

ACADEMIC REGULATIONS PROGRAM STRUCTURE and DETAILED SYLLABUS

Master of Computer Applications

(Effective for the students admitted from the Academic Year 2014-15)



**GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY**
(Autonomous)



**Gokaraju Rangaraju
Institute of Engineering and Technology, Hyderabad
Department of Master of Computer Applications (MCA)
GR14 Regulations**

Gokaraju Rangaraju Institute of Engineering & Technology 2014 Regulations (GR 14 Regulations) are given hereunder. These regulations govern the programmes offered by the Department of MCA with effect from the students admitted to the programmes in 2014-15 academic year.

1. **Programme Offered:** The programme offered by the Department is Master of Computer Applications (MCA), a three-year regular programme.
2. **Medium of Instruction:** The medium of instruction (including examinations and reports) is English.
3. **Admissions:** Admission to the MCA Programme shall be made subject to the eligibility, qualifications and specialization prescribed by the State Government/University from time to time. Admissions shall be made either on the basis of the merit rank obtained by the student in Common Entrance Examination conducted by the Government/University or on the basis of any other order of merit approved by the Government/University, subject to reservations prescribed by the Government/University from time to time.
4. **Programme Pattern:**
 - a) Each Academic year of study is divided into two semesters.
 - b) Minimum number of instruction days in each semester is 90.
 - c) Total Credits: 150.
 - d) All the registered credits will be considered for the calculation of the final percentage of marks.
5. **Award of MCA Degree:** A student will be declared eligible for the award of the MCA Degree if he/she fulfills the following academic requirements:
 - a) A student shall be declared eligible for the award of MCA degree, if he/she pursues the course of study and completes it successfully in not less than three academic years and not more than six academic years.
 - b) A student has to register for all 150 credits and secure all credits.
 - c) A student, who fails to fulfill all the academic requirements for the award of the degree within six academic years from the date of admission, shall forfeit his/her seat in MCA course.



- d) The degree of Master of Computer Applications (MCA) shall be conferred by Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, on the students who are admitted to the programme and fulfill all the requirements for the award of the degree.

6. Attendance Requirements

- A student shall be eligible to appear for the semester-end examinations if he/she puts in a minimum of 75% of attendance in aggregate in all the courses concerned in the semester.
- Condonation of shortage of attendance in aggregate up to 10% (65% and above but less than 75%) in a semester may be granted. A committee headed by Dean, Academic Affairs, shall be the deciding authority for granting the condonation.
- Students who have been granted condonation shall pay a fee as decided by the Academic Council.
- Shortage of attendance more than 10 % (Attendance less than 65% in aggregate) shall in no case be condoned.
- Students, whose shortage of attendance is not condoned in any semester, are detained and are not eligible to take their end examinations of that semester. They may seek re-registration for that semester when offered next with the academic regulations of the batch into which he/she gets re-registered.

7. Paper Setting, Evaluation of Answer Scripts, Marks and Assessment

- Paper setting and evaluation of the answer scripts shall be done as per the procedures laid down by the Academic Council of the College/Institute from time to time.
- Distribution and Weight age of Marks

Particulars	Internal Evaluation	External Evaluation	Total
Theory	40	60	100
Practical	40	60	100
Mini Project	40	60	100
General Seminar	50	–	50
Project Work and Dissertation	–	Grade	Grade

- Continuous Internal Evaluation and Semester End Examination
The assessment of the student's performance in each course will be based on continuous internal evaluation and semester-end examinations. The marks for each of the components of assessment are fixed as shown in the following Table.



Assessment Procedure

S.No	Component of Assessment	Marks Allotted	Type of Assessment	Scheme of Examinations
1	Theory	40	Internal Exams & Continuous Evaluation	Two mid semester examinations shall be conducted for 30 marks each for duration of 2 hours. Average of the two mid semester examinations shall be considered i. Mid semester exams : 30 Marks ii. Tutorial : 5 Marks iii. Attendance : 5 Marks Total : 40 Marks
		60	Semester-end examination	Semester End Examinations is for duration of 3 hours. The pattern of examination paper shall be as per the guidelines of the Academic Council.
2	Practical	40	Internal Exams & Continuous Evaluation	1) Lab Internal :15 marks 2) Record : 5 marks 3) Continuous Assessment : 15 marks 4) Attendance : 5 marks
		60	Semester-end examination	The Semester-end examination shall be conducted by the faculty member concerned and an External Examiner nominated by the Head of Department. Evaluation is based on Writing and Execution of the program combined with the performance of viva voce.



- d) **Seminar:** In the III Year II Semester the student has to give two types of seminars.
- i. General Seminar
 - ii. Project Review Seminars
 - i. **General Seminar Evaluation:** The student has to choose a topic related to latest technology, other than project work and present it before Project Review Committee (PRC) constituting Head of the Department and two other senior faculty members of the Department. The Performance of the student is evaluated for 20 marks based on the presentation skills and the content of the topic and its relevance to the Present Industry.
 - ii. **Project Review Seminars Evaluation:** There will be atleast four reviews on the progress of the Project Work conducted by PRC. The performance of the student is evaluated for 30 marks. The student has to get 50% of the marks to be declared successful.
- e) **Mini Project:** Students have to take up a mini project relevant to the latest technology in II Year II semester. The Mini Project Review Committee (MPRC) constituting Head of the Department, Internal supervisor and Mini Project Coordinator as members will assess the performance of the student.
- i) **Internal Evaluation:** Assessment is based on timely progress of the work and display of Mini Projects in a road show at department level at the end of the semester.
 - ii) **External Evaluation:** A report of the Mini Project has to be submitted to the MPRC. Evaluation is based on the presentation of the Mini Project report before the MPRC and an External Examiner nominated by the Head of the Department.
- f) **Evaluation of Project Work and Dissertation:**
- (i) **Registration of project Work:** A student has to choose the topic for his MCA project work in the last 8 weeks of the III Year I Semester, in consultation with Internal /External Supervisors. In the first week of III Year II Semester, the student has to submit the proposal for project work to be carried out to the Project Review Committee (PRC) for approval. In general, the registration for the project work will be allowed twice in a year – (i) in the middle of the year, or (ii) at the end of the year.
 - (ii) The PRC shall monitor the progress of the project work and shall conduct periodic seminars in the III Year II Semester. The student must submit a status report of the project work and give the seminar/presentation after 18 weeks from the date of registration.



- (lii) The duration of the project is one Semester. The student can submit the project report after 20 weeks from the date of registration, after obtaining the approval from the PRC. Extension of time, within the total permissible limit for the completion of the program, may be considered by the PRC, on sufficient valid/genuine grounds.
- (iv) The Project work can be submitted only after successful completion of all the prescribed Courses and Seminar. The student shall submit two hard copies and one soft copy of the project work, certified in the prescribed format by the Supervisor(s).
- (v) The project work will be adjudicated by an External Examiner appointed by the Academic council.
- (vi) If the report of the examiner is favorable, Viva-voce examination shall be conducted by a Board consisting of the Supervisor, Head and the External Examiner who adjudicated the project work. The Board shall jointly evaluate the student's performance in the project work as (A) Excellent, (B) Good, (C) Satisfactory, or (D) Unsatisfactory.
In case the student gets the unsatisfactory grade, he has to reappear for the viva-voce examination, as per the recommendations of the Board. If he fails at the second Viva-voce examination also, he will not be eligible for the award of the degree, unless he is asked to revise and resubmit the Project by the Board.
- (vii) If the report of the Examiner is not favorable, the student shall revise and resubmit the project work after one Semester. If the report of the Examiner is unfavorable again, then the project work shall be summarily rejected.
- 8. Recounting of Marks in the End-Examination Answer Books:** A student can request for re-counting of his/her answer book on payment of a prescribed fee.
- 9. Re-evaluation of the End-Examination Answer Books:** A student can request for re-evaluation of his/her answer book on payment of a prescribed fee.
- 10. Supplementary Examinations:** A student who has failed in an end semester examination can appear for a supplementary examination, as per the schedule announced by the College/Institute.



11. Malpractices in Examinations: Disciplinary action shall be taken in case of malpractices during Mid / End-examinations as per the rules framed by the Academic Council.

12. Academic Requirements

- a) A student shall be deemed to have secured the minimum academic requirement in a subject if he / she secures minimum of 40% of marks in the Semester-end Examination and a minimum aggregate of 50% of the total marks in the Semester-end examination and Internal Evaluation taken together.
- b) In order to qualify for the award of MCA Degree, the student shall complete the academic requirements of passing in all the Courses as per the course structure including Seminars and Projects if any.
- c) In case a Student does not secure the minimum academic requirements in any course, he/she has to reappear for the Semester-end Examination in the course, or re-register for the same course when next offered or re-register for any other specified course, as may be required. However, one more additional chance may be provided for each student, for improving the internal marks provided the internal marks secured by a student are less than 50% and he/she failed finally in the course concerned. In the event of taking another chance for re-registration, both the internal and external marks, obtained in the previous attempt are nullified. In case of re-registration, the student has to pay the re-registration fee for each course, as specified by the College/Institute.

13. Award of Class: After a student satisfies all the requirements prescribed for the completion of the Degree and becomes eligible for the award of MCA Degree by Jawaharlal Nehru Technological University Hyderabad, he/she shall be placed in one of the following three classes (the marks awarded are from the aggregate marks secured for the 150 credits):

Class Awarded	% of Marks Secured
First Class with Distinction	Marks \geq 70%
First Class	\geq 60% Marks < 70%
Second Class	\geq 50% Marks < 60%

14. Withholding of Results: If the student has not paid dues to the Institute/ University, or if any case of indiscipline is pending against him, the result of the student (for that Semester) may be withheld and he will not be allowed to go into the next Semester. The award or issue of the Degree may also be withheld in such cases



15. Transfer of students from the Constituent Colleges of JNTUH or from other Colleges/ Universities: Transfer of students from the Constituent Colleges of JNTUH or from other Colleges/ Universities shall be considered only on case-to-case basis by the Academic Council of the Institute.

16. Transitory Regulations: Students who have discontinued or have been detained for want of attendance, or who have failed after having undergone the Degree Programme, may be considered eligible for readmission/re-registration to the same or equivalent subjects as and when they are offered.

17. General Rules

- a) The academic regulations should be read as a whole for the purpose of any interpretation.
- b) In the case of any doubt or ambiguity in the interpretation of the above rules, the decision of the Academic Council is final.
- c) In case of any error in the above rules and regulations, the decision of the Academic Council is final.
- d) The college may change or amend the academic regulations or syllabi at any time and the changes or amendments made shall be applicable to all the students with effect from the dates notified by the college/institute.





GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
Master of Computer Applications (MBA)

MCA- I Year, I Semester

Group	Sub-Code	Subject	Credits	Int	Ext	Marks
ES	GR14F5001	Mathematical Foundations of Computer Science	4	40	60	100
ES	GR14F5002	Computer Organization	4	40	60	100
ES	GR14F5003	Computer Programming	4	40	60	100
HS	GR14F5004	Probability and Statistics	3	40	60	100
HS	GR14F5005	Accounting and Financial Management	3	40	60	100
HS	GR14F5006	English Language Communication and Soft Skills	3	40	60	100
ES	GR14F5007	Computer Organization and IT Lab	2	40	60	100
ES	GR14F5008	Computer Programming Lab	2	40	60	100
Total			25	320	480	800

MCA- I Year, II Semester

Group	Sub-Code	Subject	Credits	Int	Ext	Marks
ES	GR14F5009	Data Structures and Algorithm Analysis	4	40	60	100
ES	GR14F5010	Java Programming	4	40	60	100
HS	GR14F5011	Organization Structure and Personnel Management	3	40	60	100
ES	GR14F5012	Operations Research	4	40	60	100
ES	GR14F5013	Operating System	4	40	60	100
ES	GR14F5014	Data Structures Lab	2	40	60	100
ES	GR14F5015	Java Programming Lab	2	40	60	100
HS	GR14F5016	Soft Skills Practices	2	40	60	100
Total			25	320	480	800



MBA- II Year, I Semester

Group	Sub-Code	Subject	Credits	Int	Ext	Marks
ES	GR14F5017	Advanced Unix Programming	4	40	60	100
ES	GR14F5018	Data Communication & Computer Networks	4	40	60	100
HS	GR14F5019	Management Information Systems	4	40	60	100
ES	GR14F5020	Software Engineering	4	40	60	100
ES	GR14F5021	Database Management Systems	4	40	60	100
ES	GR14F5022	Advanced Unix Programming Lab	2	40	60	100
ES	GR14F5023	Database Management Systems Lab	2	40	60	100
		Total	24	280	420	700

MCA- II Year, II Semester

Group	Sub-Code	Subject	Credits	Int	Ext	Marks
ES	GR14F5024	Web Technologies	4	40	60	100
ES	GR14F5025	Information Security	4	40	60	100
ES	GR14F5026	Data Ware Housing & Data Mining	4	40	60	100
		Elective - I	4	40	60	100
ES	GR14F5027	Scripting Languages				
ES	GR14F5028	E-Commerce				
ES	GR14F5029	Grid and Cloud Computing				
ES	GR14F5030	Computer Graphics				
		Elective - II	4	40	60	100
ES	GR14F5031	Rich Internet Applications				
ES	GR14F5032	Distributed Systems				
ES	GR14F5033	Software Project Management				
ES	GR14F5034	Soft Computing				
ES	GR14F5035	Web Technologies Lab	2	40	60	100
ES	GR14F5036	Data Mining and Information Security Lab	2	40	60	100
ES	GR14F5037	Mini Project Lab	2	40	60	100
		Total	26	320	480	800



MCA- III Year, I Semester

Group	Sub-Code	Subject	Credits	Int	Ext	Marks
ES	GR14F5038	Mobile Application Development	4	40	60	100
ES	GR14F5039	Software Testing Methodologies	4	40	60	100
ES	GR14F5040	Object Oriented Analysis & Design using UML	4	40	60	100
		Elective - III	4	40	60	100
ES	GR14F5041	Web Services				
ES	GR14F5042	Distributed Databases				
ES	GR14F5043	Big Data Analytics				
ES	GR14F5044	Ethical Hacking				
		Elective - IV	4	40	60	100
ES	GR14F5045	Design Patterns				
ES	GR14F5046	Information Retrieval Systems				
ES	GR14F5047	Human Computer Interaction				
ES	GR14F5048	Semantic Web And Social Networks				
ES	GR14F5049	Mobile Application Development Lab	2	40	60	100
ES	GR14F5050	Software Testing and UML Lab	2	40	60	100
		Total	24	280	420	700

MCA- III Year, II Semester

Group	Sub-Code	Subject	Credits	Int	Ext	Marks
ES	GR14F5051	Seminar	2	50	---	50
ES	GR14F5052	Dissertation / Project Work	24	0	---	A/B/C Grade
		Total	26	50	---	50





I-Year





GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

MATHEMATICAL FOUNDATION OF COMPUTER SCIENCE

Course Code: GR14F5001
I Year I Semester

L:3 T:1 P:0 C:4

Pre-requisites

- Learner must have knowledge on Mathematics.
- Should have well familiar with the concepts and terminology of Mathematical Concepts.
- Learner should have a good understanding of Differential Equations.

Course Objectives

- To provide the foundations on basic computer related concepts for a coherent development to the students for the courses like Fundamentals of Computer Organization, Data Structures, Design and Analysis of Algorithms, Computer Graphics and others.
- To Comprehend different Properties of Binary Relations on a Set theory, Reflexivity, Symmetry, Transitivity, Graphical representation of symmetric relations, transitive relations, Hasse diagram and their applications apart from that they also learn topics like Monoid, semi groups, Groups, Semi group, Homomorphism and Isomorphism systems
- To Develop skills in understanding and applying basic concepts on Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial & Multinomial theorems, the principles of Inclusion – Exclusion along with their applications.
- To Develop an appreciation for the use of Sequential functions and Calculating Coefficient of generating function, Characteristics roots, Solution of homogeneous, Recurrence Relation.
- To Design and Development of DFS, BFS and Spanning Trees, planar Graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers & their applications.

Course Outcomes

- Able to demonstrate knowledge on the foundations of many mathematical computer related concepts.
- Attained exposure to different Properties of Binary Relations subsequent to the course.
- Ability to think logically and mathematically on topics like Basis of counting Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients etc.



- Instill the belief that Mathematical logic is important for scientific research In Calculating Coefficient. Design and Development of Trees and Graphs & their applications

Unit-I

Mathematical Logic: Statements and notations, Connectives, Well formed formulas, Truth Tables, tautology, equivalence implication, Normal forms. Predicates: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving.

Unit-II

Set Theory: Properties of binary Relations, equivalence, compatibility and partial ordering relations, Hasse diagram. Functions: Inverse Function, Composition of functions, recursive Functions, Lattice and its Properties, Pigeon hole principles and its application.

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

Unit-III

Elementary Combinatorics: Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial and Multinomial theorems, the principles of Inclusion – Exclusion.

