(w.e.f 2015-16)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR COLLEGE OF ENGINEERING ANANTAPUR (Autonomous)
Course Structure for MASTER OF COMPUTER APPLICATIONS
(w.e.f 2015-16)

I YEAR I SEMESTER

Subject Code	THEORY SUBJECTS	L	P	C
15F05101	Discrete Mathematical Structures	4	-	4
15F05102	Object Oriented Programming through C++	4	-	4
15F05103	English language Communication skills	4	-	4
15F05104	Probability and Statistics	4	-	4
15F05105	Accounting and Financial Management	4	-	4
15F05106	English language Communication skills lab	-	4	2
15F05107	OOPS through C++ Lab	-	4	2
15F05108	IT Workshop	-	4	2
	Total	**20**	**12**	**26**

I YEAR II SEMESTER

Subject Code	THEORY SUBJECTS	L	P	C
15F05201	Data Structures	4	-	4
15F05202	Computer Organization	4	-	4
15F05203	Organization Structure and Human Resource Management	4	-	4
15F05204	Automata Theory	4	-	4
15F05205	Database Management Systems	4	-	4
15F05206	Data Structures through C++ Lab.	-	4	2
15F05207	Advanced Communications Skills Lab	-	4	2
15F05208	Database Management Systems Lab	-	4	2
	Total	**20**	**12**	**26**

(w.e.f 2015-16)

MCA II YEAR I SEMESTER

Subject Code	THEORY SUBJECTS	L	P	C
15F05301	Java Programming	4	-	4
15F05302	Computer Networks	4	-	4
15F05303	Design and Analysis of Algorithms	4	-	4
15F05304	Software Engineering	4	-	4
15F05305	Operating Systems	4	-	4
15F05306	Algorithm Analysis Lab	-	4	2
15F05307	Operating Systems Lab	-	4	2
15F05308	Java Programming Lab.	-	4	2
	Total	**20**	**12**	**26**

MCA II YEAR II SEMESTER

Subject Code	THEORY SUBJECTS	L	P	C
15F05401	Web Technologies and Scripting Languages	4	-	4
15F05402	Linux Programming	4	-	4
15F05403	Software Testing Methodologies	4	-	4
	Elective – I / CBCS	4	-	4
15F05404	Management Information System			
15F05405	Distributed Systems			
15F05406	Computer Graphics			
15F05407	Mobile Computing			
	Elective – II	4	-	4
15F05408	Operations Research			
15F05409	Information Retrieval Systems			
15F05410	Cryptography & Network Security			
15F05411	Principles of Programming Languages			
15F05412	Linux Programming Lab	-	4	2
15F05413	Web Technologies Lab.	-	4	2
15F05414	Software Testing Lab	-	4	2
	Total	**20**	**12**	**26**

(w.e.f 2015-16)

MCA III YEAR I SEMESTER

Subject Code	THEORY SUBJECTS	L	P	C
15F05501	Cloud Computing	4	-	4
15F05502	Data Warehousing & Mining	4	-	4
15F05503	Object Oriented Analysis and Design	4	-	4
	Elective – III (MOOC)	4	-	4
15F05504	1. Introduction to Big Data by Harward Univ via edx			
15F05505	2. Introduction to internet of things			
15F05506	3. Human Computer Interaction by California Univ via corsevera			
15F05507	4. Visual Design by Harward Univ via edx			
15F05508	5. Big Data Analytics by IIT Madras, Chennai			
15F05509	6. Computer Maintenances			
15F05510	**7.** Web Performance Optimization			
15F05511	8. Introduction to Hadoop and Mapreduce			
15F05512	9. Introduction to Reverse Engineering Software			
	Elective – IV	4	-	4
15F05513	Web Services			
15F05514	Distributed Databases			
15F05515	Distributed Computing			
15F05516	Mobile Application Development			
15F05517	Object Oriented Analysis and Design Lab	-	4	2
15F05518	Cloud Computing Lab	-	4	2
15F05519	Data Warehousing & Mining Lab	-	4	2
	Total	**20**	**12**	**26**

(w.e.f 2015-16)

MCA III YEAR II SEMESTER

Subject Code	SUBJECTS	CREDITS	MAX.MARKS		Total	MIN. MARKS/GRADES TO PASS
		C	Int.	Ext		
15F05601	**Project Seminar**	2	100	-	100	50
15F05602	**Dissertation / Project Work** Grades : A,B,C,D A - Excellent B - Good C – Satisfactory D - Unsatisfactory	8	-	-	-	A/B/C Grade

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05101 DISCRETE MATHEMATICAL STRUCTURES

Course Objectives:

- To develop problem-solving techniques and explore topics in a variety of areas of discrete mathematics, including but not limited to logic, graph theory, set theory, recursions, combinatory and algorithms.
- To learn to express statements in the language of formal logic and draw conclusions, model situations in terms of graph and set theory
- To find and interpret recursive definitions for mathematical sequences, use combinatorial methods to approach counting problems.

UNIT I

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

UNIT II

Relations: Properties of binary Relations, equivalence,transitive closure, compatibility and partial ordering relations, Lattices,Hasse diagram. Functions: Composition of functions, Inverse Function, Hashing functions, Natural numbers, recursive functions.
Algebraic structures: Algebraic systems, Examples and general properties, Semi groups and monoids, groups and sub groups, homomorphism. Lattice as partially ordered sets, Boolean algebra.

UNIT III

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

UNIT IV

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

UNIT V

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

(w.e.f 2015-16)

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous) :: Anantapuramu
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05102 OBJECT ORIENTED PROGRAMMING THROUGH C++

Course Objectives:

- Understand the Principles of Object Oriented Programming.
- Understand the purpose of inheritance, polymorphism ,and encapsulation.
- Identify classes, objects , members of a class and the relationship among them needed for a specific problem.
- Develop programs using C++ to solve problems.

UNIT I

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

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

UNIT II

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

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

UNIT III

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

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

UNIT IV

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

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

UNIT V

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

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

TEXT BOOKS:

(w.e.f 2015-16)

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

REFERENCE BOOKS:

1. An Introduction to OOP, 3rd Edition, T. Budd, Pearson Education,2008.
2. Programming Principles and Practice Using C++, B.Stroutstrup, Pearson Education.
3. Problem solving with C++, 6th Edition, Walter Savitch, Pearson Education,2007.
4. Mastering C++, K.R.Venu Gopal, Raj Kumar and T.Ravi Shankar, TMH.
5. OOP in C++, 3rd Edition, T.Gaddis, J.Walters and G.Muganda, Wiley DreamTech Press.
6. An Introduction to OOP in C++ with applications in Computer Graphics, 2nd Edition, G.M.Seed, Springer.
7. Programming with ANSI C++, B.Trivedi,Oxford Press.
8. Programming in C++,M.T.Somasekara,PHI.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05103 English Language Communication Skills (Theory)

1. INTRODUCTION:

English is an international language as well as a living and vibrant one. People have found that knowledge of English is a passport for better career, better pay, and advanced knowledge and for communication with the entire world. As it is a language of opportunities in this global age, English is bound to expand its domain of use everywhere. The syllabus has been designed to enhance communication skills of the students of professional courses. The prescribed books serve the purpose of preparing them for everyday communication and to face the global competitions in future.

The prescribed books for detailed study focus on LSRW skills and vocabulary development. The teachers should encourage the students to use the target language. The classes should be interactive and student-centered. They should be encouraged to participate in the classroom activities keenly. The teacher can bring variety by using authentic materials such as newspaper articles, advertisements, promotional material etc.

2. OBJECTIVES:

1. To enable the students to communicate in English for academic and social purpose
2. To enable the students to acquire structure and written expressions required for their profession.
3. To develop the reading and listening skills of the students
4. To enhance the study skills of the students with emphasis on LSRW skills

3. SYLLABUS

Unit I

a) **Reading Comprehension: Vocabulary Building – Antonyms, Synonyms – One-word substitutes, Suffixes and Prefixes, Idioms & Phrases**
b) **Grammar – Verb Forms -Tenses, Articles & Prepositions**

Unit II

Basics of Communication – Types of communication – Listening, Speaking, Reading and Writing Skills –Elements of communication – Tools of communication - Barriers to LSRW Skills

Unit III

Letter Writing – Formal Vs Informal Letters – Types of letters – Business letters, Official letters- Analysis of sample letters

Unit IV

Technical & Academic Writing –Precis writing - Report writing - Elements of reports, types of reports, Form & Style - Project reports , progress reports, routine reports - abstract, Synopsis, and thesis writing, Tips of Academic Writing

Unit V

Technical Presentations – Effective presentation, Elements of presentation – Planning, preparation, and presentation - presentation techniques – Interview Skills

4. EXPECTED OUTCOME:

The students will get the required training in LSRW skills through the prescribed texts and develop communicative competence

Prescribed Books:

1. **Communication Skills by Pushpalatha & Sanjay Kumar, OUP 2012**
2. **Technical Communication by Ashrif Rizvi, Tata McGrah Hill, 2009**

SUGGESTED READING:

1. **Raymond Murphy's English Grammar with CD,** Murphy, Cambridge University Press, 2012.
2. **English Conversation Practice –**Grant Taylor, Tata McGraw Hill, 2009.
3. **A Course in Communication Skills**- Kiranmai Dutt & co. Foundation Books, 2012.
4. **Current English grammar and usage-**S M Guptha, PHI, 2013.
5. **Modern English Grammar-**Krishna SWAMI .McMillan, 2009.
6. **Powerful Vocabulary Builder**- Anjana Agarwal New Age International Publishers, 2011.
7. **Writing with a Purpose, Tickoo and Sasi Kumar, OUP, 2011**
8. **Strengthen Your Writing, Orient Blackswan**
9. **M.L.A Hand Book- Latest Edition (7th Edition)**

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05104 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 Poison 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 simples –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 (Lactic 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-curvilinear regression-Multiple regression-correlation for univariate and bivariate distributions.

TEXT BOOKS:

1. Probability & Statistics for engineers by Dr.J.Ravi chandran WILEY-INDIA publishers.
2. Probability & Statistics by T.K.V.Iyengar, B.Krishna Gandhi and S.Ranganatham and M.V.S.S.N.Prasad, S.Chand publications.

REFERENCES:

1. Probability &statistics by E.Rukmagadachari &E.keshava Reddy, Pearson publisher.
2. Stastical methods byS.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.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05105 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.

(w.e.f 2015-16)

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 Accouting, 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.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous) :: Anantapuram
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05106 English Language Communication Skills Lab

The **Language Lab** focuses on the production and practice of sounds of language and familiarizes the students with the use of English in everyday situations and contexts.

OBJECTIVES:

- To adopt a learner-centered and task-based teaching methodology.
- To facilitate computer-aided multi-media instruction enabling individualized and independent language Learning.
- To sensitize the students to the nuances of English speech sounds, word accent, intonation and rhythm.
- To improve the fluency in spoken English and neutralize mother tongue influence.
- To train students to use language appropriately for interviews, group discussion and public speaking.

SYLLABUS:

Unit I

Phonetics – Introduction to sounds of speech -Vowels and Consonants – Phonetic transcription – Orthographic transcription

Unit II

Syllabification - Word Stress –Intonation – Accent – Rhythm

Unit III

Oral Skills – Situational Dialogues – Role plays

Unit IV

Resume Writing – Cover Letters – E-Mail Writing

Unit V

Technical Presentation – Debates, Group Discussions - Mock Interviews

EXPECTED OUTCOMES:

- Better understanding of nuances of language through audio- visual experience and group activities
- Speaking with clarity and confidence thereby enhancing employability skills of the students
- Students will also exhibit advanced skills of interview, debating and discussion.

(w.e.f 2015-16)

- **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. K-Van Advanced Communication Skills
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.
10. Sanjay Kumar & Pushp Lata. 2011. Communication Skills, OUP

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. **Strengthen Your Steps,** Maruthi Publicaions, 2012.
4. **Speaking English Effectively**, 2nd Edition Krishna Mohan & NP Singh, 2011. (Mcmillan).
5. **Listening in the Language Classroom**, John Field (Cambridge Language Teaching Library),2011
6. **A Hand book for English Laboratories**, E.Suresh kumar, P.Sreehari, Foundation Books,2011
7. **English Pronunciation in Use. Intermediate & Advanced** ,Hancock, M. 2009. CUP
8. **Basics of Communication in English** ,Soundararaj, Francis. 2012.. *New Delhi: Macmillan*
9. **Spoken English** (CIEFL) in 3 volumes with 6 cassettes, OUP.
10. **English Pronouncing Dictionary**, Daniel Jones Current Edition with CD.Cambridge, 17th edition, 2011

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous) :: Anantapuram
Department of Computer Science & Engineering

M.C.A. I Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05107 OBJECT ORIENTED PROGRAMMING THROUGH C++ Lab.

Course Objectives:

•To develop programs in C++ to solve the problems.

List of Sample Problems:

1. Write a C++ program to find the sum of individual digits of a positive integer.
2. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1.Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.
3. Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
4. Write C++ programs that use both recursive and non-recursive functions
   a) To find the factorial of a given integer. b) To find the GCD of two given integers.
   c) To find the nth Fibonacci number.
5. Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.
6. Write a C++ program that uses functions
   a) To swap two integers. b) To swap two characters.
   c) To swap two reals. Note: Use overloaded functions.
7. Write a C++ program to find both the largest and smallest number in a list of integers.
8. Write a C++ program to sort a list of numbers in ascending order.
9. Write a C++ program that uses function templates to solve problems-7&8.
10. Write a C++ program to sort a list of names in ascending order.
11. Write a C++ program to implement the matrix ADT using a class. The operations supported by this ADT are:
    a) Reading a matrix. b) Printing a matrix. c) Addition of two matrices.
    d)Multiplication of two matrices.
12. Implement the matrix ADT presented in the problem-11 using overloaded operators (<<, >>, +, *) and templates.
13. Implement the complex number ADT in C++ using a class. The complex ADT is used to represent complex numbers of the form c=a+ib, where a and b are real numbers. The operations supported by this ADT are:
    a) Reading a complex number. b) Writing a complex number.
    c) Addition of two complex numbers. d) Multiplication of two complex numbers.
14. Write a C++ program that overloads the + operator and relational operators (suitable) to perform the following operations:
    a) Concatenation of two strings. b)Comparison of two strings.
15. Implement the complex number ADT in C++ using a class. The complex ADT is used to represent complex numbers of the form c=a+ib, where a and b are real numbers. The operations supported by this ADT are:
    a) Reading a complex number. b) Writing a complex number.
    c) Addition of two complex numbers. d) Multiplication of two complex numbers.

Note: 1. overload << and >> operators in part a) and part b).

2. overload +, * operators in parts c) and d).

16. Write a template based C++ program that determines if a particular value occurs in an array of values.
17. Write a C++ program that uses functions to perform the following operations:
    a) Insert a sub-string into the given main string from a given position.
    b) Delete n characters from a given position in a given string.
18. Write a C++ program that uses a function to reverse the given character string in place, without any duplication of characters.
19. Write a C++ program to make the frequency count of letters in a given text.
20. Write a C++ program to count the lines, words and characters in a given text.
21. Write a C++ program to determine if the given string is a palindrome or not.
22. Write a C++ program to make frequency count of words in a given text.
23. Write a C++ program that displays the position or index in the string S where the string t begins , or –1 if S doesn't contain t.
24. 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C++ program to find the 2's complement of a binary number.
25. Write a C++ program that counts the number of 1 bits in a given integer.
26. Write a C++ program to generate Pascal's triangle.
27. Write a C++ program to construct of pyramid of numbers.
28. Write a C++ program to compute the Sine series.
29. Write a C++ program that converts Roman numeral into an Arabic integer.
30. Write a C++ program which converts a positive Arabic integer into its corresponding Roman Numeral.
31. Write a C++ program to display the contents of a text file.
32. Write a C++ program which copies one file to another.
33. Write a C++ program to that counts the characters, lines and words in the text file.
34. Write a C++ program to change a specific character in a file.

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

35. Write a C++ program to reverse the first n characters in a file.
36. Write a C++ program that uses a function to delete all duplicate characters in the given string.
37. Write a C++ program that uses a function to convert a number to a character string.
38. Write a C++ program that uses a recursive function to find the binary equivalent of a given non-negative integer n.
39. Write a C++ program to generate prime numbers up to n using Sieve of Eratosthenes method.
40. Write a C++ program
    a) To write an object to a file.
    b) To read an object from the file.
41. Write C++ programs that illustrate how the following forms of inheritance are supported:
    a) Single inheritance b) Multiple inheritance
    c) Multi level inheritance d) Hierarchical inheritance
42. Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class.
43. Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions.

44. Write a C++ program that illustrates the role of virtual base class in building class hierarchy.
45. Write a C++ program that illustrates the role of abstract class in building class hierarchy.

TEXT BOOKS:

1. Mastering C++, K.R.Venu Gopal, Raj Kumar and T.Ravi Shankar, TMH.
2. C++ Programming, D.S.Malik, Cengage Learning.
3. Practical C++ Programming,S.Qualline,SPD.
4. Object Oriented Programming with C++, E.Balaguruswamy, 4th Edition, TMH,2008.
5. OOP with C++,S.Sahay,Oxford Higher Education.
6. C++ and OOP Paradigm,D.Jana,2nd Edition, PHI
7. Fundamentals of C++ Programming,S.Subramanian,Jaico Publishing House.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram

Department of Computer Science & Engineering

M.C.A. I Sem.	T	P	C
	0	4	2

15F05108 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. Crimpling 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.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

	T	P	C
MCA II Sem.	**4**	**0**	**4**

15F05201 Data Structures

Course Objectives:

- Understand basic data structures such as arrays, lists, stacks, queues, trees, hash tables, search trees.
- Analyse performance of algorithms.
- Choose the appropriate data structure for a specified application.
- Implement various searching and sorting algorithms in C++.
- Write programs in C++ to solve problems using data structures such as arrays, lists, stacks, queues, trees, hash tables, search trees.

Unit I

Basic concepts-Data types, Abstact Data Types, Data structures, Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Analysis-Big O, Omega and Theta notations, Basic concepts of STL(Standard Template Library)-Containers, Iterators, algorithms.

Introduction to Linear and Non Linear data structures ,Linear data structures- Linear Lists, Sequential and Linked allocation , The list ADT, array and linked Implementations, Singly Linked Lists-Operations-Insertion, Deletion, Doubly Linked Lists- Operations-Insertion, Deletion, Circularly linked lists.

Representation of single, two dimensional arrays, Sparse matrices and their representation.

Unit II

Stack ADT, definition, operations, array and linked implementations, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definitions and operations ,array and linked Implementations, Circular queues, Insertion and deletion operations, Deque(Double ended queue)ADT, array and linked implementations.

Unit III

Non Linear data structures- Trees – Basic Terminology, Binary tree ADT, array and linked representations, traversals, threaded binary trees, Priority Queue ADT-implementation, Heaps.

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

Unit IV

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

Sorting- Bubble Sort, Insertion Sort, Selection Sort, Radix Sort, Quick sort, Merge sort, Heap Sort Comparison of Sorting methods.

Unit V

Search Trees-Binary Search Trees, Definition, ADT, Implementation, Operations-

Searching, Insertion and Deletion, AVL Trees(Definition and Example only),B-Trees, Definition, B-Tree of order m, operations- Insertion and Searching, Introduction to Red-Black and Splay Trees(Elementary treatment-only Definitions and Examples), Comparison of Search Trees.

Pattern matching algorithm- The Knuth-Morris-Pratt algorithm, Text Compression-Huffman coding algorithm, Tries (examples only).

TEXT BOOKS :

1. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Cengage Learning.
2. Data structures and Algorithm Analysis in C++, Mark Allen Weiss,3rd edition, Pearson Education. Ltd.
3. Data structures, Algorithms and Applications in C++,S.Sahani, Universities Press.

REFERENCE BOOKS :

1. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and D.Mount, Wiley student edition,seventh edition, John Wiley and Sons.
2. ADTs,Data structures and Problem Solving with C++,2nd edition,L.Nyhoff,Pearson(Prentice Hall).
3. Data Structures using C++, D.S. Malik, Cengage Learning, India Edition.
4. Data structures with C++ Using STL, 2nd edition,W.H.Ford and W.R.Topp, Pearson (Prentice Hall).
5. Data Structures using C++,V.H.Patil, Oxford University Press.
6. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
7. Advanced Data structures & Algorithms in C++,V.V.Muniswamy,Jaico Publishing House
8. C++ Plus Data Structures,4th edition,Nell Dale,Jones and Bartlett India Pvt. Ltd.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. II Sem.	T	P	C
	4	0	4

15F05202 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/0 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 Coherance,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.
6. Computer Architecture and Organization,M.Murdocca andV.Heuring,Wiley Inda.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. II Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05203 ORGANIZATIONAL 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.

(w.e.f 2015-16)

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.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. II Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05204 AUTOMATA THEORY

Course Objective:

The course aims to introduce the basic methods and conclusions of the Theory of Computation. At the end of the course, students learn to apply these methods to problems from different fields and be guided by the results in searching for computational solutions to the problems.

1. Understand formal definitions of machine models.
2. Classify machines by their power to recognize languages.
3. Understanding of formal grammars, analysis
4. Understanding of hierarchical organization of problems depending on their complexity
5. Understanding of the logical limits to computational capacity
6. Understanding of un decidable problems.

