, 2000.