Unit-IV

Recurrence Relation: Generating Functions, Function of Sequences, Calculating Coefficient of generating function, Recurrence relations, Solving recurrence relation by substitution and Generating functions, Characteristics roots solution of non homogeneous Recurrence Relation.

Unit-V

Graph Theory: Representation of Graph, DFS, BFS, Spanning Trees, planar Graphs Graph Theory and Applications, Basic Concepts Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

Teaching Methodologies

1. Board
2. Markers
3. LCD Projector.

Text Books

1. Discrete Mathematical Structures with applications to computer science Trembly J.P. & Manohar .P, TMH.
2. Discrete and Combinational Mathematics- An Applied Introduction-5th Edition – Ralph. P.Grimaldi, Pearson Education.



3. Mathematical Foundations for Computer Science Engineers, JayantGanguly, Pearson Education
4. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition. TMH.

Reference Books

1. Discrete Mathematics with Applications, Thomas Koshy, Elsevier
2. Discrete Mathematical Structures, Bernard Kolman, Robert C. Busby, Sharn Cutter Ross, Pearson Education/PHI.
3. Discrete Mathematical structures Theory and application-Malik & Sen
4. Discrete Mathematics for Computer science, Garry Haggard and others, homson.
5. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott, A. Kandel, T.P. Baker Prentice Hall.
6. Logic and Discrete Mathematics, Grass Man & Trembley, Person Education.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

COMPUTER ORGANIZATION

Course Code: GR14F5002
I Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Fundamentals of computer system and digital electronics.

Course Objectives

- A clear understanding of different set of functional units interconnected within the system with operations, interactions and communication.
- An in-depth knowledge of different number systems represented at machine level.
- Knowledge of various addressing modes to address the CPU.
- Understand about the I/O organization of various peripheral devices and the importance of DMA
- Explain the functioning and programming of the INTEL-8086 microprocessor using assembly language programming.
- Significance of memory hierarchy and details of cache memory.

Course Outcomes: After this course completion students understand in depth of

- Details of the parts of a system and how data is stored and how it is manipulated using various circuits.
- I/O interaction in the system.
- Memory is organized.
- To write programs using assembly language.

Unit-I

Number Systems and Computer Arithmetic: Signed and unsigned numbers, Addition and subtraction, multiplication, division, Floating point representation, logical operation, Gray code, BCD codes, Error detecting codes, Boolean algebra, Simplification of Boolean expressions, K-Maps. Combinational And Sequential Circuits- decoders, Encoders, Multiplexers, Half and Full adders, Shift registers, Sequential circuits- flip-flops.

Unit-II

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



Unit-III

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

Unit-IV

Intel 8086 Assembly Language Instructions: Data transfer instructions-input-output instructions, address transfer, Flag transfer, arithmetic, logical, shift, and rotate instructions. conditional and unconditional transfer, iteration control, interrupts and process control instructions, assembler directives, Programming with assembly language instructions.

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

COMPUTER PROGRAMMING

Course Code: GR14F5003
I Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Student should have right aptitude, Logical Reasoning and Problem Solving ability.

Course Objectives

- To understand the various steps in Program development.
- To understand the basic concepts in C and C++ Programming Languages.
- Understand the concepts of functions, arrays, pointers, strings structures in C environment to solve structured problems
- Understand the concepts of classes, objects, constructors, inheritance, polymorphism in C++ to solve object oriented problems
- Develop C, C++ application programs using sound POP and OOP practices

Course Outcomes

- Demonstrate the basic knowledge of computer hardware and software
- Ability to apply solving and logical skills to programming in C and C++ languages
- Should have the ability to extend his knowledge of C/C++ further on his/her own.

Unit-I

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

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

Unit-II

Functions-Designing Structured Programs, Functions, user defined functions, inter function communication, Standard functions, Scope, Storage classes-auto,



register, static, extern, scope rules, type qualifiers, recursion- recursive functions, Limitations of recursion, example C programs, Preprocessor commands.

Arrays – Concepts, using arrays in C, inter function communication, array applications, two – dimensional arrays, multidimensional arrays, C program examples. Searching and Sorting – Sorting- selection sort, bubble sort, Searching-linear and binary search methods.

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

Unit-III

Pointers: Introduction (Basic Concepts), Pointers for inter function communication, pointers to pointers, compatibility, Pointer Applications-Arrays and Pointers, Pointer Arithmetic and arrays, Passing an array to a function, memory allocation functions, array of pointers, programming applications, pointers to void, pointers to functions.

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

Unit-IV

Input and Output: Concept of a file, streams, text files and binary files, Differences between text and binary files, State of a file, Opening and Closing files, file input / output functions (standard library input / output functions for files), file status functions (error handling), Positioning functions, C program examples. Different paradigms for problem solving need for OOP paradigm, classes and instances, fundamental characteristics of OOP, differences between OOP and Procedure Oriented Programming.

Introduction to C++ Basics, Structure of a C++ program, C++ Functions- Scope of variables, Parameter passing methods, Default arguments, inline functions, Recursive functions, C++ program examples.

Unit-V

C++ Classes And Data Abstraction: Class definition, Objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction, ADT and information hiding. Dynamic memory allocation and de-allocation operators-new and delete operators.

Inheritance: 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, C++ program examples.

Polymorphism: Function overloading, Function Overriding. Virtual Functions, Base and Derived class virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, C++ program examples.



Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. A Structured Programming Approach Using C, B.A.Forouzan and R.F. Gilberg, Third Edition, Cengage Learning.
2. Programming in C. P. Dey and M Ghosh , Oxford University Press.
3. C, The Complete Reference, 4th Edition, Herbert Schildt, TMH.
4. C++, The Complete Reference, 4th Edition, Herbert Schildt, TMH.
5. Programming Principles and Practice Using C++, B.Stroutstrup, Addison- Wesley, Pearson Education.
6. Object Oriented Programming With C++,Balagurusamy, Tata McGraw-Hill Education



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

PROBABILITY AND STATISTICS

Course Code: GR14F5004
I Year I Semester

L:2 T:1 P:0 C:3

Prerequisites

- Fundamentals in Basic Mathematics.

Course Objectives: At the end of the course the student is expected to

- Know the fundamentals of Probability and Statistics.
- Understand and apply the Tests of Hypothesis, Correlation & Regression.
- Understand simple Queuing models.

Course Outcomes: Students will learn

- Strong basics of Probability.
- Applications of Probability distributions.
- Application of regression analysis to analyse a problem.
- Application of inferential statistics.

Unit-I

Probability: Basic concepts in Probability - Conditional probability– Addition and Multiplication theorems for two events (Concepts and problem solving) – Baye's theorem.

Random Variables: Definition of a random variable, discrete and continuous random variables – Distribution function and statements of its properties, probability mass function, probability density function with illustrations - Expectation and variance of a r.v and statements of their properties.

Unit-II

Distributions: Binomial, Poisson, Uniform, Normal and Exponential distributions (definition, real life examples, Statements of their Mean, Mode and Variance and problems). Fitting of Binomial and Poisson distributions. Sampling distribution: Definition of Population and sample, Overview of types of sampling(Purposive, Random, SRS with and without replacement cases, Stratified and Systematic random samplings) - Sampling distribution, standard error, statements of sampling distribution of mean(s) (Population variance(s) known and unknown) and proportion(s) (Population proportion(s) known and unknown) with examples.



Unit-III

Estimation & Testing of Hypothesis: Definitions of Point and Interval estimation. Confidence intervals for single mean, difference of two means, single proportion and difference of two proportions. Concepts of Null and Alternative hypotheses, Critical region, Type I and Type II errors, one tail and two-tail tests, Level of significance.

Large Samples Tests: Tests of hypothesis for mean(s) (single and difference between means), Tests of hypothesis for proportion(s) (single and difference between proportions), Chi-square test for testing goodness of fit, independence of attributes and single population variance.

Unit-IV

Small samples: Student's t-test for testing the significance of single mean, difference of means(independent samples and paired samples), F-test for equality of variances (Concepts and problem solving) .

Unit-V

Correlation & Regression: Fitting of straight line , second degree parabola and exponential curves by least squares method- Product moment correlation coefficient , test for its significance and statements of properties, Spearman's rank correlation coefficient and statement of its properties – Simple linear regression, Lines of Regression, Regression coefficients and statement of their properties, Multiple regression for three variables only.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Probability and statistics for engineers (Erwin Miller and John E.Freund), R.A.Johnson and C.B.Gupta, Pearson education.
2. Probability and Statistics, Dr.T.K.V.Iyengar, Dr.B.Krishna Gandhi et.al, S.Chand.

Reference Books

1. Introduction to Probability and Statistics, 12th edition, W.Mendenhall, R.J.Beaver and B.M.Beaver, Thomson. (Indian edition)
2. Probability and statistics, Murray R Spiegel, John J Schiller et al, Schaum's series.
3. Fundamentals of Mathematical Statistics, S.C.Gupta ,V.K.Kapoor, S.Chand.



**GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

ACCOUNTING AND FINANCIAL MANAGEMENT

Course Code: GR14F5005
I Year I Semester

L:2 T:1 P:0 C:3

Prerequisites

- Student should have Logical Reasoning and Problem Solving ability.

Course Objectives

- To provide necessary basic inputs and tools on maintaining books of accounts, preparation of financial statements, analysis and interpretation of financial statements. It is to provide the basic inputs to manage the finance functions.

Course Outcomes

- After the completion of the course the students should be able to understand and prepare financial statements. Students should be able to understand the management of finances of the business organization.

Unit-I

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

Unit-II

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

Unit-III

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



Unit-IV

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

Unit-V

Capital Management: Components of working capital, gross vs. net working capital. Determinants of working capital needs, the operating cycle planning of WC. Capital Budgeting: features of capital budgeting proposals, Methods of Capital Budgeting: Payback Method, Accounting Rate of Return (ARR) and Net Present Value Method (simple problems)

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

ENGLISH LANGUAGE AND COMMUNICATION SKILLS

Course Code: GR14F5006
I Year I Semester

L:2 T:1 P:0 C:3

Prerequisites

- Should be familiar with spoken English and written English
- Should have heard lectures in English previously

Course Objectives

- The course covers verbal, non-verbal, written and other forms of communication. The objectives of the course are to make the students familiar with the premises and principles of different ways and forms of communication in a business context and sharpen their communication skills.

Course Outcomes: The student should be able to

- Write and speak simple sentences which are grammatically correct.
- Be able to relate with others with official personal demeanour.
- Listen to and absorb conversations and seminars in English.
- Write letters to communicate officially.
- Write project reports in appropriate format.

Unit-I

Features of Indian English: Correction of sentences – Informal conversation Vs Formal expression – Verbal and non-verbal communication, barriers to effective communication – kinesics – Types of communication – Listening, Speaking, Reading and Writing, Telephone etiquette.

Unit-II

Non-Verbal and Intercultural Communication: Importance of non-verbal communication - personal appearance - facial expressions- movement- posture – gestures - eye contact –voice - beliefs and customs- worldview and attitude.

Unit-III

Oral Communication Listening: Types and barriers to listening - speaking - planning and audience awareness - persuasion- goals - motivation and hierarchy of needs - attending and conducting interviews participating in discussions, debates - and conferences - presentation skills- paralinguistic features -fluency development strategies



Unit-IV

Official Correspondence Business letter: Principles of business writing- memos -e-mails – agendas- minutes- sales letter- enquiries- orders- letters of complaint- claims and adjustments- notice and tenders, circulars-letters of application and resume.

Unit-V

Project proposals: Characteristics and structure- Project reports – types - characteristics – structure - Appraisal reports – performance appraisal, product appraisal- Process and mechanics of report writing- visual aids- abstract - executive summary- recommendation writing- definition of terms.

Reference Books

1. Essentials of Business Communication, Rajendra Pal, JS Korlaha Sultan Chand & Sons, New Delhi.
2. Basic Communication Skills for Technology, Andrea J. Rutherford: Pearson Education Asia, Patparganj, New Delhi-92.
3. Communication Skills, V. Prasad, Atma Ram Publications, New Delhi.
4. Raymond V. Lesikav; John D.Pettit Jr.; Business Communication; Theory &Application, All India Traveller Bookseller, New Delhi-51.
5. Business Communication, RK Madhukar, Vikas Publishing House Pvt. Ltd.
6. K.R. Lakshminarayana : English for Technical Communication – vols. 1 and2, SCITECH Publications (India) Pvt. Ltd., T. Nagar, Chennai-600 017.
7. Edmond H Weiss: Writing remedies : Practical Exercises for Technical Writing, Universities Press, Hyderabad.
8. Dr. P. Eliah, A. Handbook of English for Professionals, third ed., BSPublications, Hyderabad,2011.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

COMPUTER ORGANIZATION LAB and IT LAB

Course Code: GR14F5007
I Year I Semester

L:0 T:0 P:2 C:2

Prerequisites

Computer Organization Lab

- Basic knowledge of the instructions of a microprocessor.
- Familiarity with the assembly language programming language.

IT Workshop

- Familiarity of computer hardware and software.
- Learner should have the basic knowledge of working on the system with MS-Office and Internet.

Course Objectives: Computer Organization Lab

- To provide a clear understanding of the instructions used in writing the program (8086 processor) for solving problems.
- To teach how to save, compile, execute a program.
- IT Workshop-The IT Workshop for engineers is a training lab course spread over 54 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel and Power Point.

Course Outcomes: Computer Organization

- Learner is aware of different commands usage and how the data is stored in different registers.
- Able to write programs using assembly language.

IT Workshop

After this course completion students understand in depth of

- The various internal parts of a computer and how the connection are given to these parts.
- Able to perform assembling and disassembling of various parts of the system.
- Able to install different OS softwares.
- Able to explore on internet.
- Able to make use of MS-Office in an efficient manner.

Computer Organization Lab

List of Sample Problems:

Write assembly language programs for the following using TASM.



1. Write assembly language programs to evaluate the expressions:
 - i) $a = b + c - d * e$ ii) $z = x * y + w - v + u / k$
 - a. Considering 8-bit, 16 bit and 32 bit binary numbers as b, c, d, e.
 - b. Considering 2 digit, 4digit and 8 digit BCD numbers. Take the input in consecutive memory locations and results also display the results by using "int xx" of 8086. Validate program for the boundary conditions.
2. Write an ALP of 8086 to add two exponential numbers which are in IEEE 754 notation. Display the results by using "int xx" of 8086. Validate program for the boundary Conditions.
3. Write an ALP of 8086 to take N numbers as input. And do the following operations on them.
 - a) Arrange in ascending and descending order.
 - b) Find max and minimum
 - c) Find average
 Considering 8-bit, 16 bit binary numbers and 2 digit, 4digit and 8 digit BCD numbers display the results by using "int xx" of 8086. Validate program for the boundary conditions.
4. Write an ALP of 8086 to take a string of as input (in 'C' format) and do the following Operations on it.
 - a) Find the length
 - b) Find it is Palindrome or not
 - c) Find whether given string substring or not.
 - d) Reverse a string
 - e) Concatenate by taking another string Display the results by using "int xx" of 8086.
5. Write the ALP to implement the above operations as procedures and call from the main procedure.
6. Write an ALP of 8086 to find the factorial of a given number as a Procedure and call from the main program which display the result.
7. Write an assembly language program to encrypt digits as shown below:
 Input digit: 0 1 2 3 4 5 6 7 8 9 Encrypted digit: 4 6 9 5 0 3 1 8 7 2
 Your program should accept a string consisting of digits. The encrypted String should be displayed using "int xx" of 8086.
8. Write a procedure to locate a character in a given string. The procedure receives a pointer to a string and character to be located. When the first occurrence of the character is located its position is returned to main. If no match is found, a negative value is returned. The main procedure requests a character string and a character to be located and displays the result.
9. Write an assembly language program to read a string of characters from the user and that prints the vowel count . Display the results by using "int xx" of 8086. ex. Input: Advanced Programming in UNIX

**Out put:**

Vowel count

a or A 3e or E 1

i or I 3

o or O 1

u or U 1

10. A computer uses RAM chips of 1024 X 1 capacity.
 - i. How many chips are needed, and how should their address lines be connected to provide a memory capacity of 1024 bytes?
 - ii. How many chips are needed to provide a memory capacity of 16K bytes?
11. A computer employs RAM chips of 256X8 and ROM chips of 1024 X 8. The computer needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, 10 for interface registers.
 - a. How many RAM and ROM chips are needed?
 - b. Draw a memory-address map for the system.
 - c. Give the address range in hexadecimal for RAM, ROM and interface.
12. Obtain the complement function for the match logic of one word in an associative memory. Draw the logic diagram for it and compare with the actual match logic diagram.
13. A two-way set associative cache memory uses blocks of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128K X 32.
 - a. Formulate all pertinent information required to construct the cache memory.
 - b. What is the size of the cache memory?
14. A digital computer has a memory unit of 64K X 16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.
 - a. How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.
 - b. How many bits are there in the tag, index, block, and word fields of the address format?
 - c. How many blocks can the cache accommodate?
15. An address space is specified by 24 bits and the corresponding memory space by 16 bits.
 - a. How many words are there in the address space?
 - b. How many words are there in the memory space?
 - c. If a page consists of 2K words, how many pages and blocks are there in the system.
16. A virtual memory has a page size of 1K words. There are eight pages and four blocks. The associative memory page table contains the following entries. Make



a list of all virtual addresses(in decimal) that will cause a page fault.