UNIT I

Preliminaries: Sets, Relations and functions, Methods of proof, Graphs, Languages: Basic Concepts.

Grammars: Definitions and classifications of grammar, Ambiguity, Simplification of CFGs, Normal forms.

UNIT II

Finite State Automata: DFSA, NFSA, Regular Expressions

Finite State Automata: Characterization, Properties and decidability: FSA Regular Grammars, Pumping lemma for regular sets, Closure Properties, Decidability theorems.

Finite State Automata with Output and Minimization: Myhill-Nerode theorem, Finite Automata with output.

UNIT III

Pushdown Automata: The Pushdown Automation, Equivalence between acceptance by empty store and acceptance by Final State, Equivalence of CFG and PDA.

CFG-Properties and Parsing: Pumping Lemma for CFL, Closure Properties for CFL, and Decidability results for CFL.

UNIT IV

Turing Machines: Turing Machine as a acceptor, Turing Machine as a computing device, Techniques for Turing Machine Construction.

Variations of Turing Machine: Generalized Versions, Restricted Turing Machines, Turing Machines as Enumerated, Equivalence between Turing Machines and Type Zero Languages.

UNIT V

Computability Theory: Chomsky hierarchy of languages, linear bounded automata and context sensitive language, LR(0) grammar, decidability of problems, Universal Turing Machine, un decidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

TEXT BOOKS:

1. "Introduction to Formal Languages, Automata Theory and Computation", Kamala Krithivasan, Rama R, PEARSON.
2. "Introduction to Automata Theory Languages and Computation". Hopcroft H.E. and Ullman J. D. Pearson Education

REFERENCE BOOKS:

1. "Introduction To Languages And The Theory of Computation", John C Martin, The McGraw-Hill Companies, Third Edition. (TATA McGRAW HILL)
2. "Introduction to Automata Theory, Formal Languages and Computation", Shyamalendu kandar, PEARSON.
3. "Theory of Computation", Vivek Kulkarni, OXFORD.
4. "Theory of computer Science Automata, Languages and Computation", K.L.P. Mishra, N. Chandrasekaran, PHI, Third Edition.
5. "Formal Languages and Automata Theory", C.K. Nagpal, OXFORD.
6. "Fundamentals of the Theory of Computation, Principles and Practice", Raymond Greenlaw, H. James Hoover, MK(MORGAN KAUFMANN)

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA II Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05205 Data Base Management Systems

Course Objectives:
To provide the student with clear conceptual understandings related to databases. After this course, the student should gain knowledge in the relational model, SQL, database design, storage & indexing, failure recovery and concurrency control

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

UNIT II
Relational Algebra and Calculus: Relational Algebra – Selection and Projection,Set operations, Renaming,Joins,Division,Examples of Algebra Queries,Relational calculus – Tuple relational Calculus – Domain relational calculus – Expressive Power of Algebra and calculus.
Form of Basic SQL Query – Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set – Comparison Operators, Aggregate Operators,NULL values – Comparison using Null values – Logical connectives – AND, OR and NOT – Impact on SQL Constructs, Outer Joins,Disallowing NULL values, Complex Integrity Constraints in SQL Triggers and Active Data bases.

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

(w.e.f 2015-16)

UNIT IV

Overview of Transaction Management: The ACID Properties,Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control,Deadlocks – Performance of Locking – Transaction Support in SQL.

Concurrency Control: Serializability, and recoverability – Introduction to Lock Management – Lock Conversions,Dealing with Dead Locks, Specialized Locking Techniques – Concurrency Control without Locking.

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

UNIT V

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

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

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

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. II Sem	T	P	C
	0	4	2

15F05206 Data Structures Through C++ Lab

Course Objective

- 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

Course Outcomes

- 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

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

Week 2

a) Write a C program that uses stack operations to convert a given infix expression into its postfix
b) 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. Object Oriented Programming with ANSI & Turbo C++, Ashok N.Kamthane, Pearson Education
2. Data Structures using C++, D.S.Malik, 2nd Edition, Cengage Learning
3. Data Structures through C++, Yashavant P.Kanetkar, BPB Publication
4. Data Structures using C and C++, Yedidyah Langsam.Moshe J.Augenstein Aaron M.Tenenbaum, 2nd Edition,PHI
5. Data Structures using C & C++, Rajesh K.Shukla, Wiley-India
6. ADTs, Data Structures and Problem Solving with C++, Larry Nyhoff, Pearson

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

M.C.A. II Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05207 Advanced Communication Skills Lab

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 organize 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.

(w.e.f 2015-16)

3. SYLLABUS:

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

UNIT I

Comprehension – Listening and Reading Comprehension – Inference questions based on Vocabulay – Note-taking and Note-making Exercises – Interpreting and Paraphrasing – Summarizing – Comprehension of unseen passages

UNIT II

Remedial Grammar – Errors of Accidence and syntax with reference to parts of speech, subject-verb agreement, Tenses, Conditional clauses, Use of connectives – Question Tags and short responses.

UNIT III

Speaking Skills – Speaking about the future plans – expressing opinions– Telephone conversations – PPT presentations = Poster presentations – Public Speaking – Welcome Address(Inviting dignitaries to department workshops, symposium and university functions) – Proposing vote of thanks

UNIT IV

Writing Skills – Paragraph Writing – Essay Writing – Project report – Abstracting – Synopsis -Thesis Writing – Technical & Research Paper Writing

UNIT V

Skills Training – Non-Verbal Communication – Decision making – Goal setting – Etiquettes and grooming – Stress management – Time management – Problem Solving

LEARNING OUTCOMES:

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

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.

K-VAN SOLUTIONS-Advanced communication lab

Walden Infotech- Advanced communication lab

1. **DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.**
2. **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. **Management Shapers Series** by Universities Press (India) Pvt Ltd., Himayatnagar, Hyderabad 2008.
6. **Handbook for Technical Writing** by David A McMurrey & Joanne Buckely CENGAGE Learning 2008.
7. **English for Technical Communication for Engineering Students, Aysha Vishwamohan, Tata Mc Graw-Hill 2009.**
8. **Word Power Made Handy,** Shalini Verma, S Chand Publications, 2011.
9. **Effective Technical Communication,** Ashrif Rizvi, TataMcGrahill, 2011.
10. **Listening in the Language Classroom**, John Field (Cambridge Language Teaching Library),2011
11. **English Conversation Practice –**Grant Taylor, Tata McGraw Hill, 2009.
12. **Communication Skills, Sanjay Kumar & Pushpalatha** Oxford University Press, 2012.
13. **A Course in Communication Skills**- Kiranmai Dutt & co. Foundation Books, 2012.
14. **Living English Structures**- William Standard Allen-Pearson, 2011.
15. **Current English grammar and usage-**S M Guptha, PH

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. II Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05208 Database Management Systems Lab

Objective:

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

List of Sample Problems:

1) Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2) Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.
   Example:- Select the roll number and name of the student who secured fourth rank in the class.

3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4) Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5) i)Creation of simple PL/SQL program which includes declaration section, executable section and exception –handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
   ii)Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.
8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.

10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

<u>Example Problems:</u>

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

CLIENT_MASTER :(CLENTNO,
NAME,ADDRESS1,ADDRESS2,CITY,PINCODE,STATE,BALDUE)
PRODUCT_MASTER :
(PRODUCTNO,DESCRIPTION,PROFITPERCENT,UNITMEASURE,QTY_ON_HAND,REORDERLVL,
SELLPRICE,COSTPRICE)
SALESMAN_MASTER: (SALESMANNO,SALESMANNAME, ADDRESS1,ADDRESS2, CITY,PINCODE,
STATE,SLAMT,TGTTOGET,YTDSALES, RESALES)
SALES_ORDER : (OREDRENO,CLIENTNO,ORDERDATE,DELYADDR,
SALESMANNO,DELYTYPE, BILLYN, DELYDATE,ORDERSTATUS)
SALES_ORDER_DETAILS:(ORDERNO,PRODUCTNO, QTYORDERED,QTYDISP, PRODUCTRATE)

i) Insert data into their respective table.

ii) Exercise on retrieving records from table.

a. Find out the names of all clients.
b. Retrieve the entire contents of the client_master table.
c. Retrieve the list of names, city and the state of all clients.
d. List the various products available from the Product_master table.
e. List all the clients who are located in Mumbai.
f. Find the names of salesmen who have a salary equal to Rs.3000.

iii. Exercise on updating records in a table

a. Change the city of client No ‘C00005’ to ‘Bangalore’.
b. Change the BalDue of ClientNo ‘C00001’ to Rs.1000.
c. Change the cost price of ‘Trousers’ to Rs.950.00.
d. Change the city of the salesman to Pune.

iv. Exercise on deleting records in a table

a. Delete all sales men from the salesman_master whose salaries are equal to Rs.35000
b. Delete all products from product_master where the quantity on hand is equal to 100.
c. Delete from Client _Master where the column state holds the value ‘Tamilnadu’.

v. Exercise on altering the table

a. Add a column called ‘Telephone’ of data type ‘number’ and size=’10’ to the Client Master table.
b. Change the size of Sell Price column in Product_Master to 10,2.

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

a. Destroy the table Client_Master along with the data.

vii. Exercise on renaming the table

(w.e.f 2015-16)

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

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

   Client_Master
   Product_Master
   Salesman_Master
   Sales_Order
   Sales_Order_Details

i) Perform the following computations on table data:

a. List the names of all clients having 'a' as the second letter in their names.
b. List the clients who stay in a city whose first letter is 'M'.
c. List all clients who stay in 'Bangalore' or 'Mangalore'.
d. List all clients whose BalDue is greater than value 10000.
e. List all information from the Sales_Order table for orders placed in the month of June.
f. List the order information for Client No'C00001' and 'C00002'.
g. List products whose selling price is greater than 500 and less than or equal to 750.
h. List products whose selling price is more than 500.Calculate a new selling price as, original selling price * .15. Rename the new column in the output of the above query as new_price.
i. List the names, city and state of clients who are not in the state of 'Maharashtra'.
j. Count the total no of orders.
k. Calculate the average price of all the products.
l. Determine the maximum and minimum products prices. Rename the output as max_price and min_price respectively.
m. Count the no of products having price less than or equal to 500.
n. List all the products whose Qty On Hand is less than reorder level.

ii) Exercise on Date Manipulation

a. List the order number and day on which clients on placed their order.
b. List the months (in alphabets) and date when the orders must be delivered.
c. List the Order Date in the format 'DD-Month-YY'. e.g.12-February-02.
d. List the date, 15 days after today's date.

iii). Exercises on using Having and Group by Clauses:

a. Print the description and total qty sold for each product.
b. Find the value of each product sold
c. Calculate the average qty sold for each client that has a maximum order value of 15000.00.
d. Find out the total of all the billed orders for the month of June.

iv). Exercises on Joins and Correlation:

a) Find out the products, which have been sold to 'Ivan Bay Ross'.
b) Find out the products and their quantities that will have to be delivered in the current month.
c) List the product no and description of constantly sold products (i.e. rapidly moving products).
d) Find the names of clients who have purchased 'Trousers'.

e) List the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.

f) Find the products and their quantities for the orders placed by 'Ivan Bay Ross' and 'Mamta Muzumdar'.

g) Find the products and their quantities for the orders placed by Client No 'C00001' and 'C00002'.

v). Exercise on Sub-queries:

a. Find the Product No and Description of non_moving products i.e. Products not being sold.

b. List the customer Name ,Address1,Address2,City and Pin Code for the client who has placed order no 'O19001'.

c. List the client names that have placed orders before the month of May'02.

d. List if the product 'Lycra Top' has been ordered by any client and print the Client_no ,Name to whom it was sold.

e. List the names of clients who have placed orders worth Rs.10,000 or more.

3)Creating Views

4)Writing Assertions

5)Writing Triggers

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

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

7) Creating FORMS.

8) Generating REPORTS.

TEXT BOOKS:

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

Additional Problems:

i)Databases :

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

(w.e.f 2015-16)

“Mysql” database.

Roadway Travels

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

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

- Reservations and Ticketing
- Cancellations

Reservations & Cancellation:

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

Cancellations are also directly handed at the booking office.

In the process of ***computerization*** of **Roadway Travels** you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query’s using SQL to retrieve the data from the database.

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

Experiment 1: E-R Model

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.
Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example: **Entities:**

1. BUS
2. Ticket
3. Passenger

Relationships:

1. Reservation
2. Cancellation

PRIMARY KEY ATTRIBUTES:

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

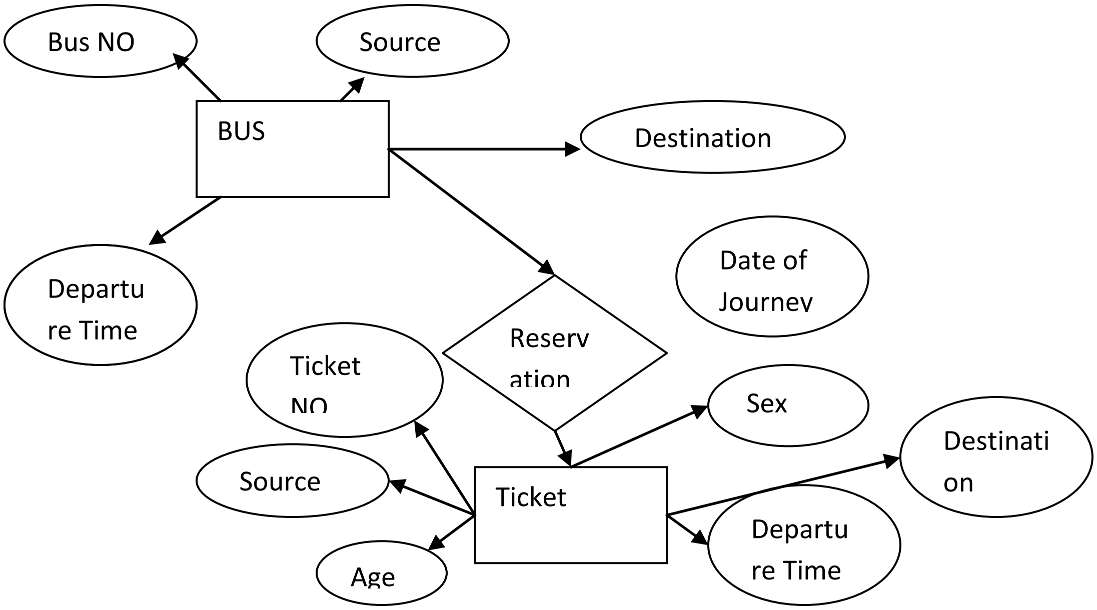
Apart from the above mentioned entities you can identify more. The above mentioned are few.

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

Experiment 2: Concept design with E-R Model

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

Example: E-R diagram for bus



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

Experiment 3: Relational Model

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

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

Passenger

Name	Age	Sex	Address	Ticket_id	Passport ID

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

Experiment 4: Normalization

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

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

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

Passenger

Name	Age	Sex	Address	Passport ID

Passport ID	Ticket_id

(w.e.f 2015-16)

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

Experiment 5: Installation of Mysql and practicing DDL commands

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

Example for creation of a normalized “Passenger” table.

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

Similarly create all other tables.

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

Experiment 6: Practicing DML commands

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

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

Inserting values into “Bus” table:

```
Insert into Bus values (1234,’hyderabad’, ‘tirupathi’);
Insert into Bus values (2345,’hyderabd’,’Banglore’);
Insert into Bus values (23,’hyderabd’,’Kolkata’);
Insert into Bus values (45,’Tirupathi,’Banglore’);
Insert into Bus values (34,’hyderabd’,’Chennai’);
```

Inserting values into “Passenger” table:

```
Insert into Passenger values (1, 45,’ramesh’, 45,’M’,’abc123’);
Insert into Passenger values (2, 78,’geetha’, 36,’F’,’abc124’);
Insert into Passenger values (45, 90,’ram’, 30,’M’,’abc12’);
Insert into Passenger values (67, 89,’ravi’, 50,’M’,’abc14’);
Insert into Passenger values (56, 22,’seetha’, 32,’F’,’abc55’);
```

Few more Examples of DML commands:

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

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

Experiment 7: Querying

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

(w.e.f 2015-16)

Practice the following Queries:

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

Experiment 8 and Experiment 9: Querying (continued…)

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN),

GROUP BY, HAVING and Creation and dropping of Views.

1. Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. **Hint:** Use GROUP BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

Experiment 10: Triggers

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

```
Eg: CREATE TRIGGER updcheck BEFORE UPDATE ON passenger
     FOR EACH ROW
     BEGIN
      IF NEW.TickentNO > 60 THEN
        SET New.Tickent no = Ticket no;
     ELSE
             SET New.Ticketno = 0;
     END IF;
    END;
```

Experiment 11: Procedures

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

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

Experiment 12: Cursors

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

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

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

Tables

BUS

Bus No: Varchar: PK (public key)

Source : Varchar

Destination : Varchar

Passenger

PPNO: Varchar(15)) : PK

Name: Varchar(15)

Age : int (4)

Sex:Char(10) : Male / Female

Address: VarChar(20)

Passenger_Tickets

PPNO: Varchar(15)) : PK

Ticket_No: Numeric (9)

Reservation

PNR_No: Numeric(9) : FK

Journey_date : datetime(8)

No_of_seats : int (8)

Address : Varchar (50)

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

Status: Char (2) : Yes / No

Cancellation

PNR_No: Numeric(9) : FK

Journey_date : datetime(8)

No_of_seats : int (8)

Address : Varchar (50)

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

Status: Char (2) : Yes / No

Ticket

Ticket_No: Numeric (9): PK

Journey_date : datetime(8)

Age : int (4)

Sex:Char(10) : Male / Female

Source : Varchar

Destination : Varchar

Dep_time : Varchar

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. III Sem.	T	P	C
	4	0	4

15F05301 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, jdialog, 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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. III Sem.	T	P	C
	4	0	4

15F05302 Computer Networks

Course Objectives:

- Understand the basic concepts of data communications and computer networks(ex.different network types,applications,protocols,OSI layered architecture model)
- Understand Internet and principles of the TCP/IP protocol suite.
- Understand LANs,Wireless LANs and WANs
- Understand the network security concepts.

UNIT I

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

Physical Layer: digital transmission, multiplexing, transmission media, circuit switched networks, Datagram networks, virtual circuit networks

UNIT II

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

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

UNIT III

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

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

UNIT IV

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

UNIT V

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

TEXT BOOKS:

1.Data Communications and Networking , Behrouz A. Forouzan, Fourth Edition TMH.

2.Computer Networks, Andrew S Tanenbaum, 4th Edition. Pearson Education.

REFERENCE BOOKS:

1.An Engineering Approach to Computer Networks,S.Keshav,2nd Edition,Pearson Education.
2.Understanding communications and Networks,3rd Edition, W.A.Shay,Cengage Learning.
3.Computer and Communication Networks Nader F. Mir, Pearson Education
4.Computer Networking:A Top-Down Approach Featuring the Internet,James F.Kurose,K.W.Ross,3rd Edition,Pearson Education.
5.Data and Computer Communications,G.S.Hura and M.Singhal,CRC Press,Taylor and Francis Group.
6.Data Communications and Computer Networks,P.C.Gupta,PHI.
7.Computer Networks: A Systems approach, Larry L. Peterson & Bruce S. Davie, Fifth edition, Elsevier, rp2012.
8. Data Communications and Computer Networks,C.Murali,ELSEVIER.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. III Sem.	T	P	C
	4	0	4

15F05303 Design and Analysis Of Algorithms

Course Objectives:

- To know the importance of the complexity of a given algorithm.
- To study various algorithmic 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.
- Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete

Unit - I :

Algorithms, Pseudo code for expressing algorithms, performance analysis:- space complexity and time complexity, Asymptotic notations:- Big Oh notation, Omega Notation, Theta notation, amortized complexity

Unit – II:

Divide and Conquer: General method, applications: Defective Chessboard, Binary Search, Quick Sort and its time complexity, Merge Sort and Stressen’s matrix multiplication.
Greedy Method: General method, applications: job sequencing with deadlines, knapsack problem, single source shortest path, Minimum cost Spanning Trees

Unit – III :

Dynamic programming: General Method, applications: Single Source Shortest path, 0/1 knapsack, All Pairs shortest path, travelling sales person problem and reliability design

Unit – IV :

Back tracking: General Method, applications: 8 – queens problem, sum of subsets problem, graph coloring and hamiltonian cycles, Knapsack Problem.

Branch and Bound: General method, applications: travelling sales person, 0/1 knapsack problem, LC Branch and Bound and FIFO Branch and Bound

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, University Press.
2. “Design and Analysis of Algorithms”, Parag Himanshu Dave, Himanshu Bhalchandra Dave, Second Edition, Pearson Education

Reference Books :

1. “Introduction to Algorithms”, secondedition,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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram

Department of Computer Science & Engineering

MCA. III Sem	**T**	**P**	**C**
	4	**0**	**4**

15F05304 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.
- Obtain knowledge about principles and practices of software engineering.
- Focus on the fundamentals of modeling a software project.
- Obtain knowledge about estimation and maintenance of software systems

Unit - I :

Software and Software Engineering: The Nature of Software, The Unique Nature of WebApps, Software Engineering, 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 Terminology, Product and Process.

Unit – II:

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements.

Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling.

Unit – III :

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

Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design.

Component-Level Design: What is a Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Level Design for WebApps.

Unit – IV :

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

Coding and Testing: Coding, Code Review, Software Documentation, Testing, Testing in the Large versus Testing in the Small, Unit Testing, Black-Box Testing, White-Box Testing, Debugging, Program Analysis Tools, Integration Testing, Testing Object-Oriented Programs, System Testing, Some General Issues Associated with Testing.

Unit – V :

Software Project Management: Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO-A Heuristic Estimation Technique, Halstead's Software Science-An Analytical Technique, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management. .

Software Maintenance: Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models, Estimation of Maintenance cost.

Text Books :

1. Software Engineering A practitioner's Approach, Roger S. Pressman, Seventh Edition McGrawHill International Edition.
2. Fundamentals of Software Engineering, Rajib Mall, Third Edition, PHI.

Reference Books :

1. Software Engineering, Ian Sommerville, Ninth edition, Pearson education.
2. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
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.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. III Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05305 OPERATING SYSTEMS

Course Objectives:

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

UNIT I

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

UNIT II

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

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

UNIT III

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

UNIT IV

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection,File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance, Case Studies: Linux, Windows.

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

(w.e.f 2015-16)

UNIT V

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.
Protection – System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection, Case Studies: Linux, Windows.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. III Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05306 ALGORITHMS ANALYSIS 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 Hamiltaonian 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.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. III Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05307 Operating Systems Lab

Course Objectives:

- To implement CPU scheduling algorithms,file allocation methods and page replacement algorithms in C.

1. Simulate the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority

2. Simulate all file allocation strategies
   a) Sequential b) Indexed c) Linked

3. Simulate MVT and MFT
4. Simulate all File Organization Techniques

   a) Single level directory b) Two level c) Hierarchical d) DAG

5. Simulate Bankers Algorithm for Dead Lock Avoidance

6. Write a C program to create a child process and allow the parent to display "Hello"and the child to display "Welcome" on the screen.

7. Simulate all page replacement algorithms

   a) FIFO b) LRU c) LFU Etc…

8. Simulate Paging Technique of memory management.

9. Write C programs that make a copy of a file using i)standard I/O and ii) system calls.

10. Write C programs that count the number of blanks in a text file using i) standard

TEXT BOOKS:

1.An Introduction to Operating Systems,P.C.P Bhatt,2nd edition,PHI.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram

Department of Computer Science & Engineering

MCA. III Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05308 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 ax2 + 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 StringToknizer 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 substraction 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 eparated 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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05401 WEB TECHNOLOGIES AND SCRIPTING LANGUAGES

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.

Learning Objectives

a. Learn to develop mock up pages with good look and feel using
   1. HTML
   2. CSS
   3. Usability features.

b. Learn to validate the above mock up pages by a client side script (Javascript).

c. Learn to create, insert, retrieve, update data in Database using JDBC.

d. Learn to work with web servers like Tomcat.

e. Learn the game execution by using Servlets, JSP's, AJAX.

f. Learn heuristic based UI principles

g. Learn the basics of PHP programming language.

h. Create dynamic web pages using the PHP scripting language and a MySQL database.

Unit I:

HTML:

Common tags- List, Tables, Images, Forms, Frames and IFrames, Cascading Style Sheets; Introduction to Java Scripts, Dynamic HTML with Java Script.

XML:

Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, DOM and SAX Parsers, XHTML

Unit II:

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions

Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads

Connecting to database (MySQL as reference), executing simple queries, handling results

Handling sessions and cookies

Unit III:

Object Oriented Programming with PHP: Creating classes and objects, public, private and protected access, constructor and destructor, Inheritance, Overriding and overloading methods, calling base class methods, static members, interfaces, abstract classes

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

XML Processing: creating an XML file, using simple XML functions, extracting elements and attributes, modifying XML elements and attributes, adding and deleting elements and attributes, parsing XML file with parser functions

Simple Ajax implementation with PHP

Unit IV:

Introduction to Servlets: Lifecycle of a Servlet, The Servlet API, The javax.servlet Package, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions.

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages
Using Cookies-Session Tracking, Security Issues

Database Access: Using JDBC to access Database from JSPs and Servlets

Unit V:

Introduction to MVC architecture, Anatomy of a simple struts application, struts-config.xml file, Presentation layer with JSP, Struts Controller class, JSP bean, html and logic tag libraries, ActionForms, DynaActionForm, Actions, Forwarding, Error Handling, Database Connection Pooling, validation frame work and examples for simple data types, Internationalization

TEXT BOOKS:

1. Web Programming, building internet applications, Chris Bates 2nd edition, WILEY Dreamtech (UNIT 1)
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill (Unit 2,3)
3. Java Server Pages –Hans Bergsten, SPD O'Reilly (UNITs 3,4,5)
4. The World of Scripting Languages , David Barron,Wiley Publications.
5. Professional Jakarta Struts - James Goodwill, Richard Hightower, Wrox Publishers.

REFERENCE BOOKS:

1. Programming world wide web,R.W.Sebesta,Fourth edition,Pearson.
2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES , Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program , Dietel and Nieto,Pearson.
4. Jakarta Struts Cookbook , Bill Siggelkow, S P D O'Reilly.
5. Professional Java Server Programming,S.Allamaraju and othersApress(dreamtech).
6. Java Server Programming ,Ivan Bayross and others,The X Team,SPD
7. Web Warrior Guide to Web Programmming-Bai/Ekedaw-Thomas
8. Beginning Web Programming-Jon Duckett WROX.
9. Java Server Pages, Pekowsky, Pearson.
10. Java Script,D.Flanagan,O'Reilly,SPD.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05402 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.

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05403 Software Testing Methodologies

Course Objectives:

- Understand different software testing techniques and strategies.

UNIT-I

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

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

UNIT-II

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

UNIT-III

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

Paths, Path products and Regular expressions:- path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

UNIT-IV

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

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

UNIT-V

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

TEXT BOOKS:

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

REFFERENCE BOOKS

1. Software Testing,3rd edition,P.C.Jorgensen,Aurbach Publications(Dist.by SPD).
2. Software Testing in the Real World – Edward Kit, Pearson.
3. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.

4. Art of Software Testing – Meyers, John Wiley.
5. Software Testing,N.Chauhan,Oxford University Press.
6. Software Testing,M.G.Limaye,TMH.
7. Software Testing,S.Desikan,G.Ramesh,Pearson.
8. Foundations of Software Testing,D.Graham & Others,Cengage Learning.
9. Foundations of Software Testing,A.P.Mathur,Pearson.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA IV Sem.	**T**	P	C
	4	0	4

15F05404 Management Information Systems (ELECTIVE-I)

Course Objectives:

- Trace the evolution, components, objectives, benefits, limitations and role of MIS in the changing global scenario.
- Offer knowledge about information different types of information and data collection methods.
- MIS and its role in managerial decision, information system development and Data Communication and Networking.

Learning outcome:

- This course gives the student the foundation for application of MIS for taking managerial decisions in the changing technological and global scenario.

UNIT-I: Introduction to MIS: Meaning and Definition- Evolution-Characteristics- Objectives - Components of Information Systems-- Benefits and limitations in MIS Development-Role of MIS in the changing global scenario.

UNIT-II: Information- Knowledge- Business Intelligence: Information Concept- Classification of Information- Methods of Data and Information Collection- Value of the Information- Knowledge and Knowledge Management Systems-Database Management Systems-Organizing Data in Traditional File Environment- The Database Approaches to Data Management- Using Databases to improve Business and Decision Making.

UNIT- III: MIS and Decision Making: Decision-Making Concepts- Decision-Making Process- Behavioral Concepts in Decision-Making- Organizational Decision-Making. International Information Systems- Organizing International Information Systems- Managing Global Systems- Technology Issues and Opportunities for Global Value Chains.

UNIT – IV: Information Systems Development: Transaction Processing System -Transaction Processing cycle- Decision Support Systems(DSS) - Executive Support Systems(ESS) - Office Automation System(OAS)-Information system security and Control.

UNIT – V: Data Communication and Networking : Data Communication methods- Communication Networks – Local Area Network(LAN)-Wide Area Network(WAN)- Metropolitan Area Network(MAN)- Integrated Services Digital Networks(ISDN)-Electronic Communication. Emerging need for MIS in Service Sector- Ethical and Social issues in IT.

TEXT BOOKS:

1. Kenneth C. Laudon, Jane P. Laudon. Management Information Systems- Managing the Digital Firm, Pearson Education, 11th edition, 2010.
2. Waman S Jawadekar, Management Information Systems Text and Cases- MGH- Jun 2009.

REFERENCES:

1. Ralph Stair, George Reynolds , Information Systems, 10th edition,Cengage Learning,2013.
2. C.S.V Murthy. Management information systems text and applications-. HPH- 2011- 3rd ed. reprint.
3. R.G. Murdick- J.E Ross and J.R clagget. Information Systems for Modern Management. PHI- 1994- 3rd Ed-.
4. Robert schultberis- Mary sumner. Management information systems. PHI-1999.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05405 DISTRIBUTED SYSTEMS
(ELECTIVE- I)

Course Objectives:

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

UNIT I

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

UNIT II

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

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

UNIT III

Peer to Peer Systems–Introduction, Napster and its legacy,Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies-Squirrel, Ocean Store.

Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement-Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT IV

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

UNIT V

Security-Introduction, Overview of Security techniques, Cryptographic algorithms, Digital signatures, Case studies-Kerberos,TLS,802.11 WiFi.

Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy case study, Release consistency and Munin case study, Other consistency models, CORBA case study-Introduction, CORBA RMI,CORBA Services.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05406 COMPUTER GRAPHICS
(ELECTIVE-I)

Objectives :

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

UNIT I

Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

UNIT II

2-D Geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D Viewing : The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT III

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

UNIT IV

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

UNIT V

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

Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications

(w.e.f 2015-16)

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA IV Sem.	T	P	C
	4	0	4

15F05407 MOBILE COMPUTING
(ELECTIVE-I)

Course Objectives:

- Understand the basic concepts of mobile communication and mobile devices.
- Understand the basic concepts of mobile computing and architecture of mobile communication.

UNIT I

MOBILE COMMUNICATIONS: Introduction to Mobile Communications, Mobile Computing, Mobile Computing Architecture, Mobile Devices, Mobile System Networks, Mobility Management, Security.

UNIT –II

GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS(GSM): Mobile Services, System Architecture, Protocols, Localization & Calling, Handover, Security. **GPRS:** GPRS System Architecture, **UMTS**: UMTS System Architecture.

UNIT –III

MOBILE IP NETWORK LAYER: IP and Mobile IP Network Layers, Packet Delivery and Handover Management, Location Management, Registration, Tunnelling and Encapsulation, Route Optimization, DHCP.
MOBILE TRANSPORT LAYER: Conventional TCP/IP Protocols, Indirect TCP, Snooping TCP, Mobile TCP, Other Transport Layer Protocols for Mobile Networks.

UNIT IV

DATABASE ISSUES: Database Hoarding & Caching Techniques, C – S Computing & Adaptation, Transactional Models, Query processing, Data Recovery Process & QoS Issues.
DATA DISSEMINATION AND SYNCHRONIZATION: Communications Asymmetry, Classification of Data Delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing Methods, Digital Audio and Video Broadcasting (DAB & DVB). Data Synchronization – Introduction, Software, and Protocols

UNIT V

MOBILE AD HOC NETWORKS (MANETS): Introduction, Applications & Challenges of a MANET, Routing, Classification of Routing Algorithms, Algorithms such as DSR, AODV, DSDV, TORA, Cluster-head Gateway Switch Routing, Flat routing Table Driven Protocol, Optimized Link State Routing Protocol.

TEXT BOOKS:

1.Raj Kamal, “Mobile Computing”, Oxford University Press, Second edition.
2.Jochen Schiller, “Mobile Communications”, Pearson education, Second Edition.

(w.e.f 2015-16)

REFERENCE BOOKS:

1. Stojmenovic and Cacute, “Handbook of Wireless Networks and Mobile Computing”, Wiley, 2002, ISBN 0471419028.
2. Reza Behravanfar, “Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML”, ISBN: 0521817331, Cambridge University Press,Oct 2004.
3. Asoke K Talukder, et al, “Mobile Computing”, Tata McGraw Hill, 2008.
4. Yi-Bang Lin, et al, ‘Wireless and Mobile Network Architectures”, Wiley-INDIA, 2008.
5. Dharma prakash Agarwal et al, “Iintroduction to Wireless and Mobile Systems”, Cengage

Learning, Second Edition, 2007.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05408 OPERATIONS RESEARCH
(ELECTIVE-II)

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

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

UNIT II

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

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

UNIT III

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

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

UNIT IV

Dynamic programming: Characteristics of dynamic programming. Dynamic programming approach for Priority Management employment smoothening, Stage Coach/Shortest Path and Reliability problems.
Games Theory: Competitive games, rectangular game, saddle point, minimax (maximin) method of optimal strategies, value of the game. Solution of games with saddle points, dominance principle. Rectangular games without saddle point – mixed strategy for 2 X 2 games.

UNIT V

Inventory models: Inventory costs. Models with deterministic demand – model (a) demand rate uniform and production rate infinite, model (b) demand rate non-uniform and production rate infinite, model (c) demand rate uniform and production rate finite.
Queuing Theory: Essential Features of a queuing system. Performance measures of a queuing system. Model 1: {(M/M/1) : (∞/FCFS)} Single server, Unlimited Queue model. Model 2: {(M/M/1) : (∞/SIRO)} Single server, Unlimited Queue model. Model III: {(M/M/1) : (N/FCFS)} Single server, Finite Queue model.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05409 INFORMATION RETRIEVAL SYSTEMS (ELECTIVE-II)

Course Objectives:

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

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

UNIT I

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

UNIT II

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

UNIT III

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

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS :

1. Information Storage and Retrieval Systems: Theory and Implementation, Kowalski, Gerald, Mark T Maybury,Springer.
2. Modern Information Retrival , Ricardo Baeza-Yates, Pearson Education, 2007.

3. Information Retrieval: Algorithms and Heuristics, David A Grossman and Ophir Frieder, 2nd Edition, Springer, 2004.
4. Information Retrieval Data Structures and Algorithms, William B Frakes, Ricardo Baeza-Yates, Pearson Education, 1992.
5. Information Storage & Retieval , Robert Korfhage , John Wiley & Sons.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05410 CRYPTOGRAPHY & NETWORK SECURITY (ELECTIVE-II)

Course Objectives:

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

UNIT – I

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

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

UNIT – II

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

UNIT – III

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

UNIT – IV

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

UNIT – V

Web Security: Web security considerations, Secure Socket Layer and Transport Layer Security, Secure electronic transaction **Inruders, Virus and Firewalls:** Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls **Case Studies on Cryptography and security:** Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability, Virtual Elections

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Cryptography and Network Security : Forouzan& Mukhopadhyay, Mc Graw Hill, 2nd Edition
2. Principles of Computer Sceurity: WM.Arthur Conklin, Greg White, TMH
3. Introduction to Network Security: Neal Krawetz, CENGAGE Learning
4. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning.
5. Information Systems Security,Godbole,Wiley Student Edition.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05411 PRINCIPLES OF PROGRAMMING LANGUAGES (ELECTIVE-II)

Course Objectives:

- Understand Programming languages design and implementation.
- Make a comparative study of Programming languages features.
- Explore various important programming methodologies such as functional programming, logic programming, programming with ADTs, OOP and programming with scripting languages.

UNIT I

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

UNIT II

Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization.

UNIT III

Expressions and Statements: Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.

Subprograms and Blocks: Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines.

UNIT IV

Abstract Data types: Abstractions and encapsulation, introductions to data abstraction, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95

Concurrency: Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads.

Exception handling: Exceptions, exception Propagation, Exception handler in Ada, C++ and Java.

UNIT V

Logic Programming Language : Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

Functional Programming Languages: Introduction, fundamentals of FPL, LISP, ML, Haskell, application of Functional Programming Languages and comparison of functional and imperative Languages.

Scripting Language: Pragmatics, Key Concepts, Case Study : Python – Values and Types, Variables , Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library.

TEXT BOOKS:

1. Concepts of Programming Languages Robert .W. Sebesta 8/e, Pearson Education, 2008.
2. Programming Languages, K. C.Louden, 2nd Edition, Cengage Learning, 2003.
3. Programming Language Design Concepts, D. A. Watt, Wiley dreamtech, rp-2007.
4.

REFERENCE BOOKS:

1. Programming Languages, 2nd Edition, A.B. Tucker, R.E. Noonan, TMH.
2. LISP, Patric Henry Winston and Paul Horn, Pearson Education.
3. Programming in Prolog, W.F. Clocksin, & C.S.Mellish, 5th Edition, Springer.
4. Programming Python, M.Lutz, 3rd Edition, O’reilly, SPD, rp-2007.
5. Core Python Programming, Chun, II Edition, Pearson Education, 2007.
6. Programming Language Pragmatics, Scott, 3rd edition, ELSEVIER.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	T	P	C
	0	4	2

15F05412 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.

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, Elllie Quigley, Pearson Education.
5. Sed and Awk, O.Dougherty&A.Robbins,2nd edition,SPD.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	T	P	C
	0	4	2

15F05413 WEB TECHNOLOGIES LAB.

Course Objectives:

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

List of Sample Problems

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

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

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

3. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

*4. Bean Assignments

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

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

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

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

Light Transition	Automobile State
Red ---> Yellow	Ready
Yellow ---> Green	Move
Green --> Red	Stopped

5. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
6. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.
7. Implement the “Hello World!” program using JSP Struts Framework.
8.Redo the problem 5 using PHP.

Additional Assignment Problems for the WT Lab.:

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

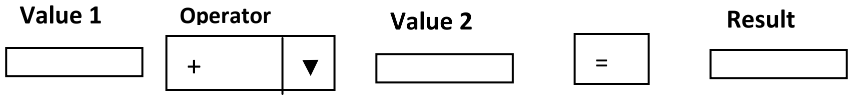
Write a java swing application that takes a text file name as input and counts the characters, words and lines in the file. Words are separated with white space characters and lines are separated with new line character.

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

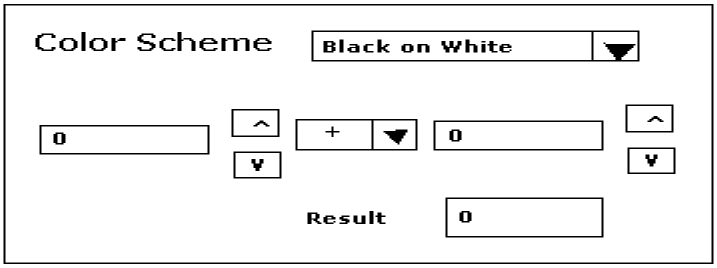
Write an HTML page that contains a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).

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

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



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



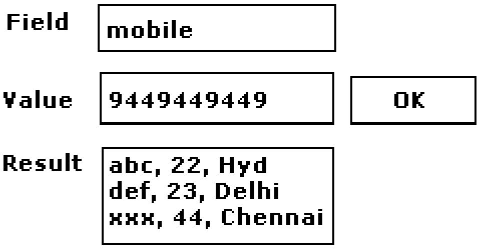
Color Scheme: Black on White

0 + 0

Result 0

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

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



Field mobile

Value 9449449449

Result
abc, 22, Hyd
def, 23, Delhi
xxx, 44, Chennai

Consider the following web application for implementation:
The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
If name and password matches, serves a welcome page with user's full name.
If name matches and password doesn't match, then serves “password mismatch” page
If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application using:

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

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. IV Sem.	T	P	C
	0	4	2

15F05414 Software Testing Lab

Course Objectives:

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

a) Win Runner/QTP for functional testing.
b) LoadRunner for Load/Stress testing.
c) Test Director for test management.
d) JUnit,HTMLUnit,CPPUnit.

Sample problems on testing:

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

Additional problems on testing:

1.Test the following using JUnit and CPPUnit:
i)Sorting problems ii)Searching problems iii)Finding gcd of two integers iv)Finding factorial of a number.
2.Test web based forms using HTMLUnit.
3.Test database stored procedures using SQLUnit.
(Use sufficient number of test cases in solving above Problems)

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA V Sem	T	P	C
	4	0	4

15F05501 CLOUD COMPUTING

Prerequisite: Computer Networks and Operating Systems

Course Objectives:
Cloud computing has evolved as a very important computing model, which enables information, software, and shared resources to be provisioned over the network as services in an on-demand manner. This course provides an insight into what is cloud computing and the various services cloud is capable.

UNIT I: Systems Modeling, Clustering and Virtualization
Distributed System Models and Enabling Technologies, Computer Clusters for Scalable Parallel Computing, Virtual Machines and Virtualization of Clusters and Data centers.

UNIT II: Foundations
Introduction to Cloud Computing, Migrating into a Cloud, Enriching the 'Integration as a Service' Paradigm for the Cloud Era, The Enterprise Cloud Computing Paradigm.

UNIT III: Infrastructure as a Service (IAAS) & Platform and Software as a Service (PAAS / SAAS)
Virtual machines provisioning and Migration services, On the Management of Virtual machines for Cloud Infrastructures, Enhancing Cloud Computing Environments using a cluster as a Service, Secure Distributed Data Storage in Cloud Computing.
Aneka, Comet Cloud, T-Sytem's, Workflow Engine for Clouds, Understanding Scientific Applications for Cloud Environments.