Page Block

03

11

42

60

IT WORKSHOP

PC Hardware introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible.

Internet & World Wide Web module introduces the different ways of hooking the PC on to the internet from home and workplace and effectively usage of the internet. Usage of web browsers, email, newsgroups and discussion forums would be covered. In addition, awareness of cyber hygiene, i.e., protecting the personal computer from getting infected with the viruses, worms and other cyber attacks would be introduced.

Productivity tools module would enable the students in crafting professional word documents, excel spread sheets and power point presentations using the Microsoft suite of office tools and LaTeX.

(Recommended to use Microsoft office 2007 in place of Libla office/ubuntu)
PC Hardware

Week 1 - Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Week 2 - Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Week 3 - Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.



Week 4 - Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva

Week 5 - Task 5: Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva

Week 6 - Task 6: Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Internet & World Wide Web

Week 7 - Task 1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

Week 8 - Task 2: Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

Week 9 - Task 3: Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

Week 10 - Task 4: Cyber Hygiene: Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to first install antivirus software, configure their personal firewall and windows update on their computer. Then they need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

Week 11- Task 5: Develop your home page using HTML Consisting of your photo, name, address and education details as a table and your skill set as a list.



Productivity tools

LaTeX and Word

Week 12 – Word Orientation: The mentor needs to give an overview of LaTeX and Microsoft (MS) office 2007/ equivalent (FOSS) tool word: Importance of LaTeX and MS office 2007/ equivalent (FOSS) tool Word as word Processors, Details of the three tasks and features that would be covered in each, using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter.

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

Week 13 - Task 2: Creating project abstract Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell-check, Track Changes.

Week 14 - Task 3: Creating a Newsletter : Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

Excel

Week 15 - Excel Orientation: The mentor needs to tell the importance of MS office 2007/ equivalent(FOSS) tool Excel as a Spreadsheet tool, give the details of the two tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

Task 1: Creating a Scheduler - Features to be covered: - Gridlines, Format Cells, Summation, auto fill, Formatting Text

Week 16 - Task 2: Calculating GPA - .Features to be covered:- Cell Referencing, Formulae in excel –average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP, Sorting, Conditional formatting LaTeX and MS/equivalent (FOSS) tool Power Point

Week 17 - Task1: Students will be working on basic power point utilities and tools which help them create basic power point presentation. Topic covered during this week includes :- PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in both LaTeX and Power point. Students will be given model power point presentation which needs to be replicated (exactly how it's asked).

Week 18 - Task 2: Second week helps students in making their presentations interactive. Topic covered during this week includes: Hyperlinks, Inserting –Images, ClipArt, Audio, Video, Objects, Tables and Charts



Week 19 - Task 3: Concentrating on the in and out of Microsoft power point and presentations in LaTeX. Helps them learn best practices in designing and preparing power point presentation. Topic covered during this week includes: - Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), Inserting – Background, textures, Design Templates, Hidden slides.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI.
2. Introduction to Assembly Language Programming, Sivarama P.Dandamudi, Springer Int. Edition, 2003.
3. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and
4. Application, 4th edition, W.A.Triebel, A.Singh, N.K.Srinath, Pearson Education.
5. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
6. LaTeX Companion – Leslie Lamport, PHI/Pearson.
7. Introduction to Computers, Peter Norton, 6/e Mc Graw Hill

Reference Books

1. Upgrading and Repairing, PC's 18th e, Scott Muller QUE, Pearson Education
2. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dreamtech
3. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
4. PC Hardware and A+ Handbook – Kate J. Chase PHI (Microsoft)



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

COMPUTER PROGRAMMING LAB

Course Code: GR14F5008
I Year I Semester

L:0 T:0 P:2 C:2

Prerequisites

- Fundamentals of Algorithms, Flow charts.
- Fundamentals of editing, compiling, executing using editors.

Course Objectives

- To learn basic algorithms and flowcharts both numeric and string concepts.
- To write simple C and C++ programs that incorporate different data types, different types of variables, expressions, selection, and iteration.
- To write structured programs using arrays, functions, strings, pointers, structures and unions.
- To write object oriented programs using classes, objects, constructors, functions using C++ syntax, inheritance and polymorphism concepts.

Course Outcomes

- Ability to write programs for different kinds of problems in C and C++.
- Work confidently in compilers like C and C++ compilers in various operating systems.
- Understand and find the output of simple C programs that incorporate different types of variables, expressions, selection, and iteration.
- Understand and find the output of more complex C programs containing arrays and invoking (calling) functions having input and output arguments using pointers.
- Understand and find the output of more complex C++ programs containing functions, classes and objects, constructors and destructors, inheritance and polymorphism.
- Understand different ways of memory allocations and deallocation mechanisms using C and C++.
- Understand and find the output of different file handling mechanisms using C and C++.

I. Programs using C language

1. Write a C Program to Print a Sentence
2. Write a C Program to Add Two Integers
3. Write a C Program to Find Quotient and Remainder of Two Integers Entered by User



4. Write a C Program to Swap Two Numbers
5. Write a C Program to Check Whether a Number is Even or Odd
6. Write a C Program to Check Vowel or Consonant
7. Write a C Program to Find the Largest Number Among Three Numbers
8. Write a C program to Find all Roots of a Quadratic equation
9. Write a C Program to Check Leap Year
10. Write a C Program to Check Whether a Number is Positive or Negative or Zero.
11. Write a C Program to Check Whether a Character is an Alphabet or not
12. Write a C Program to Calculate Sum of Natural Numbers
13. Write a C Program to Find Factorial of a Number
14. Write a C program to Generate Multiplication Table
15. Write a C Program to Display Fibonacci Series
16. Write a C Program to Find HCF of two Numbers
17. Write a C Program to Find LCM of two Numbers
18. Write a C Program to Count Number of Digits of an Integer
19. Write a C Program to Reverse a Number
20. Write a C program to Calculate the Power of a Number
21. Write a C Program to Check Whether a Number is Palindrome or Not
22. Write a C Program to Check Whether a Number is Prime or Not
23. Write a C Program to Display Prime Numbers Between Two Intervals
24. Write a C program to Check Armstrong Number
25. Write a C Program to Display Armstrong Number Between Two Intervals
26. Write a C program to Display Factors of a Number
27. Write a C program to Make a Simple Calculator to Add, Subtract, Multiply or Divide Using switch...case
28. Write a C Program to Display Prime Numbers Between Intervals by Making Function
29. Write a C Program to Check Prime and Armstrong Number by Making Function
30. Write a C program to Check Whether a Number can be Express as Sum of Two Prime Numbers
31. Write a C program to Find Sum of Natural Numbers using Recursion.
32. Write a C program to Calculate Factorial of a Number Using Recursion
33. Write a C Program to Find H.C.F Using Recursion
34. Write a C program to Reverse a Sentence Using Recursion
35. Write a C program to Calculate the Power of a Number Using Recursion
36. Write a C Program to Convert Binary Number to Decimal and Decimal to



Binary

37. Write a C Program to Calculate Average Using Arrays
38. Write a C Program to Find Largest Element of an Array
39. Write a C Program to Add Two Matrix Using Multi-dimensional Arrays
40. Write a C Program to Multiply to Matrix Using Multi-dimensional Arrays
41. Write a C Program to Find Transpose of a Matrix
42. Write a C Program to Multiply two Matrices by Passing Matrix to Function
43. Write a C Program to Sort Elements of an Array
44. Write a C Program to Access Elements of an Array Using Pointer
45. Write a C Program to Find Largest Number Using Dynamic Memory Allocation
46. Write a C Program to Find the Frequency of Characters in a String
47. Write a C Program to Find the Number of Vowels, Consonants, Digits and White space in a String
48. Write a C Program to Remove all Characters in a String Except Alphabet
49. Write a C Program to Reverse a String by Passing it to Function
50. Write a C Program to Find the Length of a String
51. Write a C program that counts the characters, lines and words in the text file.
52. Write a C program to Concatenate Two Strings
53. Write a C Program to Copy a String
54. Write a C Program to Sort Elements in Lexicographical Order (Dictionary Order)
55. Write a C Program to Store Information(name, roll and marks) of a Student Using Structure
56. Write a C Program to Add Two Distances (in inch-feet) System Using Structures
57. Write a C Program to Add Two Complex Numbers by Passing Structure to a Function
58. Write a C Program to Calculate Difference Between Two Time Period
59. Write a C Program to Store Information of 10 Students Using Structure
60. Write a C Program to Store Information Using Structures for n Elements Dynamically
61. Write a C++ program to display the contents of a text file.
62. Write a C Program to Copy the contents from one file to another.
63. Write a C program that counts the characters, lines and words in the text file.



II .Programs using C++ language

1. Write a C++ Program to implement a Class STUDENT having following members: Data members as Student number,name,marks in three subjects. Member functions as computeTotal and computeAverage and display the data.
2. Write a C++ Program to Design a Class for a Bank having following members: Data members as acno, acname, balance. Member functions as getdata(), showdata(), deposit(), withdraw(). Use constructors & destructors whenever necessary.
3. Write a C++ program to implement copy constructor mechanism. Use default and parameterized constructors to initialize data members of a class.
4. Write a C++ Program to demonstrate the Function Overloading.
5. Write a C++ Program to demonstrate Friend Function and Friend Class.
6. Write a C++ Program to demonstrate the use of Static data members and Static member functions.
7. Write a C++ Program to demonstrate the concepts of callby value,call by reference mechanisms.
8. Write C++ programs that illustrate how the following forms of inheritance are supported:
a) Single inheritance b) Multiple inheritance c) Multi level inheritance
9. Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions.
10. Write a C++ program that illustrates the role of virtual base class in building class hierarchy.

Teaching methodologies

1. Lab experiments with GCC compiler on Linux/Turbo C & C++ on Dev C++ on Windows O.S.
2. Discussion on white board and observation books.

Text Books

1. Computer Programming in C, V. Rajaraman, PHI
2. C Programming, E.Balagurusamy, 3rd edition, TMH.
3. Mastering C, K.R. Venugopal and S.R. Prasad, TMH Publications.
4. Object Oriented programming with C++,E.Balagurusamy, 4th edition, TMH.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA STRUCTURES AND ALGORITHM ANALYSIS

Course Code: GR14F5009
I Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Learner must have completed a course on Computer Organization and Understand the terminology of computer.
- Learner should have a good understanding of Computer programming and its concepts preferably “C” language.

Course Objectives

- To understand the basic concepts of Linear and Non Linear Data structures.
- To understand the notations and computation of Performance analysis of computer algorithms.
- To understand the representation and applications of data structures such as stacks, queues, trees, and Graphs.
- To understand and analyze various searching and sorting algorithms.

Course Outcomes

- Capable of analysing the time and space complexity for a given algorithm. And find ways to improve the efficiency of algorithms.
- Ability to identify appropriate data structure for solving given problem
- Able to apply suitable design strategies (greedy, divide and conquer etc.) for a given problem.
- Facilitates learner to comprehend these concepts in various courses of this programme.

Unit-I

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

Linear Data Structures: Linear Lists, Sequential and Linked allocation, The list ADT, array and Linked Implementations, Singly Linked Lists-Operations-Insertion, Deletion

Doubly Linked Lists: Operations-Insertion, Deletion.

Stack ADT: Definitions, Operations, Array and linked implementations, Applications infix to postfix conversion, recursion implementation

Queue ADT: Definitions and Operations, Array and Linked Implementations, Dequeue.



Unit-II

Non linear data structures: Trees – Basic Terminology, Binary tree ADT, array and linked representations, traversals recursive and non-recursive, Threaded Binary Trees,

Binary Search Trees: ADT, Operations-Searching, Insertion and Deletion, implementation Priority Queues-Definition, ADT, Realizing a Priority Queue using Heap.

Unit-III

AVL Trees: Definition, Operations – Insertion and Searching.

B-Trees: Definition, B-Tree of order m, operations- insertion and deletion, Disjoint Sets- Union and Find algorithms Graphs: Introduction, Basic Terminology, Graph Representations- Adjacency matrix, Adjacency lists, Adjacency multi-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, Heap Sort, Radix Sort. Algorithm Design method- Divide and Conquer method-applications- Quick sort, Merge sort, Comparison of Sorting methods, Model for External Sorting.

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Data Structures and Algorithms in Java, 2nd edition, M.T.Goodrich and R.Tamassia, John Wiley and Sons, Inc.
2. Data Structures and Algorithms in Java, 2nd edition, A.Drozdek, Thomson.
3. Data Structures and Software Development in an Object Oriented Domain, Java Edition, Tremblay, Pearson Education.

Reference Books

1. Data Structures and Java Collections Frame Work, W.J.Collins, McGraw Hill.
2. Data Structures using Java, Yeddyiah Langsam, Moshe Augenstein, Aaron M.Tenenbaum, Pearson Education.
3. Data Structures with Java, J.R.Hubbard and A.Huray, Pearson



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

JAVA PROGRAMMING

Course Code: GR14E5014
I Year II Semester

L:4 T:0 P:0 C:4

Pre-requisites

- Students should have basic computer literacy including the use of a text editor, be familiar with the Internet, with the World Wide Web.
- Experience with directory structures, saving and retrieving files, and web browsers.
- Should have basic knowledge of C & C++

Course Objectives

- Understand the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, and encapsulation.
- Identify classes, objects, members of a class and the relationships among them needed for a specific problem.
- Develop Java application programs using sound OOP practices(ex. Interfaces and APIs)
- Develop programs using the Java collection APIs as well as Java standard class library.
- Develop programs using Multithreading and Networking for operating system based and distributed based applications.

Course Outcomes

- Understanding of OOP concepts and basics of java programming (Console and GUI based)
- Describe the use of packages and interfaces
- Demonstrate understanding of exception handling
- Demonstrate understanding of Java's byte and character streams for input and output
- Demonstrate understanding of multithreaded programming
- Demonstrate understanding of applets, events and keywords.
- Should have the ability to extend his knowledge of Java programming further on his/her own.

Unit-I

Java Basics: History of Java, Java buzzwords, comments, data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow-block scope, conditional statements, loops, break and continue statements,



simple java program, arrays, input and output, formatting output, Review of OOP Concepts, encapsulation, inheritance, polymorphism, classes, objects, constructors, methods, parameter passing, static fields and methods, access control, this reference, overloading methods and constructors, recursion, garbage collection, building strings, exploring string class, Enumerations, autoboxing and unboxing, Generics.

Unit-II

Inheritance: Inheritance concepts, benefits of inheritance, Super classes and Subclasses, Member access rules, Inheritance hierarchies, super uses, preventing inheritance: final classes and methods, casting, **Polymorphism-**dynamic binding, method overriding, abstract classes and methods, the Object class and its methods Interfaces : Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations Through interface references, extending interface.

Inner classes: Uses of inner classes, local inner classes, anonymous inner-classes, static inner classes, examples.

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

Unit-III

Data structures creation and manipulation in java: Introduction to Java Collections, Overview of Java Collection frame work, Commonly used Collection classes - ArrayList, LinkedList, HashSet, HashMap, TreeMap, Collection Interfaces- Collection, Set, List, Map, Legacy Collection classes - Vector, Hashtable, Stack, Dictionary (abstract), Enumeration interface, Iteration over Collections - Iterator, Interface, List Iterator interface. Other Utility classes - String Tokenizer, Formatter, Random, Scanner, Observable, Using java.util.Files: streams- byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class, Using java.io. Networking in Java: Introduction, Manipulating URLs, Ex. Client/Server Interaction with Stream Socket Connections, Connectionless Client/Server Interaction with Datagrams, Using java.net.

Unit-IV

Exception handling: Dealing with errors, benefits of exception handling, the classification of exceptions-exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes, Guide lines for proper use of exceptions.

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



Unit-V

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

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

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Java: the complete reference, 7th editon, Herbert Schildt, TMH.
2. Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education / Java How to Program P.J.Deitel and H.M.Deitel ,8th edition, PHI.

Reference Books

1. Java Programming, D.S.Malik, Cengage Learning.
3. Core Java, Volume 1-Fundamentals, eighth edition, Cay S.Horstmann and Gary Cornell, pearson education.
4. An introduction to Java programming and object oriented application development, R.A. Johnson- Cengage Learning.
5. Advanced Programming in Java2, K.Somasundaram, Jaico Publishing House.
6. Starting out with Java, T.Gaddis, dreamtech India Pvt. Ltd.
7. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
8. Object Oriented Programming through Java, P.Radha Krishna, Universities Press.
9. An introduction to programming and OO design using Java, J.Nino, F.A.Hosch, John Wiley&Sons.
10. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
11. Maurach's Beginning Java2, D.Lowe, J.Murach, A.Stelman, SPD.
12. Programming with Java, M.P.Bhave, S.A.Patekar, Pearson Education



**GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

ORGANIZATION STRUCTURE AND PERSONNEL MANAGEMENT

Course Code: GR14F5011
I Year II Semester

L:2 T:1 P:0 C:3

Prerequisites

- Familiarity with Organizational hierarchy involved in some organization.
- Familiarity with organization functions.
- Familiar with Some principles involved in interpersonal relationships in day to day life.

Course Objectives

- To understand basic management concepts and use of management principles in the organization
- To understand function of management and organization structure.
- To acquire the knowledge of quality management practices.
- This course also covers Personnel Management and Communication.
- To understand the organizational behaviour.

Course Outcomes

Upon Completion of the course, the students should be able to:

- Understand the importance of organizational structure and design on internal organizational processes and overall effectiveness.
- Understand the relationships between organization structure and the behaviour of those who work in them or otherwise interact with them.
- Understand the Personnel Functions like position of the personnel department in the organization.
- Understand manpower planning, job description, interviewing techniques, transfers, promotion and its policies.
- Understand the training and development and career planning and Performance Appraisal.

Unit-I

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



Unit-II

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

Unit-III

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

Unit-IV

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

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

1. Udai pareek: Understanding Organizational Behaviors, PHI, 2009.
3. P.Subbarao, Management and Organizational Behaviors, HPH, 2009.
4. Mamoria & Gankar, Personnel Management, HPH, 2009.
5. VSP Rao, Organizational Behaviors, Excel, 2009.
6. Lawrence R Jauch, R.Gupta &William F.Glueck: Business Policy and Strategic Management, Fran Bros.2009.
7. Sharma:Organisational Behaviors, Jaico, 2009.
8. Koontz & Weihrich: Essentials of Management, TMH, 2009
9. Mullins: Management and Organizational Behaviors, Pearson, 2009.
10. R Satyaraju & A.Parthasarathy: Management Text and Cases, PHI, 2009.
11. Prem Vrat, K K Ahuja, P K Jain, Case Studies in Management, Vikas Publishing House Pvt. Ltd., 2006.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

OPERATIONS RESEARCH

Course Code: GR14F5012
I Year II Semester

L:3 T:1 P:0 C:4

Prerequisites

- Fundamentals of mathematics

Course Objectives: At the end of the course the student is expected to

- Formulate mathematical models to real world problems.
- Understand and apply routing problems.
- Understand simple Queuing models.

Course Outcomes: Students will learn

- Strong basics of optimization techniques.
- Basics of LPP models.
- Applications of replacement models.
- Applications of queuing models.

Unit-I

Introduction to Operations Research: 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 unbounded solutions, dual simplex method.

Unit-II

Transportation problem: Transportation problems and its Formulation, solution. Finding initial basic feasible solutions: North-west corner rule, least cost method and Vogel's approximation method. Optimality tests: Stepping stone and MODI methods. Solution of unbalanced transportation problem

Assignment Models: Assignment problem and its Formulation. Hungarian method for optimal solution. Solving unbalanced problem. Travelling salesman problem as Assignment problem.

Sequencing models: Solution of sequencing problem – Johnson's Algorithm for processing n jobs through two machines, processing n jobs through 3 machines-processing 2 jobs through m machines, processing n jobs through m machines.

Unit-III

Replacement Models: Replacement of items that deteriorate whose maintenance cost increase with time with and without change in the money



value. Replacement of items that fail suddenly: individual replacement, group replacement.

Dynamic Programming: Characteristics of Dynamic programming, Bellman's principle of optimality. Dynamic programming approach for stage coach and shortest path.

Game theory: competitive games, Two-person zero sum game, rectangular game, saddle point, min-max (max-min) method of optimal strategies, value of the game. Solution of games with saddle point, Dominance principle. Rectangular games without saddle point – mixed strategy for 2X2 games. Solution of 2Xn, mX2 games by Graphical method.

Unit-IV

Necessity of maintaining inventory, inventory costs and inventory control problem, selective inventory management techniques- ABC analysis, VED analysis, EOQ, Inventory models:

1. Classical EOQ model (Demand rate uniform, replenishment rate infinite.)
2. Demand rate uniform, production rate finite, shortages allowed.
3. (a) instantaneous models with probabilistic models.
4. Instantaneous demand, no setup cost, stock levels, continuous, lead time is zero

Unit-V

Queuing theory: Queue description, characteristics of a queuing model, Poisson process, concept of Birth and death process, steady state solutions of (M/M/1: ∞ /FIFO) and (M/M/1: N/FIFO).

Teaching Methodologies

1. White Board and Marker
2. power point presentations

Text Books

1. S.D.Sharma "Operations Research" .
2. N.D. Vohra "Quantitative Techniques in Management "TATA MCGRAW HILL publications.

Reference Books

1. P.K. Gupta and D. S.Hira , "Operations Research" S. Chand publications
2. J.K. Sharma, "Operations Research–theory and Applications". Macmillan publishers India.
3. P.Shankaralyer- "Operations Research" TATA MCGRAW HILL



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

OPERATING SYSTEMS

Course Code: GR14F5013
I Year II Semester

L:3 T:1 P:0 C:4

Prerequisites

- Basic knowledge of computer hardware and software.
- Knowledge of some of the data structure concepts.
- Familiarity of atleast one operating system.

Course Objectives

- Clear understanding of the concepts of OS, its components, types and their working.
- Clear insight of the scheduling policies of OS .
- Understand process concurrency and synchronization.
- A thorough understanding of various memory management techniques.
- Clear insight of secondary storage devices and file management.
- To understand the operations performed by OS as a resource manager.
- Understand the various protection techniques provided by OS.
- To study different OS and compare their features.

Course Outcomes: The course will help each student to better:

- Understand how the OS manages all the processes and handles the different IO operations.
- Understand process coordination.
- Learn to minimize turnaround time, waiting time and response time and also maximization of throughput keeping CPU as busy as possible.
- Appreciate the nuances of different allocation techniques.
- The importance of memory management using different techniques.
- Applying optimization techniques for the improvement of system performance.
- Understand how the OS is keeping a track of the resources among all processes without the occurrence of deadlock with different algorithms.
- Understand how protection is provided by OS.
- Ability to compare different Operating Systems.

Unit-I

Operating System Introduction: Operating Systems objectives and functions, Computer System Architecture, OS Structure, OS Operations. Evolution of Operating Systems: Simple Batch, Multi programmed, time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special -Purpose



Systems, Operating System services, User OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Virtual Machines.

Unit-II

Process and CPU Scheduling: Process concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies Linux, Windows. Process Coordination: Process Synchronization, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

Unit-III

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

Unit-IV

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

Case Studies: Linux, Windows. Mass Storage Structure: Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management.

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations



Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA STRUCTURES ALGORITHM ANALYSIS LAB

Course Code: GR14F5014
I Year II Semester

L:0 T:0 P:2 C:2

Prerequisites

- Learner must understand the terminology of computer.
- Should be able to code simpler programs using a programming Language preferably “C” language.

Course Objectives

- To write and execute programs using arrays and Linked List to implement data structures linked.
- To write and execute write programs in C to implement various sorting and searching methods.

Course Outcomes

- Able to understand and identify the appropriate data structure for given problem.
- Ability to design and analyze the time and space complexity of algorithm or program.
- Ability to effectively use compilers includes library functions, debuggers and trouble shooting.

List of Sample Problems/Experiments

1. Write a program to perform the following operations:
a) Create a SLL of integers. b) Delete an integer from SLL.
2. Write a program to perform the following operations:
a) Create a DLL of integers. b) Delete an integer from DLL.
3. Write programs to implement the following using an array.
a) StackADT b) QueueADT
4. Write programs to implement the following using a singly linked list.
a) StackADT b) QueueADT
5. Write a program to convert a given infix expression into postfix form using stack.
6. Write programs to implement the deque (double ended queue) ADT using DLL and an array.



7. Write programs that use recursive functions to traverse the given binary tree in
 - a) Preorder
 - b) inorder and
 - c) postorder
8. Write programs that use non-recursive functions to traverse the given binary tree in
 - a) Preorder
 - b) inorder and
 - c) postorder
9. Write a program to perform the following operations:
Insert an element into a BST
Delete an element from a BST
Search for a key element in a BST
10. Write programs to Implement BFS and DFS of a given graph.
11. Write programs to Implementing Linear and Binary Search methods.
12. Write programs to implement the following Sorting methods
 - a) Bubble Sort
 - b) Selection Sort
 - c) Insertion Sort
 - d) Merge Sort
 - e) Quick Sort
 - f) Heap Sort
13. Write a program to insert and search an element from an AVL-tree.
14. Write a program to insert and delete an element from B-tree
15. Write a program to implement all the functions of a dictionary (ADT) using hashing.
16. Write a program for generating Minimum cost spanning tree using Kruskal's algorithm

Teaching Methodologies

1. Program Demonstrations using Power Point Presentations
2. Whiteboard and Marker Pen

Text Books

1. Data Structures and Algorithms in Java, 2nd edition, A.Drozdek, Thomson.
2. Data Structures Using Java, Yeddiyah Langsam, Moshe Augenstein, Aaron M.Tenenbaum, Pearson Education.
3. Data Structures with Java, J.R.Hubbard and A.Huray, Pearson Education/PHI.
4. Data Structures, Algorithms and Applications in Java, 2nd Edition, S.Sahani



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

JAVA PROGRAMMING LAB

Course Code: GR14F5015
I Year II Semester

L:0 T:0 P:2 C:2

Prerequisites

- Basic OOP concept.
- Basic knowledge of C/C++ language.

Course Objectives

- To learn to write, compiling & execute basic java program.
- To learn the use of data types & variables, decision control structures: if, nested if etc.
- To learn the use loop control structures: do, while, for etc.
- To create classes and objects and use them in their program.
- To will learn the use oop concept i.e data abstraction & data hiding, encapsulation, inheritance, polymorphism.
- To create and use threads, handle exceptions and write applets, awt, swing.
- To learn the use of interfaces and inner classes, wrapper classes, generics.

Course Outcomes

- Understanding of OOP concepts and basics of java programming (Console and GUI based)
- The skills to apply OOP and Java programming in problem solving
- Should have the ability to extend his knowledge of Java programming further on his/her own.
- Implementation of logic using java lab
- Able to write reusable function/utilities.
- The student will be able to write a Java program that demonstrates a class hierarchy, inheritance, and a simple degree of polymorphism.
- The student will be able to create files and read from computer files using Java.
- The student will be able to create exception handling in Java programs
- The student will be able to write a program that demonstrates the use of static variables and interfaces in object-oriented programming



List of Sample Problems/Experiments

1.
 - a) Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a , b , c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.
 - b) The Fibonacci sequence is defined by the following rule: The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the n th value in the Fibonacci sequence.
2.
 - a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
 - b) Write a Java program to multiply two given matrices.
 - c) Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of `java.util`).
3. Write a Java program to find both the largest and smallest number in a list of integers.
4. Write a Java program to illustrate method overloading.
5. Write a Java program that implements the Sieve of Eratosthenes to find prime numbers.
6. Write a Java program to sort a list of names in ascending order.
7. Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:
 - a) Reading a matrix.
 - b) Printing a matrix.
 - c) Addition of matrices.
 - d) Subtraction of matrices.
 - e) Multiplication of matrices.
8. Write a Java Program to solve Tower's of Hanoi problem.
9. Write a Java Program that uses a recursive function to compute nCr . (Note: n and r values are given.)
10. Write a Java program to perform the following operations:
 - a) Concatenation of two strings
 - b) Comparison of two strings.
11. Implement the complex number ADT in Java using a class. The complex ADT is used to represent complex numbers of the form $c=a+ib$, where a and b are real numbers. The operations supported by this ADT are:
 - a) Reading a complex number
 - b) Writing a complex number.
 - c) Addition of Complex numbers
 - d) Subtraction of Complex numbers
 - e) Multiplication of complex numbers.
 - f) Division of complex numbers.
12. Write a Java program that makes frequency count of letters in a given text.



13. Write a Java program that uses functions to perform the following operations:
 - a) Inserting a sub-string in to the given main string from a given position.
 - b) Deleting n characters from a given position in a given string.
14. a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
b) Write a Java program to make frequency count of words in a given text.
15. a) Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
c) Write a Java program that displays the number of characters, lines and words in a text file.
d) Write a Java program to change a specific character in a file.
g) Note: Filename, number of the byte in the file to be changed and the new character are specified on the command line.
16. a) Develop an applet in Java that displays a simple message.
b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
17. Write a Java program that for simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
18. Write a Java program for handling mouse events.
19. a) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.
b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
20. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.
21. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is



the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)

22. a) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.
b) Write a Java program that allows the user to draw lines, rectangles and ovals.
23. a) Write a Java program to create an abstract class named Shape that contains an empty method named numberOfSides(). Provide three classes named Trapezoid, Triangle and Hexagon, all of which extend the class Shape. Each one of the classes contains the only method numberOfSides() which display the number of sides in the given geometrical figures.
b) A table named Table.txt stored in a text file has the first line as header and the remaining lines correspond to rows in table. The elements are separated by comma. Write a java program to display the table using Jtable component.
24. Write a Java program that illustrates the following
 - a. Creation of simple package.
 - b. Accessing a package.
 - c. Implementing interfaces.
25. Write Java programs that illustrates the following
 - a) Handling predefined exceptions
 - b) Handling user defined exceptions
26. Write a Java program for handling Key events.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

SOFT SKILLS PRACTICES

Course Code: GR14F5016
I Year II Semester

L:0 T:0 P:2 C:2

Prerequisites

- Should have gone through a course on English language communication.
- Should be eager to improve one's personality for facing panels

Course Objectives

- Enhance personality accomplished in behaviour and attitude.
- Expertise in cooperation and leadership when working with a team.
- Enhance skills in appearing for any type of interview.
- Realize the EGO states in interactions and know to approach to think laterally.
- Learn to face public with assertive approach and use aids to effectively enhance in such presentations.

Course Outcomes

- An accomplished professional with confidence and dexterity in dealings
- Able to speak before public with ease.

Unit-I

Positive attitude: Meaning- Attitude and behaviour- formation and change of attitudes - pragmatic steps to cultivate- examples of positive attitude - study of samples of change of attitude, negativity propagation and conception change.

Art of Public speaking: Importance of public speaking- Effective methods of delivery- Evaluating the audience- Addressing questions.

Unit-II

Etiquette and Manners: Modern etiquette- Purpose and Benefits of etiquette- classification of etiquette- Know to respect different cadres- Professional manners-Familiarity and Disparity.

Forming Values: What is a value?- A core of values- Relation between values and principles- values relating to self and others- Formation of values- Types of values-Power of values

Unit-III

Time Management: Setting goals and prioritization of time; Regulation and Backtracking; Major time management challenges.



Resume Writing: The purpose of resume-Types of resume- Do's & Don'ts in Resume writing-Cover letter.

Unit-IV

Personal Growth and Group Development: Awareness of self, Emotional Intelligence, Body language, Mannerism, Telephone Etiquette, Grooming, Role playing.

Team Building Conflict Resolution: Stages of team development, Teamwork for innovation and change, leadership Acclimatization of Organizational climate, Adapting with new Organizational change, Continuous learning.

Unit-V

Interview Skills: HR and Managerial Interview, Stress Interview, Interview etiquettes, Sample real-time questions.

Development of Critical and Reflective Thinking: Understanding thinking process using NLP, Out-of-box thinking, Ego states and transactions.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Soft Skills (know yourself & know the world), Dr. K. Alex, S. Chand.
2. Effective Business Communication, H. A. Murphy, H.W. Hilde brandt & Jane P.Thomas, 7th Edition, McGrawHill.

Reference Books

1. Secrets of Power Presentations, Bender, MacMillan.
2. Cool Time and the Two-Pound Bucket, Prentice, Macmillan
3. Cool Time and the Two-Pound Bucket, Prentice, Macmillan
4. Basic Business Communication – Lesikar / Flatley
5. Wallance, Masters, Personality Development, Cengage Learning, 2009
6. Mohan, Developing Communication Skills, Macmillan, 2010
7. Hurlock, Personality Development, Tata McGraw Hill, 2010
8. Bhatti, The Dynamics of Personality, Pearson Education , 2010





II-Year





GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

ADVANCED UNIX PROGRAMMING

Course Code: GR14F5017
II Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Learner must have completed a course on Computer Organization and Operating Systems.
- Should be well familiar with the concepts and terminology of Operating Systems.
- Learner should have a good understanding of a programming language preferably “C” language.

Course Objectives

- To familiarize students with Linux Environment
- To learn fundamentals of Shell programming
- To develop a deeper understanding of operating system functions.
- To familiarize students with implementation details of Operating Systems
- To simulate various basic concepts of Operating System Functions using system calls and library functions.

Course Outcomes

- Should be able to work confidently in Linux/Unix Environment.
- Should be able to write shell scripts to automate various tasks.
- Should be able to use system calls for interacting with File System.
- Should be able to use System calls to Create Process, Handle Signals.
- Should be able to perform Inter Process Communication and Network Programming.

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.