UNIT IV: Monitoring, Management and Applications
An Architecture for Federated Cloud Computing, SLA Management in Cloud Computing, Performance Prediction for HPC on Clouds, Best Practices in Architecting Cloud Applications in the AWS cloud, Building Content Delivery networks using Clouds, Resource Cloud Mashups.

UNIT V: Governance and Case Studies
Organizational Readiness and Change management in the Cloud age, Data Security in the Cloud, Legal Issues in Cloud computing, Achieving Production Readiness for Cloud Services.

TEXT BOOKS:

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing , Kai Hwang, Geoffery C.Fox, Jack J.Dongarra, Elsevier, 2012.

REFERENCE BOOKS:

1. Cloud Computing : A Practical Approach, Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Tata McGraw Hill, rp2011.
2. Enterprise Cloud Computing, Gautam Shroff, Cambridge University Press, 2010.
3. Cloud Computing: Implementation, Management and Security, John W. Rittinghouse, James F.Ransome, CRC Press, rp2012.
4. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, George Reese, O’Reilly, SPD, rp2011.
5. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O’Reilly, SPD, rp2011.

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. V Sem.	T	P	C
	4	0	4

15F05502 DATA WAREHOUSING AND MINING

Course Objectives:

- Understand data mining principles and techniques: Introduce DM as a cutting edge business intelligence method and acquaint the students with the DM techniques for building competitive advantage through proactive analysis, predictive modeling, and identifying new trends and behaviors.
- Building basic terminology.
- Learn how to gather and analyze large sets of data to gain useful business understanding.
- Learn how to produce a quantitative analysis report/memo with the necessary information to make decisions.
- Describing and demonstrating basic data mining algorithms, methods, and tools
- Identifying business applications of data mining
- Develop and apply critical thinking, problem-solving, and decision-making skills.

UNIT I

Introduction: Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Integration of a Data Mining System with a Database or a Data Warehouse System, Issues in Data Mining.

Data Preprocessing: Need for Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

UNIT II

Data Warehouse and OLAP Technology for Data Mining: Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Usage of Data Warehousing Online Analytical Processing and Mining

Data Cube Computation: Efficient Methods for simple Data Cube Computation (Full Cube, Iceberg Cube, Closed Cube and Shell Cube), Discovery Driven exploration of data cubes, Attribute-Oriented Induction for data characterization and its implementation

UNIT III

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

UNIT IV

Classification and Prediction: Description and comparison of classification and prediction, preparing data for Classification and Prediction

Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Backpropagation

Prediction, linear and non-linear regression, evaluating accuracy of a Classifier or a Predictor

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA V Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05503 OBJECT ORIENTED ANALYSIS AND DESIGN

Course Objectives:

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

UNIT I

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

UNIT II

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.
Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages.
Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT III

Basic Behavioral Modeling-I: Interactions, Interaction diagrams.
Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

UNIT IV

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

UNIT V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA V Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05513 WEB SERVICES (ELECTIVE-IV)

Course Objectives:

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

UNIT I

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

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

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

UNIT II

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

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

UNIT III

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

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

UNIT IV

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

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

UNIT V

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

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

TEXT BOOKS:

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

REFERENCES:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA V Sem

T	P	C
4	0	4

15F05514 DISTRIBUTED DATABASES
(ELECTIVE-IV)

Course Objectives

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

UNIT I

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

UNIT II

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

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

UNIT III

The Management of Distributed Transactions, A Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions

Concurrency Control, Foundation of Distributed Concurrency Control, Distributed Deadlocks, Concurrency Control based on Timestamps, Optimistic Methods for Distributed Concurrency Control.

UNIT IV

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

UNIT V

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

Database Integration, Scheme Translation, Scheme Integration, Query Processing Query Processing Layers in Distributed Multi-DBMSs, Query Optimization Issues Transaction Management Transaction and Computation Model, Multidatabase Concurrency Control, Multidatabase Recovery, Object Orientation and Interoperability, Object Management Architecture CORBA and Database interoperability, Distributed Component Object Model, COM/OLE and Database Interoperability, PUSH-Based Technologies

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Distributed Database Systems, Chanda Ray, Pearson.
2. Distributed Database Management Systems, S.K.Rahimi and Frank.S.Haug, Wiley.

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA V Sem

T	P	C
4	0	4

15F05515 DISTRIBUTED COMPUTING (ELECTIVE-IV)

Course Objectives

On completion of this subject the student will

- Be able to understand emerging distributed technologies.
- Be able to design large-scale distributed systems.
- Be able to implement cluster and grid applications.

UNIT I

Introduction

The different forms of computing, The strengths and weaknesses of Distributed computing, Operating system concepts relevant to distributed computing, the architecture of distributed applications. Paradigms for Distributed Applications, choosing a Paradigm for an application (trade-offs).

UNIT II

Cluster Computing

Parallel computing overview, cluster computing – Introduction, Cluster Architecture, parallel programming models and Paradigms, Applications of Clusters.

UNIT III

Grid Computing

Introduction, Grid Computing Anatomy – Architecture, Architecture and relationship to other Distributed Technologies, Grid computing road map. Merging the Grid services Architecture with the Web Services Architecture.

UNIT IV

Open Grid Service Architecture – Introduction, Architecture and Goal, Sample Use cases: Commercial Data Center, National Fusion Collaboratory, Online Media and Entertainment. OGSA platform Components, Open Grid Services Infrastructure.

UNIT V

Globus GT 3 Toolkit – Architecture, Programming Model, A sample implementation, High Level services, OGSI.NET Middleware Solutions.

TEXT BOOKS:

1. Grid Computing, Joshy. Joseph & Craig Fellenstein, Pearson education, 2004
2. Distributed Computing, Principles and Applications, M.L.Liu, Pearson Education, 2004
3. High Performance Cluster Computing, Rajkumar Buyya, Pearson education.

REFERENCE BOOKS:

1. Grid Computing – Making the global infrastructure a reality, Fran Berman, Geoffrey C Fox, Anthony J G Hey, Wiley India, 2010
2. A Networking Approach to Grid Computing, D.Minoli, Wiley & sons, 2006
3. Grid Computing: A Practical Guide to Technology and Applications, A.Abbas, Firewall Media, 2008

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram

Department of Computer Science & Engineering

(ELECTIVE-IV)

	T	P	C
MCA. V Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05516 MOBILE APPLICATION DEVELOPMENT

Course Objectives:

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

Unit I

J2ME Overview

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

Unit II

J2ME Architecture and Development Environment

J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2ME Style, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit
J2ME Best Practices and Patterns: The Reality of Working in a J2ME World, Best Practices

Unit III

Commands, Items, and Event Processing
J2ME User Interfaces, Display Class, the Palm OS Emulator, Command Class, Item Class, Exception Handling
High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, Text Box Class, Ticker Class
Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation

Unit IV

Record Management System:

Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener
JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions

(w.e.f 2015-16)

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

Unit V
Generic Connection Framework

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

(w.e.f 2015-16)

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA V Sem.	**T**	**P**	**C**
	4	**0**	**4**

15F05505:Introduction to Internet of Things
(MOOCs)

Course Objectives:

- Vision and Introduction to IoT.
- Understand IoT Market perspective.
- Data and Knowledge Management and use of Devices in IoT Technology.
- Understand State of the Art – IoT Architecture.
- Real World IoT Design Constraints, Industrial Automation and Commercial Building Automation in IoT

Course Outcomes:

- To learn the concepts of IoT and develop a product according to Market needs.
- To implement machine learning and Automation behavior in real-time.
- Interpret the vision of IoT from a global context.

UNIT I

Introduction to IoT, applications, connectivity layers, addressing, networking and connectivity issues, network configurations, multi-homing, Sensing and Actuation

Basics of IoT Networking: IoT components, inter-dependencies, SoA, gateways, wireless networks, scalability, protocol classification, MQTT & SMQTT, CoAP, XMPP, AMQP

Connectivity Technologies: IEEE 802.15.4, ZigBee, 6 LOWPAN, RFID, HART, wireless HART, NFC, Bluetooth, Zwave and ISA100.11A

UNIT II

Sensor Networks: Wireless Sensor Networks, Sensor Nodes, Types of object detection by sensor(s), Sensor Web, Cooperation in Wireless Ad Hoc and Sensor Networks, Security Challenges in Cooperation, Node behavior in WSNs, Event-Aware Topology management, Information theoretic self management, Social Sensing, Applications of WSN, UAV Networks, FANETs, Machine to Machine communication, Interoperability in IoT, Introduction to Arduino, Integration of Sensors and Actuators with Arduino

UNIT III

Introduction to Python Programming: Python IDE, Data types, Control statements, Functions, Scope of Variables, Modules, Exception Handling, File Read Write operations, Image read write operations, Networking in python

Software Defined Networking: Overview and limitations of current network, moving to SDN, SDN architecture, Components of SDN, Challenges

Software defined IoT Networking: SDN for IoT, SDN for WSNs, SDN for Mobile Networks
Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi

UNIT IV

Cloud Computing: Evolution, NIST visual model, Characteristics, Components, Service Models, Deployment models, Service management and security, CloudSim, CloudAnalyst, GreenCloud, Open Source and Commercial Clouds

Sensor Cloud: Introduction, Architecture, Work flow, Management issues, Target tracking case study

Fog Computing: Introduction, need for fog computing, use in IoT, Architecture, Advantages, Applications and challenges.

UNIT V

Smart Cities and Homes: Introduction and application areas, Data Fusion, Smart Parking, Energy management in Smart cities, Smart Homes, Home Area Networks, HAN standards, HAN architectures, HYDRA and Amigo

Connected Vehicles: Introduction, V2X paradigm, VANETs, Intelligent Connected Vehicles

Smart Grids : Introduction and benefits, Architecture, Components, Communication in Smart Grids, Security issues, Use of cloud computing in Smart Grids.

Industrial Internet of Things (IIoT), Data Handling and Analytics

Case Study: Agriculture, Healthcare, Activity Monitoring

TEXT BOOKS

1. "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)
2. "Internet of Things: A Hands-on Approach", by ArshdeepBahga and VijayMadisetti (Universities Press)
3. Research papers

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. V Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05517 OBJECT ORIENTED ANALYSIS AND DESIGN LAB

Course Objectives:

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

UML diagrams to be developed are:

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

(w.e.f 2015-16)

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

JNTUA College Of Engineering (Autonomous):: Anantapuram
Department of Computer Science & Engineering

MCA. V Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05518 Cloud Computing Lab Programs

1. Write a program to print “Hello World” using Aneka Thread Programming model use Single Thread.
2. Write a program to print “Hello World” based in thread model and use exactly five threads also print the executor node information along with the submission time and completion time.
3. Write a program to print “Hello World” using Aneka Thread Programming model and conventional thread and understand the differences.
4. Write a program to compute the following mathematical equation using Aneka Threads (Note: Consider each trigonometric function in independent thread)?

   $$P= \sin(x) + \cos(y) + \tan(z).$$
5. Write a program to print “Hello World” using Aneka Task Programming model.
6. Write a program to sum the two numbers using Aneka Task Programming model.

7. Write a program to print “Hello World” using Aneka Thread Programming model use Five Threads , also print the Node Ids on which the threads are executed and submission time and Completion Time of the Threads.

JNTUA College Of Engineering (Autonomous):: Anantapuram

Department of Computer Science & Engineering

MCA. V Sem.	**T**	**P**	**C**
	0	**4**	**2**

15F05519 Data Warehousing & Mining Lab.

Task 1: Credit Risk Assessment

Description:

The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

(w.e.f 2015-16)

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

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

A few notes on the German dataset
• DM stands for Deutsche Mark, the UNIT of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
• owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
• foreign_worker. There are millions of these in Germany (many from Turrkey). It is very hard to get German citizenship if you were not born of German parents.
• There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into one of two categories, good or bad.

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

1.List all the categorical (or nominal) attributes and the real-valued attributes seperately. (5 marks)

2.What attributes do you think might be crucial in making the credit assessement ? Come up with some simple rules in plain English using your selected attributes. (5 marks)

3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training. (10 marks)

4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly ? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy ? (10 marks)

(w.e.f 2015-16)

5. Is testing on the training set as you did above a good idea ? Why orWhy not ? (10 marks)

6. One approach for solving the problem encountered in the previous question is using cross-validation ? Describe what is cross-validation briefly. Train a Decistion Tree again using cross-validation and report your results. Does your accuracy increase/decrease ? Why ? (10 marks)

7. Check to see if the data shows a bias against "foreign workers" (attribute 20),or "personal-status" (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka's GUI Explorer. Did removing these attributes have any significant effect? Discuss. (10 marks)

8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.) (10 marks)

9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifcations equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)? (10 marks)

10.Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees ? How does the complexity of a Decision Tree relate to the bias of the model ? (10 marks)

11.You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain ? Also, report your accuracy using the pruned model. Does your accuracy increase ? (10 marks)

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

Task Resources:

- Mentor lecture on Decision Trees
- Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
- Decision Trees (Source: Tan, MSU)
- Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)
- Weka resources:
  - Introduction to Weka (html version) (download ppt version)
  - Download Weka
  - Weka Tutorial
  - ARFF format
  - Using Weka from command line

Task 2: Hospital Management System

Data Warehouse consists Dimension Table and Fact Table.

REMEMBER The following

Dimension

The dimension object (Dimension):

_ Name

_ Attributes (Levels) , with one primary key

_ Hierarchies

One time dimension is must.

About Levels and Hierarchies

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

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

H1: YearL > QuarterL > MonthL > WeekL > DayL

H2: YearL > WeekL > DayL

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

About Unique Key Constraints

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

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

Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

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

MEDICINE (Medicine_Brand_name, Drug_name, Supplier, no_UNITs, Uinit_Price, etc.,)
SUPPLIER :(Supplier_name, Medicine_Brand_name, Address, etc.,)
If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably.
Design the Hospital Management system data warehouse using all schemas. Give the example 4-D cube with assumption names.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
R20 Regulations

MCA I Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	20F05101	Programming & Data Structures	PC	3	0	0	3
2	20F05102	Computer Organization	PC	3	0	0	3
3	20F05103	Database Management Systems	PC	3	0	0	3
4	20F05104	Operating Systems	PC	3	0	0	3
5	20F05105	Mathematical and Statistical Foundations	BS&H	3	1	0	4
6	20F05106	Data Base Management Systems Lab	PC	0	0	3	1.5
7	20F05107	Programming & Data Structures Lab	PC	0	0	4	2
8	20F05108	Operating Systems Lab	PC	0	0	3	1.5
		Total					21

MCA II Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	20F05201	Computer Networks	PC	3	0	0	3
2	20F05202	Business English and soft skills	PC	2	0	0	2
3	20F05203	Object Oriented Programming	PC	3	0	0	3
4	20F05204	Software Engineering	PC	3	0	0	3
5	20F05205	Design and Analysis of Algorithms	PC	3	0	0	3
6	20F05206A 20F05206B 20F05206C 20F05206D	**Program Elective-1** 1. Artificial Intelligence 2. Advanced Unix Programming 3. Data Warehousing and Data mining 4. MOOCS- 1 (NPTEL/SWAYAM)	PC / PE	3	0	0	3
7	20F05207	Object Oriented Programming Lab	PC	0	0	2	1
8	20F05208	Software Engineering lab	PC	0	0	2	1
9	20F05209	Design and Analysis of Algorithms lab	PC	0	0	2	1
10	20F05210	**Program Elective-1 Lab**	PC	0	0	2	1
		Total					21

K. Madhav

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
R20 Regulations

MCA III Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	20F05301	Machine Learning	PC	3	0	0	3
2	20F05302	Web Technologies	PC	3	0	0	3
3	20F05303	Internet of Things	PC	3	0	0	3
4	20F05304A 20F05304B 20F05304C 20F05304D 20F05304E	**Program Elective-2** 1. Cloud Computing 2. Image Processing 3. Principles of Cryptography and Network Security 4. SOFTWARE TESTING 5. MOOCS-2 (NPTEL /SWAYAM)	PE	3	0	0	3
5	20F05305	Mobile Application Development		3	0	0	3
6	20F05306	Mobile Application Development Lab		0	0	2	1
7	20F05307	Web Technologies Lab	PC	0	0	3	1.5
8	20F05308	Machine Learning with Python Lab	PC	0	1	2	2
9	20F05309	Internet of Things Lab	PC	0	0	3	1.5
		Total					**21**

MCA IV Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	20F05401A 20F05401B 20F05401C 20F05401D	**Program Elective-3** 1. Block Chain technologies 2. Deep Learning 3. Design Patterns 4. MOOCs-3 (NPTEL/SWAYAM) 1.Full Stack Technologies 2. Any recommended course	PE	3	0	0	3
2	20F05402A 20F05402B 20F05402C 20F05402D	**Program Elective-4** 1. Big Data Analytics 2. Software Defined Networks 3. Malware Analysis or Bio-Metric Security 4. MOOCs-4 (NPTEL/SWAYAM) 1.Data Science 2. Any recommended course	PE	3	0	0	3
3	20F05403	**Seminar**				4	2

4	20F05404	Project Work/ Dissertation		0	0	18	9
		Total					17

K. Madhavi

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Programming & Data Structures

SEMESTER - I **L-T-P-C: 3-0-0-3**

Course Objectives:
The objective of this course is to explore basic data structures such as stacks and queues, introduce a variety of data structures such as hash tables, search trees, tries, heaps, graphs, sorting and pattern matching algorithms

Course Outcomes (CO): *At the end of the course, student will be able to*

Course Outcomes		Knowledge Level (K)#
CO1	Implement basic programs by using C concepts.	**K1**
CO2	Select the data structures that efficiently model the information in a problem	**K3**
CO3	Assess efficiency trade-offs among different data structure implementations or combinations	**K5**
CO4	Implement and know the application of algorithms for sorting and pattern matching.	**K2**
CO5	Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees	**K6**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2	1	1			1	3	1	1	1
CO2	2	3						2		
CO3	1	3					1	1		
CO4	1	3	2			1	2	2		
CO5	3	3	1	1			2	2	1	2

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT - I :
Introduction to C: Constants and variables, Operators and Expressions, Managing Input and Output operators, Decision making-branching and looping, Arrays,

UNIT-II:
Functions, Structures and Unions, Pointers, File handling in C.

UNIT – III :
Data structure: Definition, types of data structures Recursion Definition, Design Methodology and Implementation of recursive algorithms, Linear and binary recursion. Preliminaries of algorithms, analysis and complexity

.**Linear list** – singly linked list, Double linked list and circular linked list - implementation, insertion, deletion and searching operations on linear list.

UNIT - IV:

Stacks-Operations, array and linked representations of stacks, stack applications, **Queues**-operations, array and linked representations. **Hash Table Representation:** hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing and rehashing, extendible hashing.

UNIT - V:

Sorting Techniques: Insertion sort, selection sort, exchange-bubble sort, quick sort and merge sort Algorithms. **Trees:** Binary Trees, terminology, representation and traversals- pre, post & in order traversals. **Search Trees:** Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion

Text Books:

1. Programming in ANSI C, 5e, E. Balaguruswamy, TMH
2. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and
Susan Anderson Freed, Universities Press.
3. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M.J. Augenstein,
PHI/Pearson Education.

Reference Books:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Computer Organization

SEMESTER - I L-T-P-C: 3-0-0-3

Course Objectives:

The objectives of this course are to

- Conceptualize the basics of organizational and architectural issues of a digital computer.
- Learn the function of each element of a memory hierarchy.
- Study various data transfer techniques in digital computer.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Understand the basic organization of computer and different instruction formats and addressing modes	**K2**
CO2	Analyze the concept of pipelining, segment registers and pin diagram of CPU.	**K4**
CO3	Understand and analyze various issues related to memory hierarchy	**K2**
CO4	Evaluate various modes of data transfer between CPU and I/O devices	**K5**
CO5	Examine various inter connection structures of multi processors	**K4**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1						2	3		
CO2		1	1				2	2		
CO3	1	2								
CO4	2	2	1					2		
CO5		1					1	2		

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT I:

Basic Structure Of Computers: Computer Types, Functional units, Basic Operational concepts, Bus structures, Software, Performance, multiprocessor and multi computers, Historical perspective.