Unit-II

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



Unit-III

Linux Files: File Concept, File System Structure, Inodes, File types, The standard I/O (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputc, putc, putchar, fgets, gets etc.), formatted I/O, stream errors, kernel support for files, System calls, library functions, file descriptors, low level file access - usage of open, creat, read, write, close, lseek, stat family, umask, dup, dup2, fcntl, file and record locking. File and directory management - chmod, chown, links (soft links & hard links - unlink, link, symlink), mkdir, rmdir, chdir, getcwd, Scanning Directories - opendir, readdir, closedir, rewinddir, seekdir, telldir functions.

Unit-IV

Linux Process: Process concept, Kernel support for process, process attributes, process hierarchy, process states, process composition, process control - process creation, waiting for a process, process termination, zombie process, orphan process, system call interface for process management - fork, vfork, exit, wait, waitpid, exec family, system. Linux Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise, alarm, pause, abort, sleep functions.

Unit-V

Interprocess Process Communication: Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes, FIFOs, Introduction to three types of IPC (Linux) - message queues, semaphores and shared memory. Message Queues - Kernel support for messages, Linux APIs for messages, client/server example. Semaphores - Kernel support for semaphores, Linux APIs for semaphores, file locking with semaphores. Shared Memory - Kernel support for shared memory, Linux APIs for shared memory, semaphore and shared memory example.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

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.

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA COMMUNICATION AND COMPUTER NETWORKS

Course Code: GR14F5018
II Year I Semester

L:3 T:1 P:0 C:4

Prerequisites

- Should have learnt concepts of operating systems.
- Have some real-time exposure to usage of Computer Network based application or should have been introduced to laboratory with Network connections

Course Objectives

- To introduce the fundamental various types of computer networks and their applications.
- To understand the principles involved in different layers with details of their involvement in packet transfer.
- To understand the relationship between the underlying transmission network and the specifications in the packet.
- To introduce UDP and TCP Models with the constraints in its implementation.

Course Outcomes

- After this course students should get a clear idea of the details of the various Protocols and the frames involved.
- After this course he/she will be able to appreciate the different aspects involved in transfer of data in a computer network by which if he/she uses tools, he/she can easily assimilate internal principles involved.

Unit-I

Introduction to Networks: internet, protocols and standards, the OSI model, layers in OSI model, TCP/IP suite, Addressing, Introduction to Analog and digital signals. Physical Layer Introduction to digital transmission, multiplexing, transmission media, circuit switched networks, Datagram networks, virtual circuit networks, switching.

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

Computer Network Topology: Connecting LANs, backbone networks and virtual LANs, Wireless WANs, SONET, frame relay and ATM. Network Layer: Logical addressing, internetworking, tunneling, address mapping, ICMP, IGMP, forwarding, uni-cast routing protocols, multicast routing protocols.

Unit-IV

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

Unit-V

Application Layer: Domain name space, DNS in internet, electronic mail, FTP, WWW, HTTP, SNMP, TELNET, SSH, C

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

1. An Engineering Approach to Computer Networks-S.Keshav,2nd Edition,Pearson Education.
2. Understanding communications and Networks,3rd Edition, W.A.Shay,Cengage Learning.
3. Computer and Communication Networks Nader F. Mir, Person Education
4. Computer Networking: A Top-Down Approach Featuring the Internet,James F.Kurose, K.W.Ross, 3rd Edition,Pearson Education.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

MANAGEMENT INFORMATION SYSTEMS

Course Code: GR14F5019
II Year I Semester

L:3 T:1 P:0 C:4

Pre-requisites:

- Learner must have knowledge about organizational terminology .
- Should be well familiar with the functions of the organization.

Course Objectives

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

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

Course Outcomes:

- Should be able to work with Datawarehouses, Datamart and datamining
- Should be able to Understand Organizations and Business Firms
- Should be able to understand difference between the various Organizations

Unit-I

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



Unit-II

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

Unit-III

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

Unit-IV

Implementation, evaluation and maintenance of the MIS: Plan the implementation, acquire floor space and plan space layouts, organize for implementation, develop procedures for implementation, train the operating personal, computer related acquisition, develop forms for data collection and information dissemination, develop the files, test the system, cut over, document the system, evaluate the MIS, control and maintain the system.

Unit-V

Pitfalls in MIS development: Fundamental weaknesses, soft spots, in planning, design problems, implementation - the TAR PIT. Introduction to E-Commerce and E-commerce challenges, A overview of ERP, Applications of information systems to business. Security and ethical issues of information systems.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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



Reference Books

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



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

SOFTWARE ENGINEERING

Course Code: GR14F5020
II Year I Semester

L:3 T:0 P:0 C:4

Prerequisites

- Student should have the knowledge of basic engineering principles which specifies the systematic manufacturing of a product/service.
- He/she should also know the Re-Engineering Process.

Course Objectives

- To Know what is a Software and Software Engineering and how it is playing a vital role in various applications. Software Engineering is the core for today's Software implementation. Also to know the fields which are using Software Engineering applications.

Course Outcomes

- After reading Software Engineering, a student is able to know the difference between Manual and Software oriented applications as per the working conditions. Student is also able to understand the implementation of the Software Engineering Process.

Unit-I

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

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

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

Unit-II

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

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

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



Unit-III

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

Creating an architectural Design: Software Architecture, Data design, Architectural styles and patterns, Architectural Design, assessing alternative architectural designs.

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

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

Unit-IV

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

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

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

Unit-V

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

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards, Introduction to Web-based Software Engineering Development.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Software Engineering :A practitioner's Approach, Roger S Pressman, sixth edition, McGrawHill International Edition, 2005
2. Software Engineering, Ian Sommerville, seventh edition, Pearson education, 2004.



Reference Books

1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
2. Software Engineering : A Primer, Waman S Jawadkar, Tata McGraw-Hill, 2008
3. Fundamentals of Software Engineering, Rajib Mall, PHI, 2005
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition, 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA BASE MANAGEMENT SYSTEMS

Course Code: GR14F5021
II Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Experience with programming in some common programming language.
- A good understanding of data structures and algorithms.

Course Objectives

The objective of the course is to learn the foundation of Database and Database Management Systems. In this course, student will learn to design a database, removing the redundancies using Functional Dependencies and Normalization. Student also learn Querying, Improve the database performance by Indexing, Hashing and finally optimize the queries, Transaction Management and Memory management.

Course Outcomes

Demonstrate the basic elements of a data models such as ER and relational data models. Ability to identify the data models for relevant problems. Ability to design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respective data. Apply normalization and professional attitudes for the development of application software's. Ability to understand Transaction management and memory Management with relevant problems.

Unit-I

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



Unit-II

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

Unit-III

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

Unit-IV

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

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

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

Unit-V

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

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

Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM) B+ Trees: A Dynamic Index Structure, Search, Insert, Delete. Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendable vs. Linear Hashing.



Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

ADVANCED UNIX PROGRAMMING LAB

Course Code: GR14F5022
II Year I Semester

L:0 T:0 P:2 C:2

Prerequisites

- Learner must have completed a course on Computer Organization and Operating Systems.
- Should be well familiar with the concepts and terminology of Operating Systems.
- Learner should have a good understanding of a programming language preferably “C” language.

Course Objectives

- To familiarize students with Linux Environment
- To learn fundamentals of Shell programming
- To develop a deeper understanding of operating system functions.
- To familiarize students with implementation details of Operating Systems

Course Outcomes

- Should be able to work confidently in Linux/Unix Environment.
- Should be able to write shell scripts to automate various tasks.
- Should be able to use system calls for interacting with File System.
- Should be able to use System calls to Create Process, Handle Signals.

List of Sample Problems/Experiments

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 to find factorial of a given integer.
6. Write a c program that makes a copy of a file using standard I/O and system calls.



7. Implement in C the following Unix commands using System calls
 - A. Cat
 - B. ls
 - C. Mv
8. Write a program that takes one or more file/directory names as command line input and reports the following information on the file.
9. File type.
10. Number of links.
11. Time of Last Access
12. Read Write and execute permissions
13. Write a C program to list for every file in a directory, its inode number and file name.
14. Write a C program to emulate the Unix `ls -l` command.
15. Write a C program that demonstrates redirection of standard output to a file. Ex: `ls > f1`.
16. Write a C program that illustrates the uses of the `mkdir`, `opendir`, `readdir`, `closedir` and `rmdir` API's.
17. 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.
18. Write a C program to create a Zombie process.
19. Write a C program that illustrates how an orphan is created.
20. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex: `- ls -l | sort`
21. Write C programs that illustrate communication between two unrelated processes using named pipe.
 - A. 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.
 - B. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (A)) and displays them.
- 22) Simulate the following CPU Scheduling algorithms

Round Robin	SJF	FCFS	Priority
-------------	-----	------	----------

 Simulate the following page replacement algorithms

FIFO	LRU	LFU
------	-----	-----

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations
3. Demonstration using LCD

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, Eillie Quigley, Pearson Education



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA BASE MANAGEMENT SYSTEMS LAB

Course Code: GR14F5023
II Year I Semester

L:0 T:0 P:2 C:2

Prerequisites

- Experience with programming in some common programming language.
- A good understanding of data structures and algorithms.
- Fundamentals of File Systems and Storage Structures

Course Objectives

- This lab enables the students to practice the concepts learnt in the subject Database management systems.
- To give a good formal foundation on the relational model of data
- To introduce systematic database design approaches covering conceptual design, logical design and an overview of physical design
- To present the concepts and techniques relating to query processing by SQL engines

Course Outcomes

- Ability to design and implement a database schema for a given problem-domain.
- Ability to populate and query a database using SQL and PL/SQL commands.
- Ability to declare and enforce integrity constraints on a database.
- Be capable to Design and build a GUI application.
- Apply the normalization techniques for development of application software to realistic problems.
- Ability to formulate queries using SQL DML/DDD/DCL commands.

Recommended Systems / Software Requirements

- Intel based desktop PC
 - Mysql / Oracle latest version Recommended

Lab-Problem-Part 1

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:

- Reservation
- Ticketing
- Cancellations



Reservations

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

Cancellation and Modifications

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

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

Week1: E-R Model

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

1. BUS
2. Ticket
3. Passenger

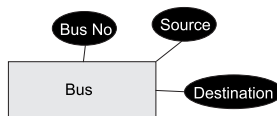
Primary Key Attributes

Ticket ID (Ticket Entity) Passport ID (Passenger Entity) Apart from the above mentioned entities you can identify more. The above mentioned are few.

Week2: Concept design with E-R Model

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

Example: E-R diagram for bus



Week3: Relational Model

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

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



Passenger

Name	Age	Sex	Address	Passport ID
------	-----	-----	---------	-------------

Week4: Normalization

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

Week5: Installation of Mysql and practicing DDL commands

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

Example for creation of a table. CREATE TABLE Passenger (Passport id INTEGER PRIMARY KEY,

Name CHAR (50) NULL, Age Integer, Sex Char);

Week6: Practicing DML commands

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

SELECT - retrieve data from the a database INSERT - insert data into a table

UPDATE - updates existing data within a table DELETE - deletes all records from a table, the space for the records remain Inserting values into Bus table:

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

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

Inserting values into Bus table:

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

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

Few more Examples of DML commands:

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

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

Week7: Querying

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



Practice the following Queries:

Display unique PNR_no of all passengers. Display all the names of male passengers. Display the ticket numbers and names of all the passengers. Display the source and destination having journey time more than 10 hours. Find the ticket numbers of the passengers whose name start with 'A' and ends with 'H'. Find the names of passengers whose age is between 30 and 45. Display all the passengers names beginning with 'A' Display the sorted list of passengers names Display the Bus numbers that travel on Sunday and Wednesday Display the details of passengers who are traveling either in AC or NON_AC (Using only IN operator)

Week 8 and week9: Querying (continued...)

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

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

Week10: Triggers

In this week you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database. Eg: CREATE TRIGGER updcheck BEFORE UPDATE ON passenger

```
FOR EACH ROW BEGIN
  IF NEW.TicketNO > 60 THEN SET New.Ticket no = Ticket no;
  ELSE
  SET New.Ticketno = 0; END IF;
END;
```

Week11: Procedures

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

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



Week12: Cursors

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

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

Tables

BUS

Bus No: Varchar: PK

Source : Varchar

Destination : Varchar

Age : int (4)

Sex:Char(10) : Male / Female

PPNO: Varchar(15)

PASSENGER

PNR_No : Numeric(9) : PK

Ticket_No: Numeric (9)

Name: Varchar(15)

RESERVATION

PNR_No: Numeric(9) : FK Journey_date : datetime(8) No_of_seats : int (8)

Address : Varchar (50)

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

Status: Char (2) : Yes / No

CANCELLATION

PNR_No: Numeric(9) : FK Journey_date : datetime(8) No_of_seats : int (8)

Address : Varchar (50)

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

Status: Char (2) : Yes / No

Ticket

Ticket_No: Numeric (9): PK

Journey_date : datetime(8)

Age : int (4)

Sex:Char(10) : Male / Female

Source : Varchar

Destination : Varchar

Dep_time : Varchar



Lab-Problem-Part 2

SQL

Task 1: DDL commands (Create, Alter, Drop and Truncate)

1) Create a table EMP with the following structure.

Name	Type
EMPNO	NUMBER (6)
ENAME	VARCHAR2 (20)
JOB	VARCHAR2 (10)
MGR	NUMBER (4)
DEPTNO	NUMBER (3)
SAL	NUMBER (7,2)

2) Add a column commission to the EMP table. Commission should be numeric with null values allowed.

3) Modify the column width of the job field of EMP table.

4) Create dept table with the following structure.

Name	Type
DEPTNO	NUMBER(2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

Deptno as the primarykey

5. Add constraints to the EMP table that is empno as the primary key and deptno as the foreign key.
6. Add constraints to the EMP table to check the empno value while entering (i.e) empno > 100.
7. Salary value by default is 5000, otherwise it should accept the values from the user.
8. Add columns DOB to the EMP table.
9. Add and drop a column DOJ to the emp table.
10. Insert few rows and truncate those from the EMP1 table and also drop it.

Task 2: DML COMMANDS (Insert, Select, Update, Delete)

1. Insert 5 records into dept table.
2. Insert 11 records into EMP table.
3. Update the EMP table to set the default commission of all employees to Rs1000/- who are working as managers.
4. Create a table employee with the same structure as the table EMP and insert rows into the table using select clauses.
5. Delete only those who are working as supervisors.



6. Delete the rows whose empno is 7599.
7. List the records in the EMP table order by salary in ascending order.
8. List the records in the EMP table order by salary in descending order.
9. Display only those employees whose deptno is 30.
10. Display deptno from the table employee avoiding the duplicated values.
11. List the records in sorted order of their employees.
12. Create a manager table from the EMP table which should hold details only about the managers.
13. List the employee names and the department name in which they are working.

Task 3: SQL Operators

1. List all employee names, salary and 15% rise in salary.
2. Display the rows whose empno ranges from 7500 to 7600.
3. Display the rows whose empno not in range from 7500 to 7600.
4. Display all the employees in dept 10 and 20 in alphabetical order of names.
5. List the employee names whose commission is null.
6. Display all the details of the records whose employee name starts with 'S'.
7. Display all the details of the records whose employee name does not start with 'M'.
8. Display the names of employees whose second character is 'i'.
9. Display all the details of the records whose employee name ends with 'A'.
10. List all employees which starts with either B or C.
11. List out the employee names whose salary is greater than 5000, 6000.

Task 4: SQL Aggregate Functions, Group By clause, Having clause

1. Count the total records in the EMP table.
2. Calculate the total and average salary of the employee.
3. Determine the max and min salary and rename the column as max_salary and min_salary.
4. Display total salary spent for employees.
5. Find no. of depts in employee table.
6. Display total salary spent for each job category.
7. Display lowest paid employee details under each manager.
8. Display highest paid employee details under each category.
9. Display job wise sum, average, max, min salaries .
10. Display maximum salaries of all the departments having maximum salary > 2000
11. Display average salaries for all departments having more than five employees.
12. Display job wise sum , avg , max , min salaries in department 10 having avg salary > 1000 and the result is ordered by sum of salary in descending order.



Task 5: SQL functions

Practice on Number functions, character functions, date functions, conversion functions and miscellaneous functions.

Task 6: Nested Queries

1. Find the third highest salary of an employee.
2. Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'M'.
3. Write a query to find all the employees who work in the same job as Jones.
4. Write a query to display information about employees who earn more than any employee in dept 30.
5. Display the employees who have the same job as Jones and whose salary \geq Fords
6. Write a query to list the employees in dept 20 with the same job as anyone in dept 30.
7. List out the employee names who get the salary greater than the maximum salaries of dept with dept no 20,30.
8. Display the maximum salaries of the departments whose maximum salary is greater than 9000.
9. Display the maximum salaries of the departments whose minimum salary is greater than 1000 and lesser than 5000.

Task 7: Joins, Set Operators.

1. Display all the employees and the departments implementing a left outer join.
2. Display the employee name and department name in which they are working implementing a right outer join.
3. Display the employee name and department name in which they are working implementing a full outer join.
4. Write a query to display their employee names and their managers salary for every employee.
5. Write a query to output the name , job, empno, deptname and location for each dept, even if there are no employees.
6. Find the name of the manager for each employee.
7. Display the details of those who draw the same salary.
8. Display all the dept numbers available with the dept and accdept tables avoiding duplicates.
9. Display all the dept numbers available with the dept and accdept tables.
10. Display dept no available in both the dept and accdept tables

Task 8: Views

1. Display only the details of the employees those who are managers.
2. Display only the details like empno, empname, deptno, deptname of the employees
3. Display only the details like empno, empname, deptno, deptname of the all the employees except the HOD and CEO .



4. Display all the views generated.
5. Execute the DML commands on the view created.
6. Drop a view.

PL/SQL

Task 9 & 10: Basic & Advanced

1. Write a PL/SQL program to find greatest of three numbers.
2. Write a PL/SQL program to find Factorial of the given number
3. Write a PL/SQL program to print Fibonacci series
4. Write a PL/SQL program to print numbers from 1 to 100
5. Write a PL/SQL program to print odd numbers from 1 to 100
6. Write a PL/SQL program to find area of rectangle
7. Write a PL/SQL program to find area of a circle
8. Write a PL/SQL program to find annual salary of employee
9. Write a PL/SQL program to Bank account
10. Write a PL/SQL program to find the reverse of the given number
11. Write a PL/SQL program to check whether given number is leap year or not
12. Write a PL/SQL program to swap two numbers
13. Write a PL/SQL program to print multiplication table
14. Write a PL/SQL program to check whether the given no is even or odd using go to statement
15. Write a PL/SQL program to demonstrate cursor
16. Write a PL/SQL program to find factorial of a given number using functions
17. Write a PL/SQL program to find employee details from emp table
18. Write a PL/SQL program for Triggers using STU table
19. Write a PL/SQL program to demonstrate assertion
20. Write a PL/SQL program to find exception using emp table.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. The complete reference, 3rd edition by James R. Groff, Paul N. Weinberg, Andrew J. Opper
2. Introduction to SQL, Rick F. Vander Lans, Pearson education.
3. Oracle PL/SQL, B. Rosenzweig and E. Silvestrova, Pearson education.
4. Oracle PL/SQL Programming, Steven Feuerstein, SPD.
5. SQL & PL/SQL for Oracle 10g, Black Book, Dr. P.S. Deshpande, Dream Tech.



Reference Books

1. Oracle Database 11g PL/SQL Programming, M. Mc Laughlin, TMH.
2. SQL Fundamentals, J.J. Patrick, Pearson Education



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

WEB TECHNOLOGIES

Course Code: GR14E5033
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Learner must have completed a course on J2SE.
- Should have well familiar with the concepts and terminology of Object Oriented Programing.
- Learner should have a good understanding of a SQL.

Course Objectives

- To provide the students a detailed knowledge on understanding the role of web services in the world of web applications, various different techniques used for building an web application.
- To provide the basic knowledge of HTML, Beans, XML for communication of information from source to destination.
- To provide the students with a solid foundation in various technologies containing technical concepts required to solve web application problems and also assists the students in pursuing higher studies.

Course Outcomes

- An ability to develop project management skills related to web development.
- Master technical skills required for Web Developers through use of W3C standards, HTML, XHTML, and Style Sheets, develop databases and supporting code in Java language, Client and Server-Side Scripting languages such as JavaScript and JSP, performing unit and integration testing.
- Ability to communicate effectively to a wide variety of audiences, verbally, in writing, and electronically

Unit-I

HTML Common tags: List, Tables, images, forms, Frames; Cascading Style sheets; Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with JavaScriptXML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Unit-II

Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDKIntrospection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's. Database Access : Database Programming using JDBC, Studying Javax.sql.*package



Unit-III

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

Unit-IV

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

Unit-V

Introduction to struts framework, Model Layer, View Layer, Controller Layer, Validator, Programming using struts framework.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Web Programming, building internet applications, Chris Bates 2nd edition, WILEY Dreamtech
2. The complete Reference Java Seventh Edition by Herbert Schildt. TMH.
3. Java Server Pages – Hans Bergsten, SPD O'Reilly
4. Struts: The Complete Reference by James Holmes. McGraw-Hill/Osborne.

Reference Books

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



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

INFORMATION SECURITY

Course Code: GR14F5025
II Year II Semester

L:3 T:1 P:0 C:4

Prerequisites

- Learner must have knowledge about computer networks.
- Should be well familiar with the concepts and terminology of computer networks.

Course Objectives

- The objectives of information security
- The importance and application of each of confidentiality, integrity, authentication and availability
- To develop a various cryptographic algorithms.
- Understand public-key cryptosystem and the enhancements made to IPv4 by IPsec.
- To how to generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
- To familiarize with Intrusions and intrusion detection, Web security and Firewalls.

Course Outcomes

- Should be able to work with various cryptographic algorithms.
- Should be able to Understand Intrusions and intrusion detection, Web security and Firewalls.
- Should be able to understand difference between the various Message Authentication algorithms.

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. Kerberos, X.509 Authentication Service

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 (SSL) and Transport Layer Security (TLS), secure electronic transaction (SET).

Intruders, Virus and Firewalls: Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls.

Teaching Methodologies

1. White board marker
2. Power point presentations

Text Books

1. Network Security Essentials (Applications and Standards) by William Stallings Pearson Education, 2008.
2. Cryptography & Network Security by Behrouz A. Forouzan, TMH 2007.
3. Cryptography and Network Security by William Stallings, Fourth Edition, Pearson Education 2007.
4. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 2nd Edition

Reference Books

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



**GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

DATA WAREHOUSING AND DATA MINING

Course Code: GR14F5026
II Year II Semester

L:3 T:1 P:0 C:4

Prerequisites

- Fundamentals of SQL and Database.
- Experience with programming in some common programming language.
- A good understanding of data structures and algorithms.

Course Objectives

- To understand data mining principles and techniques: Introduce DM as a cutting edge business intelligence method and acquaint the students with the DM techniques for building competitive advantage through proactive analysis, predictive modeling, and identifying new trends and behaviors.
- To learn how to gather and analyze large sets of data to gain useful business understanding.
- To learn how to produce a quantitative analysis report/memo with the necessary information to make decisions. And Learning how to gather and analyze large sets of data to gain useful business understanding.
- To describing and demonstrating basic data mining algorithms, methods, and tools
- To identifying business applications of data mining
- To develop and apply critical thinking, problem-solving, and decision-making skills.
- Overview of the developing areas - web mining, text mining, and ethical aspects of data mining.

Course Outcomes

- Understand the principles behind DWDM technologies and the characteristics of DWDM components
- Understand why the data warehouse in addition to traditional database systems.
- Ability to perform the preprocessing of data and apply mining techniques on it.
- Ability to identify the association rules, classification and clusters in large data sets.
- Ability to solve real world problems in business and scientific information using data mining.



Unit-I

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

Unit-II

Data Warehouse and OLAP Technology for Data Mining: Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Usage of Data Warehousing Online Analytical Processing and Mining Data Cube Computation: Efficient Methods for simple Data Cube Computation (Full Cube, Iceberg Cube, Closed Cube and Shell Cube), Discovery Driven exploration of data cubes, Attribute-Oriented Induction for data characterization and its implementation

Unit-III

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

Unit-IV

Classification and Prediction: Description and comparison of classification and prediction, preparing data for Classification and Prediction Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Back propagation Prediction, linear and non-linear regression, evaluating accuracy of a Classifier or a Predictor

Unit-V

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

Applications and Trends in Data Mining: Data Mining Applications, Data Mining Systems Products and Research Prototypes, Additional Themes on Data Mining and Social Impacts of Data Mining, Spatial Data Mining, Text Mining, Mining the World Wide Web, Graph Mining, Multimedia Data Mining, Social Network Analysis and Multi relational Data Mining



Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Data Mining –Concepts and Techniques -Jiawei Han, Micheline Kamber, and Jian Pei, Morgan Kaufmann Publishers, 3rdEdition, 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.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-I
SCRIPTING LANGUAGES

Course Code : GR14F5027
 II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Experience with programming in some common programming language
- Knowledge of Basics of computer networks and databases

Course Objectives

- The course demonstrates an in depth understanding of the tools and the scripting languages necessary for design and development of applications.
- To provide a foundation to use scripts for use on the internet
- To provide the basic knowledge to use python with object orientation and database programming

Course Outcomes

- Ability to develop scripts in PERL and use PHP for web pages.
- Develop skills to use python with objects and to access databases

Unit-I

Introduction to PERL and Scripting: Scripts and Programs, Origin of Scripting , Scripting Today, Characteristics of Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines, advance perl - finer points of looping, pack and unpack, file system, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

Unit-II

PHP Basics: Features, Embedding PHP Code in your Web pages, Outputting the data to the browser, Datatypes, Variables, Constants, expressions, string interpolation, control structures Function, Creating a Function, Function Libraries, Arrays, strings and Regular Expressions.

Unit-III

Advanced PHP Programming: PHP and Web Forms, Files, PHP Authentication and Methodologies -Hard Coded, File Based, Database Based, IP Based, Login Administration, Uploading Files with PHP,



Sending Email using PHP, PHP Encryption Functions, the Mcrypt package, Building Web sites for the World – Translating Websites- Updating Web sites Scripts, Creating the Localization Repository, Translating Files, text, Generate Binary Files, Set the desired language within your scripts, Localizing Dates, Numbers and Times.

Unit-IV

Python: Introduction to Python language-Python-syntax, statements, functions, Built-in-functions and Methods, Modules in Python, Exception Handling. Object Orientation- Data in Python, Data Structures in Python, Defining Classes, The class Statement.

Unit-V

Advanced Python Programming: Database Programming in Python-Client/Server Database Architectures, The Python Database Interface. Integrated Web Applications in Python – Building Small, Efficient Python Web Systems, Web Application Framework.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-I
E-COMMERCE

Course Code: GR14F5028
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Student should have knowledge of manual business transactions and should know the difference between local market and global market.
- He/she has to think what is a transaction and how it is implemented automatically.

Course Objective

- To Know the Electronic Commerce basics and how the Consumer applications are emerging as Online. E-Commerce is the core for today's Online Transactional Processing. Also to know the fields which are using E-Commerce applications.

Course Outcomes

- After reading E-Commerce , a student is able to know the difference between Manual Transactions and Online Transactions and also the power of the Electronic Commercial applications. Student is also able to understand the implementation of the Intra-Organizational and Inter-Organizational commercial applications.

Unit-I

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

Unit-II

Electronic payment systems: Digital Token-Based, Smart Cards, Credit Cards, Risks in Electronic Payment systems. Inter Organizational Commerce: EDI, EDI Implementation, Value added networks.

Unit-III

Intra Organizational Commerce: work Flow, Automation Customization and internal Commerce, Supply chain Management. Corporate Digital Library: Document Library, digital Document types.



Unit-IV

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

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

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Book

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

Reference Books

1. E-Commerce fundamentals and applications Hendry Chan, Raymond Lee, Tharam Dillon, Ellizabeth Chang, John Wiley.
2. E-Commerce, S.Jaiswal – Galgotia.
3. E-Commerce, Efrain Turbon, Jae Lee, David King, H.Michael Chang.
4. Electronic Commerce – Gary P.Schneider – Cengage Learning..
5. E-Commerce – Business, Technology, Society, Kenneth C.Taudon, Carol Guyerico Traver.
6. Electronic Commerce, B.Bhaskar, 3rd edition, TMH.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-I
GRID AND CLOUD COMPUTING

Course Code: GR14F5029
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

Students should have completed a course on Computer Organization and Computer Networks.

Course Objectives

- To Familiarise Basics, techniques and tools for Grid & Cloud Computing
- To understand any kind of heterogeneous resources over a network using open standards
- To get insight of the Service models
- To introduce the broad perceptive of Cloud architecture and model
- To understand the concept of Virtualization and design of Cloud Services
- To understand the features of Cloud Simulator
- To learn to design the trusted Cloud Computing system

Course Outcomes

- Student will explain the lead players in Cloud.
- Student differentiates Cluster, Grid and Cloud computing applications.

Unit-I

System models for advanced computing –clusters of cooperative computing, grid computing and Cloud computing; software systems for advanced computing-service oriented software and parallel and distributed programming models with introductory details, Features of grid and Cloud platform.

Unit-II

Cloud Computing services models and features in Saas, Paas and Iaas. Service oriented architecture and web services; Features of Cloud computing architectures and simple case studies.

Unit-III

Virtualization- Characteristic features, Taxonomy Hypervisor, Virtualization and Cloud Computing, Pros and Cons of Cloud Computing, Technology Examples/Case Studies.



Unit-IV

Cloud programming Environmental- Map Reduce Hadoop Library from Apache, Open Source Cloud Software Systems –Eucalyptus.

Unit-V

Grid Architecture and Service modeling, Grid resource management, Grid Application trends.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Distributed and Cloud Computing, Kaittwang Geoffrey C.Fox and Jack J Dongrra, Elsevier India 2012.
2. Mastering Cloud Computing- Raj Kumar Buyya, Christian Vecchiola and S.Tanurai Selvi, TMH, 2012.

Reference Books

1. Cloud Computing, John W. Ritting House and James F Ramsome, CRC Press, 2012.
2. Enterprise Cloud Computing, Gautam Shroff, Cambridge University Press, 2012.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-I
COMPUTER GRAPHICS

Course Code: GR14F5030
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Knowledge of basic geometry and Mathematics
- Awareness of data structures, 3D images, curves and surfaces

Course Objectives

- To provides students a detailed understanding of primitives to transform 2D and 3D images
- To provide foundation to know ways of generation of curves and surfaces.

Course Outcomes

- Ability to implement methods of 2D and 3D transformations
- To implement methods for generating curves & surfaces with the constraints involved

Unit-I

Geometry and line Generation: Points, Lines, Planes, Pixels and frame buffers, types of display devices, DDA and Brasenham's Line Algorithms, Brasenham's algorithms for circle generation, algorithm for ellipse generation, character generation, Aliasing and Antialiasing.

Unit-II

Graphics Primitives: Display Files, Display processors, Algorithms for polygon generation, Polygon filling algorithms, NDC (Normalized device co-ordinates), pattern filling. 2D Transformations: Scaling, Rotation, translation, homogenous for ordinates, rotation about arbitrary point Reflections, Zooming.

Unit-III

Segment tables: Operations on segments, data structures for segments and display files. Windowing and clipping: Window, viewport, viewing transformation clipping, line and polygon clipping, generalized clipping, multiple windowing.

Unit-IV

3D Graphics: 3D primitives, 3D Transformations, Projections, parallel perspective, isometric, viewing transformations, Hidden surfaces and line removal, Painter's algorithm, Z-buffers, Warnocks algorithm, shading algorithms, 3D clipping.



Unit-V

Curves and surfaces: Generation of curves and surfaces using Hermite, Beziere and BSpline, sweeping method of interpolation- Raster graphics Architecture: Simple RasterDisplay system, Display processor system, standard graphics pipeline, multiprocessor Raterilization Architecture. Pipeline and parallel front end architecture.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. David F Rogers – Procedural Elements for Computer Graphics – McGraw-Hill.
2. Foley, Vandam, Feiner & Huges – Computer Graphics Principles and Practice –Addison Wesley.
3. Newman & Sproul – Principles of Interactive Computer Graphics.
4. David F Rogers and Adams – Mathematical Elements for Computer Graphics.CS3203



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-II
RICH INTERNET APPLICATIONS

Course Code: GR14F5031
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- The course is for those interested in developing dynamic Web applications.
- One should have programming knowledge.

Course Objectives

- To develop students competency in producing dynamic and creative graphic solutions for multimedia productions.
- To provide basic concepts and techniques of interactive authoring.
- To introduce scripting skills necessary for implementing interactive, rich internet applications using multimedia technologies.
- To develop Artistic visual style and layout, editing and integration of graphic images, animation, video and audio files.
- To master industry-wide software and technologies to create highly interactive, rich internet applications.

Course Outcomes

- Know multimedia applied on Internet.
- Learn to use Adobe Flash, Flex 3 and Ajax with Dojo

Unit-I

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

Unit-II

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

Unit-III

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



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

Unit-IV

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

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

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

1. Multimedia Communications: Applications, Networks, Protocols and Standards, Fred Halsall, Pearson Education, 2001, rp 2005.
2. Multimedia Making it work, Tay Vaughan, 7th edition, TMH, 2008.
3. Introduction to multimedia communications and Applications, Middleware, Networks, K.R.Rao, Zoran, Dragored, Wiley India, 2006, rp. 2009.
4. Multimedia Computing, Communications & Applications, Ralf Steinmetz and Klara Nahrstedt, Pearson Education, 2004



5. Principles of Multimedia, Ranjan Parekh, TMH, 2006.
6. Multimedia in Action, James E.Shuman, Cengage Learning, 198, rp 2008.
7. Multimedia Systems design, Prabhat K. Andleigh, Kiran Thakrar, PHI, 1986.
8. Multimedia and Communications Technology, Steve Heath, Elsevier, 1999, rp 2003.
9. Adobe Flash CS3 Professional, Adobe press, Pearson Education, 2007.
10. Flash CS3 Professional Advanced, Russel Chun, Pearson Education, 2007.
11. Flash CS5, Chris Grover, O'Reilly, SPD, 2010.
12. SAMS Teach yourself Adobe flash CS3, Pearson Education, 2007.
13. Flex 4 Cookbook, Joshua Noble, et.al, O'Reilly, SPD 2010.
14. Flex3 – A beginner's guide, Michele E.Davis, Jon A. Phillips, TMH, 2008.
15. Mastering Dojo, R.Gill, C.Riecke and A.Russell, SPD



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-II
DISTRIBUTED SYSTEMS

Course Code: GR14F5032
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Should know concepts of operating systems
- Should have knowledge of computer Networks

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.