UNIT II:

Machine Instructions and Programs: Numbers, Arithmetic Operations, and c Characters, Memory locations and addresses, Memory operations, Instructions and Instruction sequencing, Addressing Modes, Assembly Languages, stacks and Queues Basic Input/output Operations, role of Stacks and Queues Additional Instructions

UNIT III:

Input/ Output Organization: Accessing I/O Devices, Interrupts, Processor examples, Direct Memory Access, Buses, Interface Circuits, and Standard I/O Interfaces

UNIT IV:

The Memory Systems: Some Basic concepts, Semi conductor RAM memories, Memory System Consideration, Read-Only Memories**,** Speed, Size, and cost, Cache Memories, Performance considerations, Virtual Memories, Memory Management Requirements, Secondary Storage

UNIT V:

Parallel Processing: Basic concepts, Pipeline Processors, Multiprocessors

Text Books:

1. Computer Organization, Carl Hamacher, Zvonks Vranesic, Safea Zaky, 5th Edition, McGraw Hill.
2. Computer Architecture and Organization , John P. Hayes ,3rd Edition, McGraw Hill

Reference Books:

1. Computer Organization and Architecture, William Stallings Sixth Edition, Pearson/PHI
2. Structured Computer Organization, Andrew S. Tanenbaum, 4th Edition PHI/Pearson
3. Fundamentals or Computer Organization and Design, Sivarama Dandamudi Springer Int. Edition.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Database Management Systems

SEMESTER - I L-T-P-C: 3-0-0-3

Course Objectives:

This Course will enable students to

- Explain the concept of databases, database management systems, database structures and how they work.
- Make use of Entity-Relationship Modeling and Relational Modeling for creating simple databases from the real world scenarios.
- Write relational algebra and structured query language (SQL) statements.
- Normalize a database using Normalization Rules.
- Discuss the issues associated with Transaction Management and Recovery, Tree Structured and Hash-Based Indexing

Course Outcomes(COs): At the end of the course the student will be able to:

CO	Course Outcomes	Knowledge Level (K)#
CO1	Illustrate the concept of databases, database management systems, database languages, database structures and their work	K2
CO2	Apply ER modeling and Relational modeling for designing simple databases.	K3
CO3	Summarize the concepts related to relational model and SQL and Write database queries using relational algebra and structured query language.	K2
CO4	Design and develop databases from the real world by applying the concepts of Normalization.	K6
CO5	Outline the issues associated with Transaction Management and Recovery, Tree Structured and Hash-Based Indexing	K2

#Based on suggested revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	2	1	1		1	2			
CO2	1	2	3				2	3		
CO3	2	3	3	2		2	2	2		
CO4	2	3	3	2		2	3	3	3	
CO5	2	2	3	2		2	3	3		

(Levels of Correlation: 1-low, 2-medium 3-high)

Unit-I:

Introduction to Databases: Introduction, An Example, Characteristics of the Database Approach, Actors on Scene, Workers behind the scene, Advantages of Using the DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS **[TB-3]**

Overview of Database Languages and Architectures: Data Models, Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client/Server Architecture for DBMSs, Classification of Database Management Systems **[TB-3]**

Unit-II:

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model, Conceptual Design for Large Enterprises **[TB-1]**

Relational Model: Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Querying Relational Data, Logical Database Design: ER to Relational, Introduction to Views, Destroying/Altering Tables and Views **[TB-1]**

Unit-III:

Relational Algebra: Selection and Projection, Set Operations, Renaming, Joins, Division, More Examples of Algebra Queries **[TB-1]**

SQL: Queries, Constraints, Triggers: The Form of a Basic SQL Query, UNION, INTERSECT and EXCEPT, Nested Queries, Aggregate Operators, Null Values, Complex Integrity Constraints in SQL, Triggers and Active Databases, Designing Active Databases **[TB-1]**

Unit-IV:

Introduction to Normalization Using Functional and Multivalued Dependencies:

Informal Design Guidelines for Relation Schema, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form **[TB-3]**

Unit-V:

Transaction Management and Concurrency Control: Transaction Concept, A Simple Transaction Model, Storage Structure, ACID Properties, Serializability, Transaction Isolation Levels, Concurrency Control, Lock-Based Protocols, Validation-Based Protocols **[TB-2]**

Note: For Practical Examples Please Go Through Reference 1

Text Books:

1. Data base Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, Mc Graw-Hill
2. Data base System Concepts, 6/e, Abraham Silberschatz, Henry F. Korth, S.
   Sudarshan, Mc Graw-Hill
3. Database Systems, 6/e Ramez Elmasri, Shamkant B. Navathe, Pearson

Reference Books:

1. Database Systems, 9/e, Carlos Coronel, Steven Morris, Peter Rob, Cengage
2. Introduction to Database Systems, 8/e, C J Date, Pearson

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Operating Systems

SEMESTER - I L-T-P-C: 3-0-0-3

Course Objectives:
This course enables the student to

- Introduce different types of operating systems.
- Learn process management techniques.
- Learn various memory management techniques.
- Introduce the architecture of Linux operating system.
- Learn multiple operating system like Unix and Windows.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Understand the basics of operating systems like kernel, shell, types and views of operating systems	**K2**
CO2	Understands CPU scheduling algorithms and compare the results using Gantt chart.	**K5**
CO3	Explain various memory management techniques and concept of thrashing	**K2**
CO4	Apply disk scheduling algorithms for better utilization of external memory	**K3**
CO5	Understand the architecture of UNIX operating system	**K1**
CO6	Write and execute shell programs	**K1**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2	1								
CO2	1	2	3		2		2			
CO3	2	2	3			1		1		
CO4	1	2	3					2	2	
CO5	1		2							
CO6	2	3	2				2	2		

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:
Introduction to Operating System Concept: Types of Operating Systems, Operating Systems Concepts, Operating System Operations. Operating Systems Structures- Operating System Services, User Operating-System Interface, Introduction to System calls, Types of System Calls.

UNIT–II:
Process Management: Process concept, Process State Diagram, Process control block, Process Scheduling, Inter process Communication, Threads-Threading Issues, Scheduling- Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

UNIT-III:

Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, **Principles of deadlock:** System Model, Deadlock characterization, Deadlock handling, Deadlock Prevention, Detection and Avoidance, Recovery Starvation, Critical Regions form Deadlock

UNIT–IV:

Memory Management: Swapping, Contiguous Memory Allocation, Paging, structure of the Page Table, Segmentation Virtual Memory Management-Demand Paging, Page-Replacement Algorithms, Thrashing. **File-System Interface:** File Concept, Access Methods, Directory structure, File-System mounting, Files Sharing, Protection. File-System implementation- File-System Structure, Allocation Methods, Free-Space Management, Disk Structure, Disk Scheduling

UNIT-V:

Case Studies: Linux System: Design Principles, kernel Modules, Process Management, File Systems, Input and Output, Interprocess Communication, Network Structure, Security. **Windows7:** Design Principles, System Components, Terminal Services and Fast User, File System, Networking, Programmer Interface.

Text Books:

1. Operating system concepts, Abraham Siliberschatz, Peter Baer Galvin, Greg Gagne, John Wiley & Sons, Inc., Edition 9, 2011
2. Introduction to UNIX and Shell Programming, M. G. Venkateshmurhty, Pearson, 2005
3. UNIX & Shell Programming by B.M. Harwani, OXFORD University Press, 2013

Reference Books:

1. Advanced Programming in the UNIX Environment by W. Richard Stevens, Stephen Rago, Wesley Professional, 2013
2. UNIX Network Programming by W. Richard Stevens, 1990
3. Operating systems, William stallings, PHI/Pearson, 6/E, 2009
4. Operating systems, Dietal, Dietal, Pearson, 3/e, 2007
5. Operating systems, Dhamdhere, TMH, 2/e, 2009

Web Reference:
https://onlinecourses.swayam2.ac.in/cec20_cs06/preview

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Mathematical and Statistical Foundations

SEMESTER - I L-T-P-C: 3-1-0-4

Course Objectives: This course is aimed at enabling the students to

- To understand the mathematical fundamentals that is prerequisites for variety of courses like Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems bioinformatics, Machine learning.
- To develop the understanding of the mathematical and logical basis to many modern techniques in computer science technology like machine learning, programming language design, and concurrency.
- To study various sampling and classification problems.

Course Outcomes:

After the completion of the course, student will be able to

CO	Course Outcomes	Knowledge Level (K)#
CO1	To apply the basic rules and theorems of probability theory such as Baye's Theorem, to determine probabilities that help to solve engineering problems and to determine the expectation and variance of a random variable from its distribution.	**K3**
CO2	Able to perform and analyze of sampling, means, proportions, variances and estimates the maximum likelihood based on population parameters.	**K4**
CO3	To learn how to formulate and test hypotheses about sample means, variances and proportions and to draw conclusions based on the results of statistical tests.	**K6**
CO4	Design various ciphers using number theory.	**K6**
CO5	Apply graph theory for real time problems like network routing problem.	**K3**

# based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	2	1		2		2	2	3	
CO2	3	2	3		1		1	2	1	
CO3	3	2	1	2	3					
CO4	2	3	1				1			
CO5	3	2	2							

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT I:

Basic Probability and Random Variables: Random Experiments, Sample Spaces Events, the Concept of Probability the Axioms of Probability, Some Important Theorems on Probability Assignment of Probabilities, Conditional Probability Theorems on Conditional Probability, Independent Events, Bayes Theorem or Rule. Random Variables, Discrete Probability Distributions, Distribution Functions for Random Variables, Distribution Functions for Discrete Random Variables, Continuous Random Variables

UNIT II:

Sampling and Estimation Theory: Population and Sample, Statistical Inference Sampling With and Without Replacement Random Samples, Random Numbers Population Parameters Sample Statistics Sampling Distributions, Frequency Distributions, Relative Frequency Distributions, Computation of Mean, Variance, and Moments for Grouped Data. Unbiased Estimates and Efficient Estimates Point Estimates and Interval Estimates. Reliability Confidence Interval Estimates of Population Parameters, Maximum Likelihood Estimates

UNIT III:

Tests of Hypothesis and Significance: Statistical Decisions Statistical Hypotheses. Null Hypotheses Tests of Hypotheses and Significance Type I and Type II Errors Level of Significance Tests Involving the Normal Distribution One-Tailed and Two-Tailed Tests P Value Special Tests of Significance for Large Samples Special Tests of Significance for Small Samples Relationship between Estimation Theory and Hypothesis Testing Operating Characteristic Curves. Power of a Test Quality Control Charts Fitting Theoretical Distributions to Sample Frequency Distributions, The Chi-Square Test for Goodness of Fit Contingency Tables Yates' Correction for Continuity Coefficient of Contingency.

UNIT IV:

Algebraic Structures and Number Theory: Algebraic Systems, Examples, General Properties, Semi Groups and Monoids, Homomorphism of Semi Groups and Monoids, Group, Subgroup, Abelian Group, Homomorphism, Isomorphism. Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat's Theorem and Euler's Theorem)

UNIT V:

Graph Theory: Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multigraphs, Planar Graphs, Euler's Formula, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Algorithms for Spanning Trees (Problems Only and Theorems withou*t* Proofs).

Text Books:

1. Foundation Mathematics for Computer Science, 1st Edition, John Vince, Springer,2015
2. Probability & Statistics, 3rd Edition, Murray R. Spiegel, John J. Schiller and
   R. Alu Srinivasan, Schaum's Outline Series, Tata McGraw-Hill Publishers, 2018
3. Probability and Statistics with Reliability,2nd Edition, K. Trivedi, Wiley, 2011
4. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, 7th Edition, H. Rosen, Tata McGraw Hill, 2003

Reference Books:

1. Probability and Computing: Randomized Algorithms and Probabilistic Analysis,1st Edition, M. Mitzenmacher and E. Upfal,2005
2. Applied Combinatorics,6th Edition, Alan Tucker, Wiley,2012

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Data Base Management Systems Lab

SEMESTER - I **L-T-P-C: 0-0-3-1.5**

Course Objectives:

This Course will enable students to

- Populate and query a database using SQL DDL/DML Commands.
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming PL/SQL including procedures, functions, cursors and triggers

Course Outcomes:

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

CO	Course Outcomes	Knowledge Level (K)#
CO1	Utilize SQL to execute queries for creating database and performing data manipulation operations	K3
CO2	Examine integrity constraints to build efficient databases	K4
CO3	Apply Queries using Advanced Concepts of SQL	K3
CO4	Build PL/SQL programs including stored procedures, functions, cursors and triggers.	K6

#Based on suggested revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	2	3				1	1		
CO2	1	2	3	1			2			
CO3	1	2	3				2	3		
CO4		3	3				3	3		

(Levels of Correlation: 1-low, 2-medium 3-high)

1. Execute all DDL, DML and DCL commands on sample tables.
2. Implementation of different types of operators and built-in functions with suitable examples
3. Implementation of different types of joins with suitable examples
4. Create views, partitions, Sequence, Indexes and locks for a particular DB
5. Implement different types of constraints on relations.
6. Implementation of sub queries and nested queries.
7. Implement Queries on Group By & Having Clauses, ALIAS, Sequence By, Order By
8. Control Structure
   a) Write a PL/SQL block for Addition of Two Numbers
   b) Write a PL/SQL block for IF, IF and else condition
   c) Write a PL/SQL block for implementation of loops
   d) Write a PL/SQL block for greatest of three numbers using IF ANDELSEIF

9. Exception Handling- Implement the following with respect to exception handling.

Raising Exceptions, User Defined Exceptions, Pre-Defined Exceptions

10. Write PL/SQL block for an application using exception handling

10. *Procedures*
    - a) Write a PL/SQL Procedure using Positional Parameters
    - b) Write a PL/SQL Procedure using notational parameters
    - c) Write a PL/SQL Procedure for GCD Numbers
    - d) Write a PL/SQL Procedures for cursor implementation (explicit and implicit cursors)

11. *Functions:*
    - a) Write a PL/SQL block to implement factorial using functions
    - b) Write a PL/SQL function to search an address from the given database

12. Write a DBMS program to prepare Pl/SQL reports for an application using functions.

13. Triggers:
    - a) Write a Trigger to pop-up the DML operations
    - b) Write a Trigger to check the age valid or not Using Message Alert.
    - c) Create a Trigger to Raise appropriate error code and error message.
    - d) Create a Trigger on a table so that it will update another table while inserting values

14. Write PL/SQL block for an application using cursors and all types of triggers.

15. Write a PL/SQL block for transaction operations of a typical application using package

Text Books / Suggested Readings:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, “Introduction to SQL”, Fourth Edition, Pearson Education, 2007

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Programming & Data Structures Lab

SEMESTER - I L-T-P-C: 0-0-4-2

Course Objectives: This Course will enable students to

- Design and implement various data structures.
- Implement operations like searching, insertion, and deletion, traversing mechanism
- Develop applications using data structure algorithms.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Implement various basic data structures and its operations.	**K2**
CO2	Apply sorting and searching algorithms to given numbers	**K3**
CO3	Implement various tree operations.	**K2**
CO4	Implement various graphs algorithms.	**K2**
CO5	Develop applications using various data structures.	**K6**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2	3	1				1	1		
CO2	2	3	2				2	2		
CO3	2	3	3			2	1	2	2	
CO4	2	2	3			2	1	2	2	
CO5	2	3	3	1		2	3	2	1	

(Levels of Correlation: 1-low, 2-medium 3-high)

Experiment 1:

a) Write a program in C to display the n terms of even natural number and their sum.
b) Write a program in C to display the n terms of harmonic series and their sum. 1 + 1/2 + 1/3 + 1/4 + 1/5 … 1/n terms.
c) Write a C program to check whether a given number is an Armstrong number or not.
d) Write a C program to calculate the factorial of a given number.

Experiment 2:

a) Write a program in C for multiplication of two square Matrices.
b) Write a program in C to find transpose of a given matrix.

Experiment 3:

a) Write a program in C to check whether a number is a prime number or not using the function.
b) Write recursive program which computes the n^{th} Fibonacci number, for appropriate values of n.

c) Write a program in C to add numbers using call by reference.

Experiment 4:

a) Write a program in C to append multiple lines at the end of a text file.
b) Write a program in C to copy a file in another name.

Experiment 5:

Write recursive program for the following

a) Write recursive and non recursive C program for calculation of Factorial of an integer.
b) Write recursive and non recursive C program for calculation of GCD (n, m)
c) Write recursive and non recursive C program for Towers of Hanoi: N disks are to be transferred from peg S to peg D with Peg I as the intermediate peg.

Experiment 6:

a) Write C program that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.
b) Write C program that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.

Experiment 7:

a) Write C program that implement stack (its operations) using arrays.
b) Write C program that implement stack (its operations) using Linked list.

Experiment 8:

a) Write a C program that uses Stack operations to convert infix expression into postfix expression.
a) Write C program that implement Queue (its operations) using arrays.
b) Write C program that implement Queue (its operations) using linked lists.

Experiment 9:

Write a C program that uses functions to create a singly linked list and perform various operations on it.

Experiment 10:

Write a C program to store a polynomial expression in memory using linked list and perform polynomial addition.

Experiment 11:

a) Write a recursive C program for traversing a binary tree in preorder, inorder and postorder.
b) Write a non recursive C program for traversing a binary tree in preorder, inorder and postorder.

Experiment 12:

a) Write a C program to implement Prims' algorithm.
b) Write a C program to implement Kruskal's algorithm.

Experiment 13:

Implementation of Hash table using double hashing as collision resolution function.

Experiment 14:

Implementation of Binary Search trees- Insertion and deletion.

Experiment 15:
Implementation of AVL Tree – Insertion and Deletion

Experiment 16:

a) Write C program that implement Bubble sort, to sort a given list of integers in ascending order.

b) Write C program that implement Quick sort, to sort a given list of integers in ascending order.

c) Write C program that implement merge sort, to sort a given list of integers in ascending order

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Operating Systems Lab

SEMESTER - I L-T-P-C: 0-0-3-1.5

Course Objectives:

This Course will enable students to implement CPU scheduling algorithms, Disk scheduling algorithms, Execute different types of Linux commands and Write shell scripts

Course Outcomes(COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Implement various CPU scheduling algorithms and compare results	**K5**
CO2	Implement various disk scheduling algorithms and compare results	**K5**
CO3	Implement page replace algorithms	**K2**
CO4	Implement various memory management techniques.	**K2**
CO5	Execute basic Linux commands	**K1**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	2	3		2		2	3	2	
CO2	3	2	3					2	2	
CO3	2	2	3			1		2	1	
CO4	2	2	3			1		1		
CO5	2	3	2				2	2		

(Levels of Correlation: 1-low, 2-medium 3-high)

List of Experiments:

UNIX Lab- Introduction to Unix

1. Study of Unix/Linux general purpose utility commands
2. Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system .
3. Study of UNIX/LINUX File System(tree structure).
4. C program to emulate the UNIX ls –l command
5. C program that illustrates how to execute two commands concurrently with a command pipe. Ex: - ls –l | sort
6. Multiprogramming-Memory management-Implementation of fork (), wait (), exec() and exit (), System calls

Operating Systems Lab

1. Simulate the Following CPU Scheduling Algorithms
   A) FCFS B) SJF C) Priority D) Round Robin
2. Multiprogramming-Memory Management- Implementation of fork(), wait(), exec() and exit()

3. Simulate The Following
   a. Multiprogramming with A Fixed Number Of Tasks (MFT
   b. Multiprogramming with A Variable Number Of Tasks (MVT)
4. Write a program to implement first fit, best fit and worst fit algorithm for memory management.
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention.
7. Simulate The Following Page Replacement Algorithms.
   a) FIFO b) LRU c) LFU
8. Simulate the Following File Allocation Strategies
   a) Sequenced b) Indexed c) Linked

Linux Lab

1. Write a Shell program to check whether given number is prime or not.
2. Write a shell script which will display Fibonacci series up to the given range.
3. Write a shell script to check whether the given number is Armstrong or not.
4. Write a shell script to the calculate the value of
5. Write a shell script to accept student number, name, marks in 5 subjects.
6. Find total, average and grade using the following rules:
   Avg>=80 then grade A
   Avg<80&&Avg>=70 then grade B
   Avg<70&&Avg>=60 then grade C
   Avg<60&&Avg>=50 then grade D
   Avg<50&&Avg>=40 then grade E
7. Write a shell script to find minimum and maximum elements in the given list of elements.
8. Write a shell program to check whether the given string is palindrome or not.
9. Write an awk program to print sum, avg of students marks list
10. Write a shell script to compute no. of characters and words in each line of given file
11. Write a shell script to check whether the given input is a number or a string

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Computer Networks

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

At the end of the course, the students will be able to:

- To Understands the fundamental concepts of computer networking and OSI Reference model.
- To Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- To learn and understand the advanced networking concepts, preparing the student for entry advanced courses in computer networking.
- To develop and gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Explain the network architecture, TCP/IP and OSI reference models	**K2**
CO2	Identify and understand various techniques and modes of transmission	**K3**
CO3	Demonstrate the data link protocols, multi-channel access protocols and IEEE 802 standards for LAN	**K2**
CO4	Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme	**K5**
CO5	Discuss the elements and protocols of transport layer	**K6**
CO6	Develop network security and define various protocols such as FTP, HTTP, Telnet, DNS	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	P 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		1			1			1		
CO2	1							2		1
CO3		1	1						2	
CO4	1				2					
CO5										
CO6		2						1		

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT – I

Introduction: Network Topologies WAN, LAN, MAN. Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models.**Physical Layer –**Introduction to physical layer-Data and Signals, Periodic analog signals, digital signals, transmission impairment, ,Data rate limits, performance - Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and Unguided media: Wireless-Radio waves, microwaves, infrared.

Unit-II

The Data Link Layer - Services Provided to the Network Layer – Framing – Error Control – Flow Control, Error Detection and Correction – Error-Correcting Codes – Error Detecting Codes. **Elementary Data Link Protocols-** A Utopian Simplex Protocol-A Simplex Stop and

Wait Protocol for an Error free channel-A Simplex Stop and Wait Protocol for a Noisy Channel, Sliding Window Protocols-A One Bit Sliding Window Protocol-A Protocol Using Go-Back-N- A Protocol Using Selective Repeat.