Course Outcomes

- The student distinguishes various architectures used to design distributed systems, such as client-server and peer-to-peer.
- The student will analyze and explain various interprocess communication techniques, such as Remote Procedure Calling, Design issues, Implementation, Asynchronous RPC.
- The student will build distributed systems using various techniques for tolerating partial failures, such as leasing, replication and interprocess coordination techniques, such as distributed mutual exclusion, distributed monitors, tuple spaces.

Unit-I

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

Unit-II

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

Unit-III

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



Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control.

Unit-IV

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

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-II
SOFTWARE PROJECT MANAGEMENT

Course Code: GR14F5033
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Familiarity with Software Engineering principles
- Understanding of the importance of management in software development

Course Objectives

- Understand the specific roles in the organizations with their responsibilities
- Clear understanding of life cycle phases and its milestones
- Understanding role of project management including planning, scheduling, risk management etc
- A clear understanding of principles, techniques, methods and tools for model based management of software projects
- Understanding of basic infrastructure competencies

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

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

Unit-I

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

Unit-II

Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections. The old way and the new: The principles of conventional software engineering, principles of modern software management, transitioning to an iterative process.



Unit-III

Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases. Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts. Model based software architectures: A Management perspective and technical perspective.

Unit-IV

Work Flows of the process: Software process workflows, Inter trans workflows. Checkpoints of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments. Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Interaction planning process, Pragmatic planning. Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations. Process Automation: Automation Building Blocks, The Project Environment.

Unit-V

Project Control and Process instrumentation: The server care Metrics, Management indicators, quality indicators, life cycle expectations pragmatic Software Metrics, Metrics automation. Tailoring the Process: Process dicriminants, Example. Future Software Project Management: Modern Project Profiles Next generation Software economics, modern Process transitions. Case Study: The Command Center Processing and Display System-Replacement (CCPDS-R)

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. Software Project Management, Walker Royce, Pearson Education, 1998
2. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, Tata Mc Graw Hill, 2006

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-II
SOFT COMPUTING

Course Code: GR14F5034
II Year II Semester

L:4 T:0 P:0 C:4

Prerequisites

- Must have an aptitude for logical thinking, problem solving.
- Students should have completed a course on Algorithm Analysis and Computer Programming.

Course Objectives

- To make the students understand the concepts of soft computing
- Gives a clear idea of Artificial Intelligence and Neural Networks.
- Student gets introduced to Fuzzy Sets and Fuzzy Logic and Genetic Algorithms.

Course Outcomes

- Learn to apply Predicate Logic and Rules.
- Techniques of training Neural Networks.
- Able to use classical relations and fuzzy relations.

Unit-I

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

Unit-II

Artificial Neural Networks: Introduction, Basic models of ANN, important terminologies, Supervised Learning Networks, Perceptron Networks, Adaptive Linear Neuron, Back propagation Network. Associative Memory Networks. Training Algorithms for pattern association, BAM and Hopfield Networks.

Unit-III

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



Unit-IV

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

Unit-V

Fuzzy Arithmetic and Fuzzy Measures, Fuzzy Rule Base and Approximate Reasoning Fuzzy Decision making Fuzzy Logic Control Systems.

Genetic Algorithm- Introduction and basic operators and terminology.

Applications: Optimization of TSP, Internet Search Technique.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

WEB TECHNOLOGIES LAB

Course Code: GR14F5035
II Year II Semester

L:0 T:0 P:2 C:2

Prerequisites

- Learner must have completed a course on J2SE.
- Should have well familiar with the concepts and terminology of Object Oriented Programing.
- Learner should have a good understanding of a SQL.

Course Objectives

- Design and develop a simple interactive web application
- Enable students to write simple server-side scripts for web interactivity
- Students should be able to develop simple back-end database to support a website

Course Outcomes

- An ability to design and implement web services.
- Master the techniques needed for developing web applications using consistent databases.

Hardware and Software requirement

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

Week-1

Design the following static web pages required for an online book store web site.
1) HOME PAGE: The static home page must contain three frames.

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

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

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



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

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	<i>Description of the Web Site</i>			

Fig 1.1

2) LOGIN PAGE:

This page looks like below:

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

3) CATALOGUE PAGE: The catalogue page should contain the details of all the books available in the web site in a table.

The details should contain the following:





Snap shot of Cover Page

Author Name

Publisher

Price

Add to cart button

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	   	Book : XML bible Author : Winston Publication : Wiely Book : AI Author : S. Russel Publication : Princeton hall Book : Java2 Author : Watson Publication : BPB Publication Book : HTML in 24 hours Author : Sam Peter Publication : Sam publication	\$40.5 \$63 \$35.5 \$50	<input type="button" value="Add Cart"/> <input type="button" value="Add Cart"/> <input type="button" value="Add Cart"/> <input type="button" value="Add Cart"/>



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

4) CART PAGE:

The cart page contains the details about the books which are added to the cart. The cart page should look like this:

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

5) REGISTRATION PAGE:

Create a “registration form “with the following fields

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

Week 3:

VALIDATION:

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

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

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

Week-4:

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

- 1) Use different font, styles:

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

**For example:**

```
<HTML>
<HEAD>
  <style type="text/css">
    B.headline {color:red; font-size:22px; font_family:arial; text-
    decoration:underline}
  </style>
</HEAD>
<BODY>
  <b>This is normal bold</b><br>
  Selector{cursor:value}
```

For example

```
<html>
<head>
  <style type="text/css">
    .xlink{cursor:crosshair}
    .hlink{cursor:help}
  </style>
</head>
<body>
  <b>
    <a href="mypage.htm" class="xlink">CROSS LINK</a>
    <a href="mypage.htm" class="hlink">HELP LINK</a>
  </b>
</body>
</html>
  <b class="heading">This is headline style bold</b>
</body>
</html>
```

- 2) Set a background image for both the page and single elements on the page. You can define the background image for the page like this:
 BODY {background-image:url(myimage.gif);}
- 3) Control the repetition of the image with the background-repeat property, as background repeat: repeatTiles the image until the entire page is filled, just like an ordinary background image in plain HTML.
- 4) Define styles for links as A:link
 a:visited
 a:active
 a:hover Example:
 <style type="text/css"> a:link {text-decoration: none}



```
a:visited {text-decoration: none}
A:active {text-decoration: none}
a:hover {text-decoration: underline; color: red;} </style>
```

5) Work with layers: For example:

LAYER 1 ON TOP:

```
<div style="position:relative; font-size:50px; z-index:2;">
```

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

LAYER 2 ON TOP:

```
<div style="position:relative; font-size:50px; z-index:3;">
```

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

6) Add a customized cursor: Selector {cursor:value} For example:

```
<html>
```

```
<head>
```

```
  <style type="text/css">
```

```
  .xlink{cursor:crosshair}
```

```
  .hlink{cursor:help}
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <b>
```

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

```
  <a href="mypage.htm" class="hlink">HELP LINK</a>
```

```
  </b>
```

```
</body>
```

```
</html>
```

Week-5:

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

- 1) Title of the book
- 2) Author Name
- 3) ISBN number
- 4) Publisher name
- 5) Edition
- 6) Price Write a Document Type Definition (DTD) to validate the above XML file. Display the XML file as follows.

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



Note: Give at least for 4 books. It should be valid syntactically.
Hint: You can use some xml editors like XML-spy

Week-6:

JAVA BEANS:

Create a simple visual bean with a area filled with a color. The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false. The color of the area should be changed dynamically for every mouse click. The color should also be changed if we change the color in the “property window “.

Week-7:

- 1) Install TOMCAT web server and APACHE. While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.
- 2) Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root. Access the pages by using the urls:
<http://localhost:4040/rama/books.html> (for tomcat)
<http://localhost:8080/books.html> (for Apache)

Week-8:

User Authentication :

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

1. Create a Cookie and add these four user id's and passwords to this cookie.
2. Read the user id and passwords entered in the Login form (week1). The values (user id and passwords) available in the cookies. If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display “You are not an authenticated user “.Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInit Parameters() method.

Week-9:

Install a database (Mysql or Oracle).Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).Practice 'JDBC' connectivity. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

**Week-10:**

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

Week-11:

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

Week-12:

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

DATA MINING & INFORMATION SECURITY LAB

Course Code: GR14F5036
II Year II Semester

L:0 T:0 P:2 C:2

Prerequisites

- Fundamentals of SQL and Database.
- Experience with programming in some common programming language.
- A good understanding of data structures and algorithms

Course Objectives

- Learn to perform data mining tasks using a data mining toolkit (such as open source WEKA), Understand the data sets and data preprocessing, Demonstrate the working of algorithms for data mining tasks such as association rule mining, classification, clustering and regression, Exercise the data mining techniques with varied input values for different parameters.
- To obtain practical experience using data mining techniques on real world data sets.
- Emphasize hands-on experience working with all real data sets.
- Understand various cryptographic algorithms by implementing and executing them.
- Implementing public-key crypto system.
- Implementing the fundamental ideas of public-key and private-key cryptography.

Course Outcomes

- Ability to design data ware house from heterogenous databases using data preprocessing
- Apply different data mining techniques using data mining soft ware with different data sets
- Ability to know different Data Mining Algorithms
- Ability to work with cryptographic Algorithms
- Apply public and private key Algorithms

Data Warehousing and Data Mining Lab- Problem-Part 1

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.

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. 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. 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. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

The German Credit Data:

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

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

1. List all the categorical (or nominal) attributes and the real-valued attributes separately.
2. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.
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.
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?



5. Is testing on the training set as you did above a good idea ? Why orWhynot ?
6. One approach for solving the problem encountered in the previous question is using cross-validation ? Describe what is cross-validation briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease ?Why ?
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.
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.)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)?
10. 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 ?
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
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.

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 of 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, Unit_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.



Lab- Problem-Part 2

Task 1

- 1) Data Mining Basic Concepts
- 2) WEKA Tool Basic Concepts, How do you load Weka?

Task 2: Introduction to the Weka machine learning toolkit. (To learn to use the Weka machine learning toolkit)

1. What options are available on main panel?
2. What is the purpose of the following in Weka?
 - a) The Explorer
 - b) The Knowledge Flow interface
 - c) The Experimenter
 - d) The command-line interface
3. Describe the arff file format.
4. Press the Explorer button on the main panel and load the weather dataset and answer the following questions
 - a) How many instances are there in the dataset?
 - b) State the names of the attributes along with their types and values.
 - c) What is the class attribute?
 - d) In the histogram on the bottom-right, which attributes are plotted on the X,Y-axes? How do you Change the attributes plotted on the X, Y-axes?
 - e) How will you determine how many instances of each class are present in the data
 - f) What happens with the Visualize All button is pressed?
 - g) How will you view the instances in the dataset? How will you save the changes?
5. What is the purpose of the following in the Explorer Panel?
 - a) The Preprocess panel
 - i. What are the main sections of the Preprocess panel?
 - ii. What are the primary sources of data in Weka?
 - b) The Classify panel
 - c) The Cluster panel
 - d) The Associate panel
 - e) The Select Attributes panel
 - f) The Visualize panel.
6. Load the iris dataset and answer the following questions:
 - a) How many instances are there in the dataset?
 - b) State the names of the attributes along with their types and values.
 - c) What is the class attribute?
 - d) In the histogram on the bottom-right, which attributes are plotted on the X,Y-axes? How do you change the attributes plotted on the X, Y-axes?



- e) How will you determine how many instances of each class are present In the data
- f) What happens with the Visualize All button is pressed?
- 7. Load the weather dataset and perform the following tasks:
 - a) Use the unsupervised filter Remove with Values to remove all instances where the attribute 'humidity' has the value 'high'?
 - b) Undo the effect of the filter.
 - c) Answer the following questions:
 - i. What is meant by filtering in Weka?
 - ii. Which panel is used for filtering a dataset?
 - iii. What are the two main types of filters in Weka?
 - iv. What is the difference between the two types of filters? v. What is the difference between and attribute filter and an instance filter?
- 8. Load the iris dataset and perform the following tasks:
 - a) Press the Visualize tab to view the Visualizer panel.
 - b) What is the purpose of the Visualizer?
 - c) Select one panel in the Visualizer and experiment with the buttons on the panel.

Task 3: Performing data preprocessing tasks for data mining in Weka (To learn how to use various data preprocessing methods as a part of the data mining)

Section -A

Write Procedure and Experiment on Data Pre-processing techniques in WEKA tool using Bank Data set.

- a) Loading the Data
- b) Selecting or Filtering Attributes
- c) Discretization

Section –B: Application of Discretization Filters

1. Perform the following tasks
 - a) Load the 'sick.arff' dataset
 - b) How many instances does this dataset have?
 - c) How many attributes does it have?
 - d) Which is the class attribute and what are the characteristics of this attribute?
 - e) How many attributes is numeric? What are the attribute indexes of the numerical attributes?
 - f) Apply the Naive Bayes classifier. What is the accuracy of the classifier?
2. Perform the following tasks:
 - a) Load the 'sick.arff' dataset.
 - b) Apply the supervised discretization filter.
 - C) What is the effect of this filter on the attributes?



- d) How many distinct ranges have been created for each attribute?
- e) Undo the filter applied in the previous step.
- f) Apply the unsupervised discretization filter. Do this twice:
 - i. In this step, set 'bins'=5
 - ii. In this step, set 'bins'=10
 - iii. What is the effect of the unsupervised filter on the dataset?
- g) Run the Naive Bayes classifier after apply the following filters
 - i. Unsupervised discretized with 'bins'=5
 - ii. Unsupervised discretized with 'bins'=10
 - iii. Unsupervised discretized with 'bins'=20.
- h) Compare the accuracy of the following cases
 - i. Naive Bayes without discretization filters
 - ii. Naive Bayes with a supervised discretization filter
 - iii. Naive Bayes with an unsupervised discretization filter with different values for the 'bins' attributes.

Section-C: Attribute Selection

- a) Perform the following tasks:
 - i. Load the 'mushroom.arff' dataset
 - ii. Run the J48, 1Bk, and the Naive Bayes classifiers.
 - iii. What is the accuracy of each of these classifiers?
- b) Perform the following tasks:
 - i. Go to the 'Select Attributes' panel
 - ii. Set attribute evaluator to CFSSubsetEval
 - iii. Set the search method to 'Greedy Stepwise'
 - iv. Analyze the results window
 - v. Record the attribute numbers of the most important attributes
 - vi. Run the meta classifier AttributeSelectedClassifier using the following:
 - i. CFSSubsetEval
 - ii. GreedyStepwise
 - iii. J48, 1Bk, and NaiveBayes
- c) Record the accuracy of the classifiers
- D) What are the benefits of attribute selection?

Section -D

1. Perform the following tasks:
 - a) Load the 'vote.arff' dataset.
 - b) Run the J48, 1Bk, and Naive Bayes classifiers.
 - c) Record the accuracies.
2. Perform the following tasks:
 - a) Go to the 'Select Attributes' panel
 - b) Set attribute evaluator to 'WrapperSubsetEval'
 - C) Set search method to "RankSearch"



- d) Set attribute evaluator to 'InfoGainAttributeEval'
 - e) Analyze the results
6. Run the metaclassifier AttributeSelectedClassifier using the following:
- i. WrapperSubsetEval
 - ii. RankSearch
 - iii. InfoGainAttributeEval
7. Sampling
- i. Load the 'letter.arff' dataset
 - ii. Take any attribute and record the min, max, mean, and standard deviation of the attribute
 - iii. Apply the Resample filter with 'sampleSizePercent' set to 50 percent
 - iv. What is the size of the filtered dataset? Observe the min, max, mean, and standard deviation of the attribute that was selected in step 2. What is the percentage change in the values?
 - v. Give the benefit of sampling a large dataset.

Task 4: Using Weka to determine Association rules (To learn to use Association algorithms on datasets)

Section -A

Write Procedure and Experiment on Association Rule Mining in WEKA tool using Buy Computer Data set Example.

Section -B

1. Perform the following tasks
 - a) Define the following terms
 - i. item and itemset
 - ii. Association
 - iii. Association rule
 - iv. Support of an association rule
 - v. Confidence of an association rule
 - vi. Large itemset
 - vii. Association rule problem
 - b) What is the purpose of the Apriori Algorithm?
2. Perform the following tasks:
 - a) Load the 'vote.arff' dataset
 - b) Apply the Apriori association rule
 - c) What is the support threshold used? What is the confidence threshold used?
 - d) Write down the top 6 rules along with the support and confidence values.
 - e) What does the figure to the left of the arrow in the association rule represent?