UNIT-III

The Medium Access Control Sub layer-The Channel Allocation Problem-Static Channel Allocation-Assumptions for Dynamic Channel Allocation, Multiple Access Protocols-Aloha-Pure aloha- slotted aloha-Carrier Sense Multiple Access Protocols- Collision-Free Protocols-Limited Contention Protocols. **Wireless LAN Protocols-** Ethernet-Classic Ethernet Physical Layer-Classic Ethernet MAC Sub-layer Protocol-Ethernet Performance-Fast Ethernet-Wireless LANs-The 802.11 Architecture and Protocol Stack-The 802.11 Physical Layer-The802.11 MAC Sub-layer Protocol- The 805.11 Frame Structure-Services.

Unit-IV

The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks, Routing Algorithms-The Optimality principle-Shortest path, Flooding, Distance vector, Link state, Hierarchical. **Congestion Control algorithms-**General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing- Admission Control-Traffic Throttling-Load Shedding. **Internet Working:** How networks differ- How networks can be connected- Tunneling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version 4 protocol-, IP addresses-, Subnets-IP Version 6-The main IPV6 header- Internet control protocols- ICMP-ARP-DHCP.

UNIT-V

The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services- TCP features- Segment- A TCP connection- windows in TCP- flow control-Error control. **Application Layer –-** World Wide Web: HTTP , FTP-Two connections-control connection-Data connection-security of FTP-Electronic mail-Architecture- web based mail- email security- TELENET-local versus remote Logging. **Domain Name System:** Name Space, DNS in Internet, - Resolution-Caching- Resource Records- DNS messages-Registrars-security of DNS Name Servers.

Text Books:

1. Computer Networks: Andrew S Tanenbaum David J. Wetherall, 5/e, Pearson
2. Data communications and networking: Behrouz Forouzan, 5/e, McGraw Hill

Reference Books

1. Computer Networks – A System Approach, Peterson, Bruce Davie,2/e , Harcourt Asia
2. Compute communications and networking technologies, Gallo, Hancock, Cengage

An Engineering approach to compute networking, Kesha, Pearson

Business English and Soft Skills

SEMESTER - II **L-T-P-C: 2-0-0-2**

INTRODUCTION:

English is an international language as well as a living and vibrant one. People have found that knowledge of English is a passport for better career, better pay, advanced knowledge and for communication with the entire world. As it is a language of opportunities in this global age, English is bound to expand its domain of use everywhere. The syllabus has been designed to enhance communication skills of the students of professional courses. The prescribed books serve the purpose of preparing them for everyday communication and to face the global competitions in future.

The prescribed books for detailed study focus on soft skills and Business English Communication. The teachers should encourage the students to use the target language in business scenario. The classes should be interactive and student-centered. They should be encouraged to participate in the classroom activities keenly. The teacher can bring variety by using authentic materials such as newspaper articles, advertisements, promotional material etc.

OBJECTIVES:

1. To enable the students to communicate in English for business purpose.
2. To enhance presentation skills of the students required to persuade the target customers.
3. To develop the business writing skills of the students
4. To enhance the soft skills of the students in business setting.
5. To improve employability and negotiation skills necessary in business world.

Unit I

Introduction to Essentials of Business English -Importance of Business English -Definition– Types &Channels of Business English – Cultural barriers in Business English. Business etiquettes& Manners.

Unit II

Delivering Effective speech- How to make first impression -Managing questions and answering. Speaking at meetings. Business Presentations– Body language-Developing self confidence.

Unit III

Business report writing - Business Letter Writing– Writing Effective Resume and covering letters

Unit IV

Goal setting-Time Management- Leadership Skills– Team Work - Team building- Art of Negotiation– Emotional Intelligence- Logical Thinking.

Unit V

Employability Skills- Interview Skills- Group Discussions–Video Conferencing

EXPECTED OUTCOME:

The students will be able to

- Define types of Business English and its barriers
- Discuss the different aspects of business communication
- Apply the Soft Skills in business scenario

- Analyze different ideas in Group Discussions
- Appraise ones strengths and weaknesses to project a good image in an interview
- Develop a resume suitable to the requirements of the job

Prescribed Books:

1. Soft Skills - Key to Success in Workplace and Life by Meenakshi Raman and Shalini Upadhyay, Cengage, 2018
2. Business English: A Complete Guide for All Business and Professional Communications by PREM P.BHALLA, V&S Publishers, 2018 edition

SUGGESTED READING:

1. Personality Develpment and Soft Skills by Barun K. Mitra, Oxford, 2016.
2. English Conversation Practice –Grant Taylor, Tata McGraw Hill,2009.
3. A Course in Communication Skills- Kiranmai Dutt & co. Foundation Books, 2012.
4. Writing with a Purpose, Tickoo and Sasi Kumar, OUP, 2011
5. Strengthen Your Writing, Orient Blackswan

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Object Oriented Programming

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To understand the basic concepts of object oriented programming concepts.
- To introduce the principles of inheritance and polymorphism and demonstrate how they are related to the design of abstract classes
- To understand the implementation of packages and interfaces
- To introduce the concept of multithreading and exception handling
- To learn and understand the design of Graphical User Interface using applets and swing controls

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Describe the uses OOP concepts	**K2**
CO2	Apply OOP concepts to solve real world problems	**K3**
CO3	Distinguish the concept of packages and interfaces	**K4**
CO4	Demonstrate the exception handing, multithread applications with synchronization	**K2**
CO5	Design the GUI based applications using AWT and Swings	**K6**
CO6	Discuss the Collection Framework	**K6**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1				1					
CO2		2						1	1	
CO3	1									
CO4		2	1						1	
CO5	1					1				

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Basics of Object Oriented Programming (OOP): Need for OO paradigm , A way of viewing world- Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, overriding and exceptions, summary of OOP concepts, coping with complexity, abstraction mechanisms. **Java Basics:** Data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and costing, simple java program, classes and objects- concepts of classes, objects, constructors methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling.

UNIT–II:

Inheritance: Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism, abstract classes. **Packages and Interfaces:** Defining, Creating and Accessing a package, Understanding CLASSPATH, Importing packages, differences between classes and interfaces, defining an interface, Implementing interface, applying interfaces variables in interface and extending interfaces.

UNIT–III:

Exception handling and Multithreading: Concepts of exception handling, benefits of exception handling, Termination or presumptive models, exception hierarchy, usage of try, catch, throws and finally, built in exceptions, creating own exception sub classes. Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.

UNIT–IV:

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. The AWT class hierarchy , user-interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, list panes- scroll pane, dialogs, menu bar, graphics, layout manager- layout manager types- boarder, grid, flow, card and grid bag.

UNIT-V:

Applets: Concepts of Applets, differences between applets and applications, lifecycle of an applet, types of applets, creating applets, passing parameters to applets, **Swings:** Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons-The JButton class, Check boxes, Radio Buttons, Combo boxes, Tabbed panes, Scroll panes, Trees and Tables.

Text Books:

1. Java-The complete reference,7/e, Herbert schildt, TMH
2. JAVA: How to program, 8/e, Dietal , Dietal,PHI
3. Introduction of programming with JAVA,S.Dean,TMH
4. Introduction to Java programming, 6/e, Y.Daniel Liang, Pearson

Reference Books:

1. Core Java 2, Vol 1(Vol 2) Fundamentals(Advanced), 7/e, Cay.S.Horstmann, Gary Cornell, Pearson

Big Java2,3/e, Cay.S. Horstmann, Wiley

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Software Engineering

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To understand the nature of software development and software life cycle models.
- To understand methods of capturing, specifying, visualizing and analyzing software requirements.
- To know basics of testing and understanding concept of software quality assurance and software configuration management process.
- To learn to provide correctness proofs for algorithms.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Analyze software application domains and process models used in software development.	**K4**
CO2	Explain the software requirements collection and develop specifications and evaluate them.	**K2**
CO3	Convert the requirements model into the design model and evaluate the complexity metrics.	**K5**
CO4	Compare various testing strategies and tactics and their applications with the supporting tools.	**K4**
CO5	Adopt the activities of Software Project Development principles in project development.	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		2								
CO2	1	1			1					
CO3		1					1			
CO4			3						1	1
CO5				2						1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Introduction: Software Engineering and its history, Software crisis, Evolving of a Programming System Product, Characteristics of Software, Brooks' No Silver Bullet, and Software Myths. **Software Development Life Cycles :** Software Development Process, The Code-and-Fix model, The Waterfall model, The Evolutionary Model, The Incremental Implementation, Prototyping, The Spiral Model, Software Reuse, Critical Comparisons of SDLC models. **An Introduction to Non-Traditional Software Development Process:** Rational Unified Process, Rapid Application Development, Agile Development Process.

UNIT–II:

Requirements: Importance of Requirement Analysis, User Needs, Software Features and Software Requirements. **Classes of User Requirements :** Enduring and Volatile, Sub phases of Requirement Analysis, Functional and Non-functional requirements, Barriers to Eliciting User requirements, The software requirements document and SRS standards, Requirements Engineering, Case Study of SRS for a Real Time System. **Tools for Requirements Gathering:** Document Flow Chart, Decision Table, Decision Tree, Introduction to non-traditional Requirements.

UNIT- III:

Software Design: Goals of good software design, Design strategies and methodologies, Data oriented software design. **Structured Design:** Structure chart, Coupling, Cohesion, Modular structure, Packaging, Object oriented design, Top-down and bottom-up approach, Design patterns. **Structured Analysis:** DFD, Data Dictionary, Software Measurement and Metrics : Various Size Oriented Measures : Halstead's software science, Function Point (FP) based measures, Cyclomatic Complexity Measures : Control flow graphs Development : Selecting a language, Coding guidelines, Writing code, Code documentation.

UNIT– IV:

Software Testing : Testing process, Design of test cases, Functional Testing : Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path testing, Data flow and mutation testing, Unit testing, Integration and system testing, Debugging, Alpha & beta testing, testing tools & standards.

UNIT-V:

Software Maintenance: Management of maintenance, Maintenance process, Maintenance models, Regression testing, Reverse engineering, Software reengineering, Configuration management, documentation.

Text Books:

1. Software Engineering: A Practitioner's Approach by R. S. Pressman, McGraw Hill, 9th Edition, Sept 2019

Reference Books:

1. Zero Defect Software, G. G. Schulmeyer, Published by McGraw Hill, 1992
2. Object Oriented Modeling and Design, J. Rumbaugh, Published by Prentice Hall, 1991
3. Software Engineering K.K. Aggarwal, Yogesh Singh, Published by New Age International Publishers, Third Edition, 2007
4. Software Engineering , Ian Sommerville, Published by Addison Welsley, 9th Edition, 2010.

1. An Integrated Approach to Software Engineering, Pankaj Jalote, Published by Narosa Publishing House, 3rd Edition, 2007Data base System Concepts, 6/e, Abraham Silberschatz, Henry F. Korth, S.
   Sudarshan, Mc Graw-Hill
2. Database Systems, 6/e Ramez Elmasri, Shamkant B. Navathe, Pearson

Reference Books:

1. Database Systems, 9/e, Carlos Coronel, Steven Morris, Peter Rob, Cengage
2. Introduction to Database Systems, 8/e, C J Date, Pearson

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Design and Analysis of Algorithms

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To analyze the asymptotic performance of algorithms.
- To understand the write rigorous correctness proofs for algorithms.
- To familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Course Outcomes (COs): At the end of the course, student will be able to

CO	Course Outcomes	Knowledge Level (K)#
CO1	Explain the basic concepts of time and space complexity,	**K2**
CO2	Explain the basic concepts of divide-and-conquer Strategy, dynamic programming, greedy and approximate algorithm	**K3**
CO3	Describe the methodologies of how to analyze the following applications by Dynamic Programming algorithm	**K5**
CO4	Discuss the concept of graph coloring and back tracking	**K6**
CO5	Analyze the performance of algorithms	**K4**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3				1			1		
CO2				1						
CO3							1			
CO4	1								1	
CO5							2			

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

Introduction: Algorithm, Pseudo code for expressing algorithms, performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, probabilistic analysis, Amortized analysis. Disjoint Sets- disjoint set operations, union and find algorithms, spanning trees, connected components and bi- connected components.

UNIT-II:

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Stassen's matrix multiplication. Greedy method: General method, applications-Job sequencing with deadlines, 0/1 knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT-III:

Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

UNIT-IV:

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT-V:

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

Text Books:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press
2. The Algorithm Design Manual, 2nd edition, Steven S. Skiena, Springer
3. Introduction to Algorithms, second edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd

Reference Books:

1. Introduction to the Design and Analysis of Algorithms, Anany Levitin, PEA
2. Design and Analysis of Algorithms, Pearson Education, Parag Himanshu Dave, Himansu Balachandra Dave
3. Introduction to Design and Analysis of Algorithms A strategic approach, R.C.T. Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc GrawHill.
4. Design and Analysis of algorithms, Pearson education, Aho, Ullman and Hopcroft.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-1 Artificial Intelligence

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To learn the basic State space representation. Intelligent Systems Categorization of Intelligent concepts and techniques of AI and machine learning
- To explore the various mechanism of Knowledge and Reasoning used for building expert system.
- To become familiar with supervised and unsupervised learning models
- To design and develop AI and machine learning solution using modern tools.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents	**K6**
CO2	Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.	**K5**
CO3	Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing	**K6**
CO4	Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.	**K1**
CO5	Solve problems with uncertain information using Bayesian approaches.	**K3**

- *#Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2							1		
CO2					1	1		2		
CO3		2			1				1	1
CO4	2							1		
CO5					1		1			

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT– I

Introduction to AI- Definition, Problem, System, Components of AI Program, Foundations of AI, Applications of AI, Current trends in AI, Intelligent Agents: Anatomy, structure, Types.

UNIT– II

Problem solving-Solving problem by Searching: Problem Solving Agent, Formulating Problems. Uninformed Search Methods: Breadth First Search (BFS), Depth First Search (DFS), Depth Limited Search, Depth First Iterative Deepening (DFID), Informed Search Methods- Greedy best first Search, A* Search, Memory bounded heuristic Search. Local Search Algorithms and Optimization Problems- Hill climbing search Simulated annealing and local beam search.

UNIT – III

Knowledge and Reasoning-Knowledge based Agents, The Wumpus World, and Propositional logic**. First Order Logic**- Syntax and Semantic, Inference in FOL, Forward chaining, backward Chaining, Knowledge Engineering in First-Order Logic, Unification and Resolution.

UNIT –IV

Agents: Definition of agents, Agent architectures (e.g., reactive, layered, cognitive), Multi-agent systems- Collaborating agents, Competitive agents, Swarm systems and biologically inspired models. Expert Systems: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.

UNIT –V

Expert Systems: Architecture of expert systems, Roles of expert systems, Knowledge Acquisition, Meta knowledge, Heuristics. Typical expert systems- MYCIN, DART, XOON, Expert systems shells.

Text Books:

1. Artificial Intelligence, Saroj kaushik Published by Cengage Learning India, 2011
2. Artificial Intelligence and Machine Learning By Vinod Chandra S.S., Anand Hareendran S
3. Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach “Second Edition" Pearson Education

Reference Books:

1. Ivan Bratko "PROLOG Programming for Artificial Intelligence", Pearson Education, Third Edition.
2. Elaine Rich and Kevin Knight "Artificial Intelligence "Third Edition
3. Han Kamber, “Data Mining Concepts and Techniques”, Morgann Kaufmann Publishers.

G. Luger, W. A. Stubblefield, "Artificial Intelligence", Third Edition, AddisoWesley Longman, 1998.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-1 **Advanced Unix Programming**

SEMESTER - II L-T-P-C: 3-0-0-3

Course Objectives:

- To understand the fundamental design of the unix Programming
- To become fluent with the systems calls provided in the unix environment
- To be able to design and build an application/service over the unix operating system

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

Course Outcomes		Knowledge Level (K)#
CO1	Explain Networking Commands, File handling Utilities and shell script examples	**K3**
CO2	Discuss about Unix file structure ,directories and system calls	**K6**
CO3	Compare process and threads concepts	**K3**
CO4	How to Implementing client server program using pipes and FIFOs	**K1**
CO5	Demonstrate socket structure, socket system calls for connection oriented protocol and connectionless protocol.	**K5**

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	1									
CO2		1						1		
CO3						1				1
CO4			1				1			
CO5	1				2				1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I

Review of Unix Utilities and Shell Programming: -File handling utilities, security by file permissions, process utilities, disk utilities, networking commands, backup utilities, text processing utilities. **Shell Programming:** shell, shell responsibilities, pipes and input redirection, output redirection, here documents, the shell as a programming language, shell meta characters, shell variables, shell commands, the environment, control structures, shell script examples.

UNIT-II

Unix Files: Unix file structure, directories, files and devices, System calls, library functions, low level file access, usage of open, create, read, write, close, lseek, stat, fstat, octl, umask, dup, dup2, Differences between system call and library functions. File and directory maintenance: chmod, chown, unlink, link, symlink, mkdir, rmdir, chdir, getcwd. **Directory handling system calls:** opendir, readdir, closedir, rewinddir, seekdir,telldir

UNIT-III

Unix Process: Threads and Signals: process, process structure, starting new process, waiting for a process, zombie process, orphan process, process control, process identifiers, system call interface for process management, -fork, vfork, exit, wait, waitpid, exec, system.

Signals: Signal functions, unreliable signals, interrupted system calls, kill and raise functions, alarm, pause functions, abort, sleep functions.

UNIT-IV

Inter process Communication: Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes, FIFOs, message queues, semaphores and shared memory. Differences between pipes and FIFOs. Implementing client server program using pipes and FIFOs. **Message Queues-**: IPC, permission issues, Access permission modes, message structure, working with message queues, client/server example. **Semaphores:** Creating semaphore sets, Unix kernel support for semaphores, Unix APIs for semaphores, file locking using semaphores.

UNIT-V

Shared Memory: Working with a shared memory segments, Unix kernel support for shared memory, client/server example.**Sockets:** Berkeley sockets, socket structure, socket system calls for connection oriented protocol and connectionless protocol, implementing client server programs using TCP and UDP sockets.

Text books:

1. Advanced programming in the unix environment, w- Richard Stevens 2nd Edition Pearson education
2. Unix Concepts and Applications, 3/e, Sumitabha Das, TMH

Reference books:

1. Unix and shell Programming, Sumitabha Das,TMH
2. A Beginner's Guide to Unix, N.P.Gopalan, B.Sivaselva,PHI
3. Unix Shell Programming, Stephen G.Kochan, Patrick Wood, 3/e,Pearson

Unix Shell Programming, Lowell Jay Arthus & Ted Burns,3/e,GalGotia

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-1 **Data Warehousing and Data mining**
SEMESTER - II L-T-P-C: 3-0-0-3

Course Objectives:

- Be familiar with mathematical foundations of data mining tools..
- Understand and implement classical models and algorithms in data warehouses and data mining
- Characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	An ability to understand the basics of types of data, quality of data, suitable techniques required for preprocessing and measures required to perform data analysis	**K2**
CO2	Describe the need of classification, identify suitable technique(s) to perform classification, model building and evaluation	**K3**
CO3	Identify the requirements and usage of association rule mining on categorical and continuous data.	**K3**
CO4	Compare and Identify suitable clustering algorithm(s) (apply with open source tools), interpret, evaluate and report the result	**K4**
CO5	Describe the requirements and the need of web mining	**K2**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1				1					1
CO2	1		1							
CO3								3		
CO4			3							
CO5	1						2			

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-1:
Introduction to Data mining, types of Data, Data Quality, Data Processing, Measures of Similarity and Dissimilarity, Exploring Data: Data Set, Summary Statistics, Visualization, Data Warehouse, OLAP and multi dimensional data analysis.

UNIT-II:
Classification: Basic Concepts, Decision Trees and model evaluation: General approach for solving a classification problem, Decision Tree induction, Model over fitting: due to presence of noise, due to lack of representation samples, Evaluating the performance of classifier. Nearest Neighborhood classifier, Bayesian Classifier, Support vector Machines: Linear SVM, Separable and Non Separable case.

UNIT-III:
Association Analysis: Problem Definition, Frequent Item-set generation, rule generation, compact representation of frequent item sets, FP-Growth Algorithms. Handling Categorical, Continuous attributes, Concept hierarchy, Sequential, Sub graph patterns

UNIT-IV:

Clustering: Over view, K-means, Agglomerative Hierarchical clustering, DBSCAN, Cluster evaluation: overview, Unsupervised Cluster Evaluation using cohesion and separation, using proximity matrix, Scalable Clustering algorithm

UNIT-V:

Web data mining: Introduction, Web terminology and characteristics, Web content mining, Web usage mining, web structure mining, Search Engines: Characteristics, Functionality, Architecture, Ranking of WebPages, Enterprise search

Text Books:

1. Introduction to Data Mining, Tan, Steinbach and Vipin Kumar, Pearson Education, 2016
2. Data Mining:Concepts and Techniques, 2ndEdition, Jiawei Han and MichelineKamber, ELSEVIER

Reference Books:

1. Data Mining: The Textbook, Springer, May 2015, Charu C. Aggarwal.

Web resources:

1. NPTEL: https://nptel.ac.in/courses/106/105/106105174/
https://www.saedsayad.com/data_mining.htm

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Object Oriented Programming Lab

SEMESTER - II **L-T-P-C: 0-0-2-1**

Course Objectives:

- To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling and functions.
- To understand the importance of Classes & objects along with constructors, Arrays and Vectors.
- Discuss the principles of inheritance, interface and packages and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages.
- To understand importance of Multi-threading & different exception handling mechanisms.
- To learn experience of designing, implementing, testing, and debugging graphical user interfaces in Java using applet and AWT that respond to different user events.
- To understand Java Swings for designing GUI applications based on MVC architecture

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Apply OOP concepts to solve real world problems	**K2**
CO2	Implement different forms of inheritance	**K3**
CO3	Create packages and to reuse them.	**K6**
CO4	Implement multi threaded programs using synchronization concepts	**K3**
CO5	Create user defined exceptions	**K6**
CO6	Design GUI applications using AWT and SWINGS.	**K6**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	1							1	
CO2			1		1					
CO3							1			
CO4			1		1					
CO5					1					
CO6		2							1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

List of Experiments:

1. The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java Program that uses both recursive and non recursive functions to print the nth value of the Fibonacci sequence.
2. Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer.

3. Write a Java Program that checks whether a given string is a palindrome or not. Ex. MALAYALAM is a palindrome.
4. Write a Java Program for sorting a given list of names in ascending order.
5. Write a Java Program that illustrates how runtime polymorphism is achieved.
6. Write a Java Program to create and demonstrate packages.
7. Write a Java Program, using StringTokenizer class, which reads a line of integers and then displays each integer and the sum of all integers.
8. Write a Java Program that reads on file name form the user then displays information about whether the file exists, whether the file is readable/ writable, the type of file and the length of the file in bytes and display the content of the using File Input Stream class.
9. Write a Java Program that displays the number of characters, lines and words in a text/text file.
10. Write an Applet that displays the content of a file.
11. 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.
12. Write a Java Program for handling mouse events.
13. Write a Java Program demonstrating the life cycle of a thread.
14. Write a Java Program that lets users create Pie charts. Design your own user interface (with Swings & AWT).

Write a Java Program to implement a Queue, using user defined Exception Handling (also make use of throw, throws).

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
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 Hamiltaonian 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.
10. A Practical guide to Data structures and Algorithms using Java,Goldman&Goldman, Chapman and Hall/CRC,Taylor and Francis Group.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Software Engineering Lab

1. Write down the problem statement for a suggested system of relevance.
2. Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.
3. To perform the function oriented diagram: Data Flow Diagram (DFD) and Structured chart.
4. To perform the user‘s view analysis for the suggested system: Use case diagram.
5. To draw the structural view diagram for the system: Class diagram, object diagram.
6. To draw the behavioral view diagram : State-chart diagram, Activity diagram
7. To perform the behavioral view diagram for the suggested system : Sequence diagram, Collaboration diagram
8. To perform the implementation view diagram: Component diagram for the system.
9. To perform the environmental view diagram: Deployment diagram for the system.
10. To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.
11. Perform Estimation of effort using FP Estimation for chosen system.
12. To Prepare time line chart/Gantt Chart/PERT Chart for selected software project.

Text Books: 1. K.K. Aggarwal & Yogesh Singh, ―Software Engineering‖, New Age International, 2005 2. Pankaj Jalote, ―An Integrated Approach to Software Engineering‖, Second Edition, Springer

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Machine Learning

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

From the course the student will learn

- To learn patterns and concepts from data without being explicitly programmed in various IOT nodes.
- To design and analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
- Explore supervised and unsupervised learning paradigms of machine learning.
- To explore Deep learning technique and various feature extraction strategies.

Course Outcomes(CO's): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Illustrate and comprehend the basics of Machine Learning with Python	**K2**
CO2	Demonstrate the algorithms of Supervised Learning and be able to differentiate linear and logistic regressions	**K2**
CO3	Demonstrate the algorithms of Unsupervised Learning and be able to understand the clustering algorithms	**K2**
CO4	Evaluate the concepts of binning, pipeline Interfaces with examples	**K5**
CO5	Apply the sentiment analysis for various case studies	**K3**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1		2							
CO2					1	1				
CO3							2			
CO4							1			
CO5	1				1				2	

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

Introduction to Machine Learning with Python: Introduction to Machine Learning, basic terminology, Types of Machine Learning and Applications, Using Python for Machine Learning: Installing Python and packages from the Python Package Index, Introduction to NumPy, SciPy, matplotlib and scikit-learn, Tiny application of Machine Learning.

UNIT-II:

Supervised Learning: Types of Supervised Learning, Supervised Machine Learning Algorithms: k-Nearest Neighbors, Linear Models, Naive Bayes Classifiers, Decision Trees, Ensembles of Decision Trees, Kernelized Support Vector Machines, Uncertainty Estimates from Classifiers.

UNIT-III:

Unsupervised Learning: Types of Unsupervised Learning, challenges, Preprocessing and scaling, Dimensionality Reduction, Feature Extraction, Manifold Learning, Clustering: K-Means Clustering, Agglomerative Clustering, DBSCAN, Comparing and Evaluating Clustering Algorithms.

UNIT-IV:

Representing Data and Engineering Features: Categorical Variables, Binning, Discretization, Linear Models, Trees, Interactions and Polynomials, Univariate Nonlinear Transformations, Automatic Feature Selection. Parameter Selection with Preprocessing, Building Pipelines, The General Pipeline Interface

UNIT-V:

Working with Text Data (Data Visualization) : Types of Data Represented as Strings, Example Application: Sentiment Analysis of Movie Reviews, Representing Text Data as a Bag of Words, Stop Words, Rescaling the Data with tf-idf, Investigating Model Coefficients, Approaching a Machine Learning Problem, Testing Production Systems, Ranking, Recommender Systems and Other kinds of Learning.

Text Books:

1. Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Muller & Sarah Guido, Orielly Publications, 2019.
2. Python Machine Learning, Sebastian Raschka & Vahid Mirjalili, 3rd Edition, 2019.
3. Building Machine Learning Systems with Python, Luis Pedro Coelho, Willi Richert, 2nd Edition, 2015.

Reference Books:

1. Machine Learning, Tom M. Mitchell, Mc Graw-Hill Publication, 2017

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Web Technologies

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

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

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Analyze a web page and identify its elements and attributes.	**K4**
CO2	To acquire knowledge of xml fundamentals and usage of xml technology in electronic data interchange	**K2**
CO3	Build dynamic web pages using JavaScript (client side programming).	**K3**
CO4	To design and develop web based enterprise systems for the enterprises using technologies like jsp, servlet.	**K6**
CO5	Build web applications using PHP	**K3**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1			1				2			
CO2	2		1				1	1		
CO3							3			
CO4	1	2						1		
CO5			3							

(Levels of Correlation: 1-low, 2-medium 3-high)

Unit I:

Web Basics- Introduction, Concept of Internet- History of Internet, Protocols of Internet, World Wide Web, URL, Web Server, Web Browser. **HTML- Introduction**, History of HTML, Structure of HTML Document: Text Basics, Structure of HTML Document: Images and Multimedia, Links and webs, Document Layout, Creating Forms, Frames and Tables, Cascading style sheets.

Unit II:

XML Introduction- Introduction of XMLXML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

Unit III:

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

Unit IV:

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP. Client-side Scripting: Introduction to JavaScript, JavaScript language – declaring variables, scope of variables, functions. event handlers (onClick, onSubmit etc.), Document Object Model, Form validation.

Unit V:

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

Text Books:

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

Reference Books:

1. Web Programming, building internet applications, Chris Bates 2″ edition, Wiley Dreamtech.
2. Java Server Pages —Hans Bergsten, SPD O'Reilly.
3. Java Script, D.Flanagan

Beginning Web Programming-Jon Duckett WROX.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Internet of Things

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

- Introduce the fundamental concepts of IoT and physical computing
- Expose the student to a variety of embedded boards and IoT Platforms
- Create a basic understanding of the communication protocols in IoT communications.
- Familiarize the student with application program interfaces for IoT.
- Enable students to create simple IoT applications.

UNIT I
Overview of IoT:

The Internet of Things: An Overview- The Flavor of the Internet of Things, The "Internet" of "Things", The Technology of the Internet of Things, Enchanted Objects, Who is Making the Internet of Things?
Design Principles for Connected Devices: Calm and Ambient Technology, Privacy, Web Thinking for Connected Devices, Affordances.
Prototyping: Sketching, Familiarity, Costs Vs Ease of Prototyping, Prototypes and Production, Open source Vs Close source, Tapping into the community.

Learning Outcomes:
After completing this Unit, students will be able to

- Explain IoT architecture. [L2]
- Interpret the design principles that govern connected devices [L2]
- Understand the significance of Prototyping [L2]

UNIT II
Embedded Devices:

Electronics, Embedded Computing Basics, Arduino, Raspberry Pi, Mobile phones and tablets, Plug Computing: Always-on Internet of Things

Learning Outcomes:
After completing this Unit, students will be able to

- Explain the basics of microcontrollers [L2]
- Outline the architecture of Arduino [L2]
- Develop simple applications using Arduino [L3]
- Outline the architecture of Raspberry Pi [L2]
- Develop simple applications using Raspberry Pi [L3]
- Select a platform for a particular embedded computing application [L3]

UNIT III

Communication in the IoT:
Internet Communications: An Overview, IP Addresses, MAC Addresses, TCP and UDP Ports, Application Layer Protocols
Prototyping Online Components:
Getting Started with an API, Writing a New API, Real-Time Reactions, Other Protocols Protocol

Learning Outcomes:
After completing this Unit, students will be able to

- Interpret different protocols and compare them [L2]
- Select protocol to be used for a specific application [L3]
- Utilize the Internet communication protocols for IoT applications [L3]
- Select IoT APIs for an application [L3]
- Design and develop a solution for a given application using APIs [L6]
- Test for errors in the application [L4]

UNIT IV

Business Models: A short history of business models, The business model canvas, Who is the business model for, Models, Funding an Internet of Things startup, Lean Startups.
Manufacturing: What are you producing, Designing kits, Designing printed circuit boards.

Learning Outcomes:
After completing this Unit, students will be able to

- Plan the business model [L6]
- Predict the market value [L5]
- Assemble the product [L6]

UNIT V

Manufacturing continued: Manufacturing printed circuit boards, Mass-producing the case and other fixtures, Certification, Costs, Scaling up software.
Ethics: Characterizing the Internet of Things, Privacy, Control, Environment, Solutions.

Learning Outcomes:
After completing this Unit, students will be able to

- Employ the manufacturing techniques [L4]
- Adapt the Ethics [L6]

Text Book:

1. Adrian McEwen, Hakim Cassimally - Designing the Internet of Things, Wiley Publications, 2012

Reference Books:

1. Arshdeep Bahga, Vijay Madisetti - Internet of Things: A Hands-On Approach, Universities Press, 2014.
2. The Internet of Things, Enabling technologies and use cases – Pethuru Raj, Anupama C. Raman, CRC Press.

Reference sites:

1. https://www.arduino.cc/
2. https://www.raspberrypi.org/

Course outcomes:

At the end of the course, students will be able to

- Choose the sensors and actuators for an IoT application (L1)
- Select protocols for a specific IoT application (L2)
- Utilize the cloud platform and APIs for IoT applications (L3)
- Experiment with embedded boards for creating IoT prototypes (L3)
- Design a solution for a given IoT application (L6)

Establish a startup [L4]

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **Cloud Computing**

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

- To explain the evolving computer model caned cloud computing.
- To introduce the various levels of services that can be achieved by cloud.
- To describe the security aspects in cloud.
- To motivate students to do programming and experiment with the various cloud computing environments.

Course Outcomes(COs): At the end of the course, student will be able to

Course Outcomes		**Knowledge Level (K)#**
CO1	Illustrate the key dimensions of the challenge of Cloud Computing	**K2**
CO2	Classify the Levels of Virtualization and mechanism of tools.	**K2**
CO3	Analyze Cloud infrastructure including Google Cloud and Amazon Cloud.	**K4**
CO4	Explain Cloud Programming and Software Environments.	**K5**
CO5	Apply authentication, confidentiality and privacy issues in Cloud resource management.	**K3**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	1		1		1					
CO2						2				
CO3				1				1		
CO4					1		1			
CO5	2								1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Systems modeling, Clustering and virtualization: Scalable Computing over the Internet, Technologies for Network based systems, System models for Distributed and Cloud Computing, Software environments for distributed systems and clouds, Performance, Security And Energy Efficiency.

UNIT–II:

Virtual Machines and Virtualization of Clusters and Data Centers: Implementation Levels of Virtualization, Virtualization Structures/ Tools and mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data Centre Automation.

UNIT–III:

Cloud Platform Architecture: Cloud Computing and service Models, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms, Inter Cloud Resource Management, Cloud Security and Trust Management. Service Oriented Architecture, Message Oriented Middleware.

UNIT–IV:

Cloud Programming and Software Environments: Features of Cloud and Grid Platforms, Parallel & Distributed Programming Paradigms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments. **Storage Systems**: Evolution of storage technology, storage models, file systems and database, distributed file systems, general parallel file systems. Google file system., Apache Hadoop, BigTable, Megastore, Amazon Simple Storage Service(S3).

UNIT-V:

Cloud Resource Management and Scheduling : Policies and Mechanisms for Resource Management Applications of Control Theory to Task Scheduling on a Cloud, Stability of a Two Level Resource Allocation Architecture, Feedback Control Based on Dynamic Thresholds. Coordination of Specialized Autonomic Performance Managers, Resource Bundling, Scheduling Algorithms for Computing Clouds, Fair Queuing, Start Time Fair Queuing, Borrowed Virtual Time, Cloud Scheduling Subject to Deadlines, Scheduling MapReduce Applications Subject to Deadlines.

Text Books:

1. Distributed and Cloud Computing, Kai Hwang, Geoffry C. Fox, Jack J. Dongarra MK Elsevier.
2. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
3. Cloud Computing, A Hands on approach, Arshadeep Bahga, Vijay Madisetti, University Press

Reference Books:

1. Cloud Computing: A Practical Approach. Anthony T.Velte. Toby J.VeFte, Robert Elsenpeter. Tata McGraw Hill. rp2Oll.
2. Enterprise Cloud Computing Gautam Shroif, Cambridge University Press. 2010.
3. Cloud Computing: Implementation, Management and Security, John W. Rittinouse, James F Ransome. CRC Press, rp2012.
4. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. George Reese, O'Really SPD, rp2Oll.
5. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Ktriaraswamy, Shahed Latif, O'Redç SPD, rp2Oll.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **Image Processing**

SEMESTER - III L-T-P-C: 3-0-0-3

Course Objectives:

- To comprehend the relation between human visual system and machine perception and processing of digital images.
- To provide a detailed approach towards image processing applications like enhancement, segmentation, and compression.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Analyze the computational methods on digital images.	**K4**
CO2	Explain the implement the spatial and frequency domain image transforms on enhancement and restoration of images.	**K2**
CO3	Elaborate understanding on machine learning techniques.	**K6**
CO4	Expected to Define and Apply the need for compression and evaluate the basic compression algorithms	**K5,1**
CO5	Make use of introduce object tracking approaches.	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		1	2						1	
CO2					2	1				
CO3	1					1	2			
CO4										
CO5					2					

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT – I:

Image Formation and Coordinate Transformations Camera Matrix, Motion/Stereo Pin-hole model, Human eye / cognitive aspects of colour / 3D space; illumination; Sampling and Quantization Coordinate transformations and camera parameters

UNIT – II:

Image Processing - Noise Removal, Blurring, Edge Detection: Canny / Gaussian/ Gabor/Texture Edges/ Curvature / Corner Detection.

UNIT – III:

Segmentation - Concept of Figure vs. Ground, Watershed, Change Detection, Background Subtraction, Texture Segmentation Gaussian Mixture Models - Applications in Color/Motion based Image Segmentation, Background Modeling and Shape Clustering

UNIT – IV:

Machine Learning techniques in Vision Bayesian Classification, Maximum Likelihood Methods, Neural Networks; Non-parametric models; Manifold estimation Support Vector Machines ; Temporal sequence learning.

UNIT – V:

Introduction to Object Tracking - Exhaustive vs. Stochastic Search Shapes, Contours, and Appearance Models. Mean-shift tracking; Contour-based models, Object Modeling and Recognition Fundamental matrix / Epipolar geometryAdaboost approaches: Face Detection / Recognition Large Datasets; Attention models.

Text Books

1. [FP]: David Forsyth and Jean Ponce, Computer Vision: A modern Approach,
   Prentice
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2008.

Reference Books:

1. E.R. Davies, Machine Vision, Theory Algorithms Practicalities, Elsevier 2005
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision. Brooks/Cole / Thomson 1999
3. Basics of some image processing aspects. Texture Chapter 24 (Perception) of Russell and Norvig: AI: A modern Approach, Prentice Hall 2000.
4. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Cambridge Univ Press 2000 More detailed treatment of 3D structure recovery
5. Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification, 2nd ed., Wiley Asia, 2002

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 Principles of Cryptography and Network Security

SEMESTER - III L-T-P-C: 3-0-0-3

Course Objectives:

- To learn various cryptographic algorithms including secret key cryptography, hashes and message digests, public key algorithms,
- To Familiar in design issues and working principles of various authentication protocols and various secure communication standards including Kerberos, IPsec, and S/MIME

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

	Course Outcomes	Knowledge Level (K)#
CO1	Explain Basic Principles, different security threats, countermeasures, foundation course of cryptography mathematics and Symmetric Encryption.	**K2**
CO2	Classify the basic principles of Asymmetric key algorithms and operations of asymmetric key cryptography.	**K4**
CO3	Design Cryptographic Hash Functions as SHA-3 and Digital Signatures as Elgamal	**K6**
CO4	Explain the concept of Revise Key Management and Distribution and User Authentication	**K3**
CO5	Determine the knowledge of Network and Internet Security Protocols such as S/MIME	**K5**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2									
CO2		1						1		
CO3		2								1
CO4	1		1							
CO5	1								1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT I:

Basic Principles: Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography. **Symmetric Encryption:** Mathematics of Symmetric Key Cryptography, Introduction to Modern Symmetric Key Ciphers, Data Encryption Standard, Advanced Encryption Standard.

UNIT II:

Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography-Primes, primality Testing, Factorization, Asymmetric Key Cryptography-RSA Cryptosystem, Rabin Cryptosystem, ElGamal Cryptosystem, Elliptic Curve Cryptosystem

UNIT III:

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions Requirements and Security Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA), SHA-3. **Digital Signatures:** Elgamal Digital Signature Scheme, Schnorr Digital Signature, NIST Digital Signature Algorithm

Unit IV:

Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates. **User Authentication:** User Authentication, Remote User-Authentication Principle, Remote User-Authentication Using Symmetric Encryption, Kerberos, Remote User-Authentication Using Asymmetric Encryption

Unit V: Network and Internet Security

Electronic Mail Security: Internet Mail Architecture, Email Formats, Email Threats and Comprehensive Email Security, S/MIME. **IP Security:** IP Security Policy, Encapsulating Security Payload, Combining Security Associations Internet Key Exchange

Text Books:

1. Cryptography and Network Security, 3rd Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015
2. Cryptography and Network Security, William Stallings, Global Edition, 7e Pearson, 2017

Reference Books:

1. Network Security and Cryptography, First Edition, Bernard Meneges, Cengage Learning, 2018

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **SOFTWARE TESTING**

SEMESTER - III **L-T-P-C: 3-0-0-3**

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

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
MOBILE APPLICATION DEVELOPMENT

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

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

Unit I

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

Unit II

J2ME Architecture and Development Environment, J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2ME Style, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit, J2ME Best Practices and Patterns: The Reality of Working in a J2ME World, Best Practices

Unit III

Commands, Items, and Event Processing, J2ME User Interfaces, Display Class, the Palm OS Emulator, Command Class, Item Class, Exception Handling

High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, Text Box Class, Ticker Class

Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation

Unit IV

Record Management System: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener

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

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

Unit V

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

TEXT BOOKS:

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

REFERENCE BOOKS:

1.Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, Pearson Education, 2004

2. Beginning Java ME Platform, Ray Rischpater, Apress, 2009

3.Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, Apress, 2005

4.Kicking Butt with MIDP and MSA:Creating Great Mobile Applications,1st edition,J.Knudsen,Pearson.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
WEB TECHNOLOGIES LAB

SEMESTER - III **L-T-P-C: 0-0-3-1.5**

Course Objectives:

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

List of Sample Problems

1. Develop static pages (using Only HTML) of an online Book store. The pages should resemble: www.amazon.com The website should consist the following pages.
   Home page, Registration and user Login
   User Profile Page, Books catalog
   Shopping Cart, Payment By credit card
   Order Conformation

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

3. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

*4. Bean Assignments

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

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

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

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

Light Transition	Automobile State
Red ---> Yellow	Ready
Yellow ---> Green	Move
Green --> Red	Stopped

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

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

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

8.Redo the problem 5 using PHP.

Additional Assignment Problems for the WT Lab.:

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

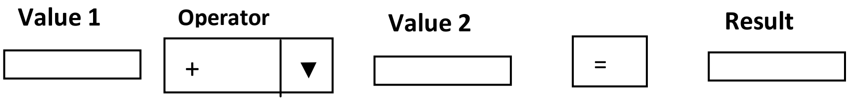
Write a java swing application that takes a text file name as input and counts the characters, words and lines in the file. Words are separated with white space characters and lines are separated with new line character.

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

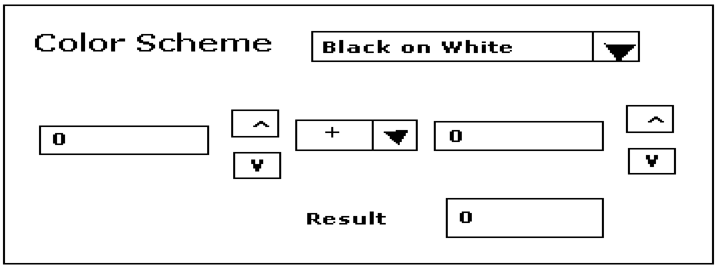
Write an HTML page that contains a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).

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

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

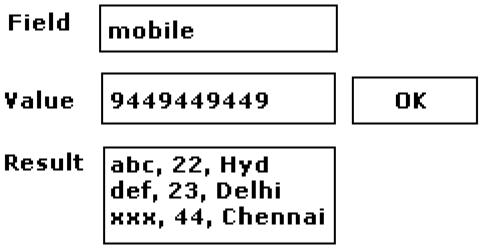


Write a Java program that creates a calculator GUI, as shown in figure. Extra components may be added for convenience:



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

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



Field mobile

Value 9449449449

Result
abc, 22, Hyd
def, 23, Delhi
xxx, 44, Chennai

Consider the following web application for implementation:
The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
If name and password matches, serves a welcome page with user's full name.
If name matches and password doesn't match, then serves “password mismatch” page
If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application using:

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

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

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **Machine Learning with Python Lab**

SEMESTER - III L-T-P-C: 0-1-2-2

Course Objectives:

- Make use of Data sets in implementing the machine learning algorithms
- Implement the machine learning concepts and algorithms in any suitable language of choice.
- Design Python programs for various Learning algorithms.

Course Outcomes(COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Implement procedures for the machine learning algorithms	**K4**
CO2	Design Python programs for various Learning algorithms	**K6**
CO3	Apply appropriate data sets to the Machine Learning algorithms	**K3**
CO4	Identify and apply Machine Learning algorithms to solve real world problems	**K2**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	1								
CO2		1	3							
CO3								1	2	
CO4	3							1	1	

(Levels of Correlation: 1-low, 2-medium 3-high)

Experiment 1:

Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .csv file

Experiment 2:

For a given set of training data examples stored in a .csv file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples

Experiment 3:

Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Experiment 4:

Write a Python program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions.

Experiment 5:

Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

Experiment 6:

Write a program to implement the naive Bayesian classifier for a sample training data set stored as a .csv file. Compute the accuracy of the classifier, considering few test data sets.

Experiment 7:

Write a Python program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set.

Experiment 8:

Assuming a set of documents that need to be classified, use the naive Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision and recall for your data set.

Experiment 9:

Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering using Python Programming.

Experiment 10:

Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

INTERNET OF THINGS LABORATORY

Semester – III **L-T-P-C: 0-0-3-1.5**

Practicals:

1. Select any one development board (Eg., Arduino or Raspberry Pi) and control LED using the board.
2. Using the same board as in (1), read data from a sensor. Experiment with both analog and digital sensors.
3. Control any two actuators connected to the development board using Bluetooth.
4. Read data from sensor and send it to a requesting client. (using socket communication)
   Note: The client and server should be connected to same local area network.
5. Create any cloud platform account, explore IoT services and register a thing on the platform.
6. Push sensor data to cloud.
7. Control an actuator through cloud.
8. Access the data pushed from sensor to cloud and apply any data analytics or visualization services.
9. Create a mobile app to control an actuator.
10. Design an IoT based air pollution control system which monitors the air pollution by measuring carbon monoxide, ammonia, etc and gives alarm or sends message when the pollution level is more than permitted range.
11. Design an IoT based system which measures the physical and chemical properties of the water and displays the measured values.
12. Identify a problem in your local area or college which can be solved by integrating the things you learned and create a prototype to solve it (Mini Project).
13. Design a business model canvas for a digital display

Text Book:

2. Adrian McEwen, Hakim Cassimally - Designing the Internet of Things, Wiley Publications, 2012.
3. Alexander Osterwalder, and Yves Pigneur – Business Model Generation – Wiley, 2011

Reference Books:

3. Arshdeep Bahga, Vijay Madisetti - Internet of Things: A Hands-On Approach, Universities Press, 2014.
4. The Internet of Things, Enabling technologies and use cases – Pethuru Raj, Anupama C. Raman, CRC Press.

Reference sites:

https://www.arduino.cc/

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
MOBILE APPLICATION DEVELOPMENT LABORATORY

Semester – III **L-T-P-C: 0-0-2-1**

List of Experiments:

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

1) If the Java Development Kit (JDK) is not there or only having the Java Runtime Environment (JRE) installed, install the latest JDK from http://java.sun.com/javase/downloads/index.jsp. Current stable release of Java is JDK 6 Update 7 but check the web page in case there are newer non-beta releases available.
2) Next, download the Java Wireless Toolkit (formerly called J2ME Wireless Toolkit) from: http://java.sun.com/products/sjwtoolkit/download.html.
3) Run the installer (for example, for Windows it is: sun_java_wireless_toolkit- 2_5_2-windows.exe). The installer checks whether a compatible Java environment has been pre-installed. If not, it is necessary to uninstall old versions of Java and perform Step 1 again.

Once after successful installation of Java and the tool kit compile this program and run the following program in the toolkit.

Steps to run this program in toolkit:

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

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
public class HelloWorld extends MIDlet{
  private Form form;
  private Display display;
public HelloWorld(){
super();
  }
public void startApp(){
form = new Form("Hello World");
    String msg = "Hello World!!!!!!!";
form.append(msg);
    display = Display.getDisplay(this);
display.setCurrent(form);
  }
public void pauseApp(){}
public void destroyApp(boolean unconditional){
notifyDestroyed();
} }
```

Week - 2 Working with J2ME Features:

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

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

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

2.2 Event Handling.

Create a menu which has the following options:

* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

2.3. Input checking

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

* Area code should be one of the following: 040, 041, 050, 0400, 044
* There should 6-8 numbers in telephone number (+ area code)

Week - 3 Threads & High Level UI:

3.1. Create a slide show which has three slides, which includes only text. Program should change to the new slide after 5 seconds. After the third slide program returns to the first slide.

3.2 High-level UI

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

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

Week - 4 Working on Drawing and Images

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

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

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

Week - 5 Developing Networked Applications using the Wireless Toolkit

Creating a Simple Client-Server Application

Create, compile and run a basic UDP-based client-server application.

Creating the Datagram Server project

1) Click on Wireless Toolkit 2.5.2 under the group: All Programs→Sun Java
(TM) Wireless Toolkit 2.5.2.

2) Click on 'New Project...' button.

3) Enter project name as 'DatagramServer'. Enter MIDlet name as 'DatagramServer'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class, otherwise the application won‘t run.

4) Another window pops up where it is required to select a target platform. Select 'MIDP 1.0' from the drop down list.

5) After clicking OK, the project is created; and the Wireless Toolkit tells that the name of the folder where source code files are created. The path of the source code folder is displayed in the debug output window.

Creating and Compiling the Datagram Server source files

The Wireless Toolkit does not come with an IDE by default so Use any IDE or a text editor like Notepad.

1) Create a new text file called DatagramServer.java in the source folder of the project. The exact path of this folder is displayed in the Wireless Toolkit window.

2) Paste contents DatagramServer.java from into the source file.

Running your Server application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of your application highlighted on its screen as shown below.

3) To start the application, click on the right soft-key (marked with a dot) below the ‗Launch‘ command.

4) The phone simulator might ask if it is OK to run the network application. Select ‗Yes‘ by clicking on the appropriate soft-key. The server is now up and running.

5) Keep the server running during the creation, compilation and running of the Datagram Client application.

Creating the Datagram Client project

1) Use the same instance of the Wireless Toolkit that is used for creating and compiling the Datagram Server project.

2) Click on 'New Project...' button.

3) A new window pops up. Enter project name as 'DatagramClient'. Enter MIDlet name as 'DatagramClient'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class.

4) Another window pops up where one has to select a target platform. Select 'MIDP 1.0' from the drop down list.

5) After clicking OK, the project is created and the Wireless Toolkit tells where to place the source code files. The path of the source code folder is displayed in the debug output window as explained before.

Creating and Compiling the Datagram Client source files

1) Create a new text file called DatagramClient.java in the source folder of the project.

2) Paste contents DatagramClient.java into the source file.

3) Then click on the Build button in the Wireless Toolkit window. If the compilation is OK, it will say Build Complete in the window's debug output window, otherwise it will show the errors. Note: In the source code, use the System.out.println() statement to output debug information to this window.

Running your Client application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of the application highlighted on its screen.

3) To start the application, click on the right soft-key (marked with a dot) below the _Launch‘ command.

4) The phone simulator might ask if it is OK to run the network application. Select _Yes‘ by clicking on the appropriate soft-key. The client is now up and running.

5) When the client executes on the phone simulator, one should see a text box with the caption 'Message'. Enter any message and press the right soft-key (corresponding to Send). If the client- server application is working properly, the screen of the server phone will display the message sent by the client and the client screen will now display a message sent by the server in response. The response message from the server is the original client message in reverse. 6) Try various features of the phone simulator including the different look-and feel options.

Week - 6 Authentication with a Web Server

6.1 Write a sample program to show how to make a SOCKET Connection from j2me phone.

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

6.2 Login to HTTP Server from a J2ME Program

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

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

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

Text Books:

1. J2ME: The Complete Reference, James Keogh, TMH.

References:

4. Enterprise J2ME: Developing Mobile Java Applications, Michael Juntao Yuan, Pearson Education, 2004.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Program Elective:3
Block Chain technologies

SEMESTER - IV **L-T-P-C: 3-0-0-3**

Course Objectives:

- Impart strong technical understanding of Blockchain technologies
- Develop familiarity of current technologies, tools, and implementation strategies
- Introduce application areas, current practices, and research activity

Course Outcomes (Cos): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Demonstrate the foundation of the Blockchain technology and understand the processes in payment and funding.	K2
CO2	Identify the risks involved in building Blockchain applications.	K5
CO3	Review of legal implications using smart contracts.	K4
CO4	Choose the present landscape of Blockchain implementations and Understand Cryptocurrency markets.	K3
CO5	Examine how to profit from trading cryptocurrencies.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3						2			
CO2								3		
CO3				2					1	
CO4										
CO5										2

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT – I:

The consensus problem, Asynchronous Byzantine Agreement, AAP protocol and its analysis, Nakamoto Consensus on permission-less, nameless, peer-to-peer network, Abstract Models for BLOCKCHAIN, GARAY model, RLA Model, Proof of Work (PoW) as random oracle, formal treatment of consistency, liveness and fairness - Proof of Stake (PoS) based Chains, Hybrid models (PoW + PoS).

UNIT – II:

cryptographic basics for cryptocurrency, A short overview of Hashing, signature schemes, encryption schemes and elliptic curve cryptography

UNIT – III:

Bitcoin, Wallet, Blocks, Merkley Tree, hardness of mining, transaction verifiability, anonymity, forks, double spending, mathematical analysis of properties of Bitcoin.

UNIT – IV:

Ethereum: Ethereum Virtual Machine (EVM),Wallets for Ethereum, Solidity, Smart Contracts, some attacks on smart contracts

UNIT – V:

(Trends and Topics): Zero Knowledge proofs and protocols in Blockchain, Succinct non interactive argument for Knowledge (SNARK), pairing on Elliptic curves ,Zcash.

Text Books:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016. (Free download available)

Reference Books:

1. Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015 (article available for free download) {curtain raiser kind of generic article, written by seasoned experts and pioneers}.
2. J.A.Garay et al, The bitcoin backbone protocol - analysis and applications EUROCRYPT 2015 LNCS VOl 9057, (VOLII), pp 281-310. (Also available at eprint.iacr.org/2016/1048). (serious beginning of discussions related to formal models for bitcoin protocols).

R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks , EUROCRYPT 2017, (eprint.iacr.org/2016/454) . A significant progress and consolidation of several principles).

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective:3
Deep Learning

SEMESTER - IV **L-T-P-C: 3-0-0-3**

UNIT 1:

Introduction to TensorFlow :Computational Graph, Key highlights, Creating a Graph, Regression example, Gradient Descent, TensorBoard, Modularity, Sharing Variables,Keras Perceptrons: What is a Perceptron, XOR Gate

UNIT 2:

Activation Functions : Sigmoid,ReLU, Hyperbolic Fns,Softmax Artificial Neural Networks : Introduction, Perceptron Training Rule, Gradient Descent Rule

UNIT 3:

Gradient Descent and Backpropagation: Gradient Descent, Stochastic Gradient Descent, Backpropagation, Some problems in ANN Optimization and Regularization :Overfitting and Capacity, Cross Validation, Feature Selection, Regularization, Hyperparameters

UNIT 4:

Introduction to Convolutional Neural Networks: Introduction to CNNs, Kernel filter, Principles behind CNNs, Multiple Filters, CNN applications Introduction to Recurrent Neural Networks: Introduction to RNNs, Unfolded RNNs, Seq2Seq RNNs, LSTM, RNN applications

UNIT 5:

Deep Learning applications: Image Processing, Natural Language Processing, Speech Recognition, Video Analytics

Text Book 1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.

References 1. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006.

2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.

3. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.

4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective:3

Design Patterns

SEMESTER - IV L-T-P-C: 3-0-0-3

Course Objectives:

- Understand the concept of Design patterns and its importance.
- Understand the behavioral knowledge of the problem and solutions.
- Relate the Creational, Structural, behavioral Design patterns.
- Apply the suitable design patterns to refine the basic design for given context.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Illustrate the appropriate design patterns to solve object-oriented design problems.	K2
CO2	Apply structural patterns to solve design problems.	K3
CO3	Evaluate the design solutions by using behavioral patterns.	K5
CO4	Develop design solutions using creational patterns	K6
CO5	Demonstrate about Advanced Patterns like Pattern Catalogs	K3

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		3					2			
CO2		3					1			
CO3					2		1			
CO4								2		
CO5			1					2		

(Levels of Correlation: 1-low, 2-medium 3-high)

Unit-I:

Introduction: History and Origin of Patterns, Design Patterns in MVC, Describing Design Patterns, How Design Patterns Solve Design Problems, selecting a Design Pattern, Using a Design Pattern

Unit-II:

Design Patterns-1: Creational, Abstract Factory-Builder, Factory Method, Prototype-Singleton

Unit- III:

Design Patterns-2: Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy

Unit-IV:

Design Patterns-3: Behavioural Patterns, Chain of Responsibility, Command-Interpreter, Iterator- Mediator, Memento, Observer, State, Strategy, Template Method, Visitor

Unit-V:

Advanced Patterns: Pattern Catalogs and Writing Patterns, Patterns and Case Study: Designing a Document Editor Anti-Patterns - Case Studies in UML and CORBA, Pattern Community.

Text Books:

1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design patterns: Elements of Reusable object-oriented software, Addison-Wesley, 1995.
2. James W Cooper, Java Design Patterns - A Tutorial, Addison-Wesley, 52000.

Reference Books:

1. Craig Larman, Applying UML and Patterns: An Introduction to object-Oriented Analysis and Design and iterative development,3rd Edition, Pearson,2005.
2. Thomas J Mowbray and Raphael Malveau, CORBA and Design Patterns, John Wiley, 1997.
3. William J Brown, Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis, John Wiley, 1998.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Program Elective-4

Big Data Analytics

SEMESTER - IV **L-T-P-C: 3-0-0-3**

Course Objectives:

- To know the fundamental concepts of big data and analytics.
- To explore tools and practices for working with bigdata
- To learn about stream computing.
- To know about the research that requires the integration of large amounts of data.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Identify the need-based tools, viz., Pig and Hiveand to handle and formulate an effective strategy to implement a successful Data analytics project	K3
CO2	Organize the existing technologies and the need of distributed files systems to analyze the big data	K3
CO3	To Discuss the cluster and classification techniques	K5
CO4	Analyze the concepts of strem memory and spark models.	K4
CO5	Explain the use of NoSQL database in data analytics.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1			3					1		
CO2	1				2					
CO3					2					
CO4								3		
CO5			3							

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

Introduction to Big Data- Evolution of Big data, Best Practices for Big data Analytics, Big data characteristics, Validating, The Promotion of the Value of Big Data,Big Data Use Cases, Characteristics of Big Data Applications, Perception and Quantification of Value, Understanding Big Data Storage, A General Overview of High, Performance Architecture, HDFS, MapReduce and YARN, Map Reduce Programming Model

UNIT-II:

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 Zoo Keeper, IBM InfoSphere Big Insights and Streams

UNIT-III:

Clustering and Classification-Advanced Analytical Theory and Methods: Overview of Clustering, K-means, Use Cases - Overview of the Method, Determining the Number of Clusters, Diagnostics, Reasons to Choose and Cautions. Classification: Decision Trees, Overview of a Decision Tree, The General Algorithm, Decision Tree Algorithms, Evaluating a Decision Tree, Decision Trees in R, Naïve Bayes, Baye'sTheorem, Naïve Bayes Classifier.

UNIT- IV:

Stream Memory and Spark- Introduction to Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Introduction to Spark Concept, Spark Architecture and components, spark installation, spark RDD(Resilient Distributed Dataset), spark RDD operations.

UNIT-V:

NOSQL Data Management for Big Data and Visualization- NoSQL Databases: Schema-less Models‖: Increasing Flexibility for Data Manipulation, Key Value Stores, Document Store, Tabular Stores, Object Data Stores, Graph Databases Hive, Sharding, Hbase, Analyzing big data with twitter, Big data for E-Commerce Big data for blogs, Review of Basic Data Analytic Methods using R.

Text Books:

1. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.
2. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", Morgan Kaufmann/El sevier Publishers, 2013.

Reference Books:

1. MichaelBerthold,DavidJ.Hand,"IntelligentDataAnalysis",Springer,2007.
2. TomWhite"Hadoop:TheDefinitiveGuide"ThirdEdition,O'reillyMedia,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. BillFranks,"TamingtheBigDataTidalWave:FindingOpportunitiesinHugeD ata Streams with Advanced Analytics", John Wiley& sons,2012.
5. GlennJ.Myatt,"MakingSenseofData",JohnWiley&Sons,2007.
6. Pete Warden, "Big Data Glossary", O'Reilly,2011.
7. Jiawei Han, Micheline Kamber "Data Mining Concepts and Techniques", 2nd Edition, Elsevier, Reprinted2008.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Program Elective-4

Software Defined Networks

SEMESTER - IV **L-T-P-C: 3-0-0-3**

Course Objectives:

- To learn threats and risks within context of the cyber security architecture.
- Student should learn and Identify security tools and hardening techniques.
- To learn types of incidents including categories, responses and timelines for response.

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

	Course Outcomes	**Knowledge Level (K)#**
CO1	Explain the key benefits of SDN by the separation of data and control planes	K5
CO2	Interpret the SDN data plane devices and Openflow Protocols	K3
CO3	Apply the operation of SDN control plane with different controllers	K3
CO4	Apply techniques that enable applications to control the underlying network using SDN	K4
CO5	Design Network Functions Virtualization components and their roles in SDN	K6

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1		1						3		
CO2	1		3							
CO3			2					1		
CO4										
CO5			2							

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

SDN Background and Motivation-Evolving network requirements-The SDN Approach: Requirements, SDN Architecture, Characteristics of Software-Defined Networking, SDN and NFV-Related Standards: Standards-Developing Organizations, Industry Consortia, Open Development Initiatives.

UNIT-II:

SDN Data plane and OpenFlow-SDN data plane: Data plane Functions, Data plane protocols, Openflow logical network Device: Flow table Structure, Flow Table Pipeline, The Use of Multiple Tables, Group Table- OpenFlow Protocol.

UNIT-III:

SDN Control Plane-SDN Control Plane Architecture: Control Plane Functions, Southbound Interface, Northbound Interface, Routing, ITU-T Model- Open Daylight-REST- Cooperation and Coordination among Controllers.

UNIT-IV:

SDN Application Plane-SDN Application Plane Architecture: Northbound Interface, Network Applications, User Interface- Network Services Abstraction Layer :Abstractions in SDN, Frenetic-Traffic Engineering Measurement and Monitoring- Security- Data Center Networking- Mobility and Wireless.

UNIT-V:

Network Functions Virtualization- Background and Motivation for NFV-Virtual Machines- NFV Concepts: Simple Example of the Use of NFV,NFV Principles, High-Level NFV Framework, NFV Benefits and Requirements-NFV Reference Architecture: NFV Management and Orchestration

Text Books:

1. William Stallings, “Foundations of Modern Networking”,Pearson Ltd.,2016.
2. Software Defined Networks: A Comprehensive Approach by Paul Goransson and Chuck Black,Morgan Kaufmann Publications, 2014
3. SDN - Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013

Reference Books:

1. Feamster, Nick, Jennifer Rexford, and Ellen Zegura. "The road to SDN: an intellectual historyof programmable networks." ACM SIGCOMM Computer Communication Review 44.2 (2014): 87-98.
2. Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedingsof the IEEE 103.1 (2015): 14-76.

Web Reference:

https://www.coursera.org/learn/sdn