- f) What does the figure to the right of the arrow in the association rule represent?
- g) For rule 8, verify that numerical values used for computation of support and confidences are in accordance with the data by using the Preprocess panel. Then compute the support and confidence values. Are they above the threshold values?

3. Perform the following tasks:

- a) Load the dataset 'weather.nominal.arff'.
- b) Apply the Apriori association rule
 - i. Consider the rule "temperature=hot ==> humidity=normal." Compute the support and confidence for this rule.
 - ii. Consider the rule "temperature=hot humidity=high ==> windy=TRUE." Consider the support and confidence for this rule.
 - iii. Is it possible to have a rule like the following rule: "outlook=sunny temperature=cool" ==> humidity=normal play=yes

4. Perform the following tasks:

- a) Load the bank-data.csv file.
- b) Apply the Apriori association rule algorithm. What is the result? Why?
- c) Apply the supervised discretization filter to the age and income attributes.
- d) Run the Apriori rule algorithm
- e) List the rules that were generated.

Task 5: Classification using the Weka toolkit (To perform classification on data sets using the Weka machine learning toolkit)

Section -A

Write Procedure and Experiment on Classification (different decision trees) in WEKA tool using Bank Data set Example.

Section -B

1. Load the 'weather.nominal.arff' dataset into Weka and run Id3 classification algorithm. Answer the following questions
 - a) List the attributes of the given relation along with the type details
 - b) Create a table of the weather.nominal.arff data
 - c) Study the classifier output and answer the following questions
 - i. Draw the decision tree generated by the classifier
 - ii. Compute the entropy values for each of the attributes
 - iii. What is the relationship between the attribute entropy values and the nodes of the decision tree?
 - d) Draw the confusion matrix? What information does the confusion matrix provide?
 - e) Describe the Kappa statistic?



- f) Describe the following quantities:
 - i. TP Rate
 - ii. FP Rate
 - iii. Precision
 - iv. Recall
2. Load the 'weather.arff' dataset in Weka and run the Id3 classification algorithm. What problem do you have and what is the solution?
3. Load the 'weather.arff' dataset in Weka and run the OneR rule generation algorithm. Write the rules that were generated.
4. Load the 'weather.arff' dataset in Weka and run the PRISM rule generation algorithm. Write down the rules that are generated.

Section -C

1. Load the glass.arff dataset and perform the following tasks?
 - a) How many items are there in the dataset?
 - b) List the attributes are there in the dataset.
 - c) List the classes in the dataset along with the count of instances in the class.
 - d) How will you determine the color assigned to each class?
 - e) By examining the histogram, how will you determine which attributes should be the most important in classifying the types of glass?
2. Perform the following classification tasks:
 - a) Run the 1Bk classifier for various values of K?
 - b) What is the accuracy of this classifier for each value of K?
 - c) What type of classifier is the 1Bk classifier?
3. Perform the following classification tasks:
 - a) Run the J48 classifier
 - b) What is the accuracy of this classifier?
 - c) What type of classifier is the J48 classifier?
4. Compare the results of the 1Bk and the J48 classifiers. Which is better?
5. Run the J48 and 1Bk classifiers using
 - a) the cross-validation strategy with various fold levels. Compare the accuracy results.
 - b) holdout strategy with three percentage levels. Compare the accuracy results.
6. Perform following tasks:
 - i. Remove instances belonging to the following classes:
 - a) build wind float
 - b) build wind non-float
 - ii. Perform classification using the 1Bk and J48 classifiers. What is the effect of this filter on the accuracy of the classifiers?
7. Perform the following tasks:
 - a) Run the J48 and the NaiveBayes classifiers on the following datasets



and determine the accuracy:

- i. weather.arff
- ii. buycomputer.arff
- iii. glass.arff
- iv. iris.arff

On which datasets does the NaiveBayes perform better? Why?

8. Perform the following tasks
 - a) Use the results of the J48 classifier to determine the most important attributes
 - b) Remove the least important attributes
 - c) Run the J48 and 1Bk classifiers and determine the effect of this change on the accuracy of these classifiers. What will you conclude from the results?

Section –D

Load the 'buycomputer.arff' dataset into Weka and run Random Tree classification algorithm. Answer the following questions

- d) List the attributes of the given relation along with the type details
 - e) Create a table of the buycomputer.arff data
 - f) Study the classifier output and answer the following questions
 - iv. Draw the decision tree generated by the classifier
 - v. Compute the entropy values for each of the attributes
 - vi. What is the relationship between the attribute entropy values and the nodes of the decision tree?
 - g) Draw the confusion matrix? What information does the confusion matrix provide?
 - h) Describe the Kappa statistic?
 - i) Describe the following quantities:
 - v. TP Rate
 - vi. FP Rate
 - viii. Recall
2. Load the 'buycomputer.arff' dataset in Weka and run the Id3 classification algorithm. What problem do you have and what is the solution?
 3. Load the 'buycomputer.arff' dataset in Weka and run the OneR rule generation algorithm. Write the rules that were generated.
 4. Load the 'buycomputer.arff' dataset in Weka and run the PRISM rule generation algorithm. Write down the rules that are generated.

Task 6 : Performing clustering using the data mining toolkit. (To learn to use clustering techniques)

Section -A

Write Procedure and Experiment on clustering in WEKA tool using Weather Data set.



Section -B

Perform the following tasks:

1. Load the 'bank.arff' data set in Weka.
2. Write down the following details regarding the attributes:
 - i. names
 - ii. types
 - iii. values.
3. Run the SimpleKMeans clustering algorithm on the dataset
 - a) How many clusters are created?
 - b) What are the number of instances and percentage figures in each cluster?
 - c) What is the number of iterations that were required?
 - d) What is the sum of squared errors? What does it represent?
 - e) Tabulate the characteristics of the centroid of each cluster.
 - f) Visualize the results of this clustering (let the X-axis represent the cluster name, and the Y-axis represent the instance number)
 - i. Is there a significant variation in age between clusters?
 - ii. Which clusters are predominated by males and which clusters are predominated by females?
 - iii. What can be said about the values of the region attribute in each cluster?
 - iv. What can be said about the variation of income between clusters?
 - v. Which clusters are dominated by married people and which clusters are dominated by
 - ii. Unmarried people?
 - i. How do the clusters differ with respect to the number of children?
 - ii. Which cluster has the highest number of people with cars?
 - iii. Which clusters are predominated by people with savings accounts?
 - iv. What can be said about the variation of current accounts between clusters?
 - v. What can be said about the variation of mortgage holdings between clusters?
 - vi. Which clusters comprise mostly of people who buy the PEP product and which ones are comprised of people who do not buy the PEP product?
4. Run the SimpleKMeans algorithm for values of K (no. of clusters) ranging from 1 to 12. Tabulate the sum of squared errors for each run. What do you observe about the trend of the sum of squared errors?
5. For the run with K=12, answer the following questions:
 - a) Is there a significant variation in age between clusters?
 - b) Which clusters are predominated by males and which clusters are predominated by females?
 - c) How do the clusters differ with respect to the number of children?
 - d) Which clusters comprise of people who buy the PEP product and



- which ones are comprised of people who do not buy the PEP product?
- e) Do you see any differences in your ability to evaluate the characteristics of clusters generated for $K=6$ versus $K=12$? Why does this difference arise?

Section-C

Perform the following tasks:

1. Load the 'iris.arff' dataset
2. Write down the following:
 - i. The names of the attributes
 - ii. The types of the attributes
 - iii. The class attribute, its type, and possible values.
3. Run the SimpleKMeans clustering algorithm on the dataset as follows
 - i. Set $K=2$ and observe the sum of squared errors.
 - ii. Set $K=3$ and observe the sum of squared errors.
 - iii. Set $K=4$ and observe the sum of squared errors.
 - iv. Set $K=5$ and observe the sum of squared errors.
 - v. What can be said about the trend of the sum of squared errors? What does this trend imply
 - vi. For $K=3$, tabulate the characteristics of each centroid? How do the clusters correspond to the class values

Section-D

In this task, you would be comparing the performance of three clustering algorithms: Simple K-means, DBScan and Optic based on different parameters.

1. Select the "segment-test.arff" dataset in Weka and run the Simple K-means algorithm on it using 3, 6, 9, and 15 clusters with 2 different distance functions: Euclidean Distance and Manhattan Distance respectively with a 44% percentage split (in total you would need to run it 8 times – 4 clusters with each distance function). You can keep the default values of the remaining parameters like maximum number of iterations, random seed, etc. a) Is there any correlation between the number of clusters and the sum of the squared errors (SSE) for the Euclidean distance function? If there is, why? b) Is there any correlation between the number of clusters and the sum of within cluster distances for the Manhattan distance function? If there is, why?
2. Select the "segment-test.arff" dataset in Weka and run the DBScan algorithm on it using 3 values of epsilon: 0.3, 0.6 and 1.0 for 3 different values of minPoints: 4, 8, and 12 (in total you would need to run it 9 times – 3 for each) with a 44% percentage split. You can keep the default values for the remaining parameters (database type and distance function). a) Is there any correlation between the epsilon values (keeping the minPoints constant) and the number of clustered and un-clustered instances



(noise)? b) As we increase both, the epsilon and minPoints values, what trend do you observe in the number of un-clustered instances (noise)? Why do you think this occurs?

Section –E

1. Write Procedure and Experiment on hierarchical clustering in WEKA tool using Italian cities data set. A hierarchical clustering of distances in kilometers between some Italian cities.

Task 7: Performing Regression techniques using the data mining toolkit. (To learn to use Regression techniques)

Section -A

Write Procedure and Experiment on Simple Linear Regression in WEKA tool using Student Data set.

Task 8: Performing Naive Bayesian Classification techniques using the data mining toolkit. (To learn to use Naive Bayesian Classification)

1. Write Procedure and Experiment on Naive Bayesian Classification technique in WEKA tool using Weather Data set.

Task 9: Performing Data Mining Techniques using the WEKA toolkit and applying Knowledge flow menu.

Section -A

Write Procedure and Experiment on Data Preprocessing techniques in WEKA tool using Buy Computer Data set.

Section –B

Write Procedure and Experiment on Association Rule Mining in WEKA tool using Buy Computer Data set.

Section –C

Write Procedure and Experiment on Classification (different decision trees) in WEKA tool using Weather Data set.

Section –D

Write Procedure and Experiment on clustering in WEKA tool using Buy Computer Data set.

Task 10: Performing Data Mining Techniques using the WEKA toolkit and applying Classification technique using Experimenter menu.



Information Security Lab

List of programs:

1. Write a Java program to perform encryption and decryption using the following algorithms
 - a. Ceaser cipher
 - b. Substitution cipher
 - c. Hill Cipher
2. Write A C/ JAVA program to implement the DES algorithm logic.
3. Write a C/JAVA program to implement the Blowfish algorithm logic.
4. Write a C/JAVA program to implement the AES algorithm logic.
5. Write the RC4 logic in Java.
6. Implement DES-2 and DES-3 using Java cryptography package.
7. Write a Java program to implement RSA algorithm.
8. Implement the Diffie-Hellman Key Exchange mechanism
9. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
10. Calculate the message digest of a text using the MD5 algorithm in JAVA.
11. Explore the Java classes related to digital certificates.
12. Write a program in java, which performs a digital signature on a given text.





III-Year





GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

MOBILE APPLICATION DEVELOPMENT

Course Code: GR14F5038
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisite

- Basic understanding of java programming.
- Need basic OOP concept.

Course Objectives

- To learn the characteristics of mobile applications in J2ME and Android.
- Understand the intricacies of UI required by mobile applications.
- To study about the design aspects of mobile application.
- To understand the best practices to develop mobile application.
- To store data on mobile device.
- To retrieve data from server.
- To realize the differences between developing conventional applications and mobile applications.
- To learn programming skills in J2ME and Android.
- To learn development and programming of mobile applications.

Course Outcomes

Upon Completion of the course, the students should be able to:

- Design and implement the user interfaces of mobile applications in J2ME and Android
- Design the mobile applications that are aware of the resource constraints of the mobile devices.
- Develop advanced mobile applications that accesses network.
- Develop mobile application which uses local storage on mobile device.

Unit-I

J2ME Architecture and Development Environment Java 2 Micro Edition overview Small Computing Technology: Wireless Technology, Radio Data Networks J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, 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-II

Commands, Items, and Event Processing J2ME User Interfaces, Display Class, 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-III

Record Management System and Generic Connection Framework: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener. The Connection, Hypertext Transfer Protocol, Communication Management Using HTTP. Commands, Session Management, Transmit as a Background Process.

Unit-IV

Introduction to Android: Overview Android, Features of Android , Android architecture , Dalvik Virtual Machine & .apk file extension. Fundamentals: Android application life cycle, Basic building blocks – Activities , Services, Broadcast Receivers & Content providers .

UI Components - Views & notifications. Components for communication -Intents & Intent Filters Android API levels (versions & version names)

Unit-V

Android Application Structure: AndroidManifest.xml , ,Resources & R.java,Layouts First sample application, Emulator-Android Virtual Device, Launching emulator, Editing emulator settings, Emulator shortcuts , Second App:- (switching between activities)

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

1. J2ME: The Complete Reference, James Keogh, Tata Mc Graw Hill.
2. Programming for Mobile and Remote Computers, G.T.Thampi, dreamtech press.
3. Beginning Android Application Development by Wei-Meng Lee Wiley India

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. Beginning Android by Mark Murphy Apress
5. Android Application Development: Programming with the Google SDK by Rick Rogers,



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

SOFTWARE TESTING METHODOLOGIES

Course Code: GR14F5039
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites: Object Oriented Analysis and Design with UML, Software Engineering and Software metrics understanding.

Course Objectives

- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- To understand software test automation problems and solutions.
- To learn how to write software testing documents, and communicate with engineers in various forms.

Course Outcomes

- By the end of the course, student should:
- Have an ability to apply software testing knowledge and engineering methods.
- Have an ability to design and conduct a software test process for a software testing project.
- Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
- Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.

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).

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

OBJECT ORIENTED ANALYSIS AND DESIGN

Course Code: GR14F5040
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Knowledge of object oriented concepts
- Knowledge of structured programming
- Students must demonstrate familiarity with several different areas of knowledge and several different modes of inquiry

Course Objectives

- A clear understanding of all the notations used in the UML
- Understanding the key terms like class, object, state, behaviors, object etc
- A detailed understanding of different diagrams in UML
- To model any real worked application using UML diagrams
- Understanding of when to use generalization, aggregation and composition of different kinds of relationships

Course Outcomes

- Upon completion of this course, the student should be able to
- Understand all the standard notation of UML
- Take up a case study and model it in different views with respect to user requirements
- Should be able to apply object-oriented concepts and modeling techniques to simulate real-life situations

Unit-I

Introduction to UML: Importance of modeling, principles of modeling, objectoriented modeling, conceptual model of the UML, Architecture, SoftwareDevelopment 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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-III
WEB SERVICES

Course Code : GR14F5041
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Be aware of Java and Web applications
- Be familiar with details of Client/Server architecture

Course Objectives

- To Understand the details of the core distributed computing technologies
- To learn the underlying web services architecture
- To provide foundations of WSDL and interoperability means in web services
- To explore interoperability between different frameworks.
- To learn basic concepts of SOA.

Course Outcomes

- To have knowledge of interoperability methods between different architecture in web services
- To be aware of limitations of interoperability

Unit-I

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

Unit-II

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

Unit-III

Core fundamentals of SOAP: SOAP Message Structure, SOAP encoding, SOAP message exchange models, SOAP communication and messaging,



SOAP security - Developing Web Services using SOAP: Building SOAP Web Services, developing SOAP Web Services using Java, limitations of SOAP.

Unit-IV

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

Unit-V

Web Services Interoperability: Means of ensuring Interoperability, Overview of .NET and J2EE.

Web Services Security: XML security frame work, XML encryption, XML digital signature, XKMS structure, guidelines for signing XML documents.

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

1. Building Web Services with Java, 2nd Edition, S. Graham and others, Pearson Edn., 2008.
2. Java Web Services, D.A. Chappell & T. Jewell, O'Reilly, SPD.
3. McGovern, et al., "Java Web Services Architecture", Morgan Kaufmann Publishers, 2005.
4. J2EE Web Services, Richard Monson-Haefel, Pearson Education.
5. Web Services, G. Alonso, F. Casati and others, Springer, 2005.
6. Java Web Services Programming, R. Mogha, V.V. Preetham, Wiley India Pvt.Ltd.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-III
DISTRIBUTED DATABASES

Course Code: GR14F5042
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Fundamentals of Relational Algebra, SQL, DBMS
- Fundamentals of Distributed Systems and Middleware - Concepts.

Course Objectives

- The objective of the course is to learn the management of distributed data using distributed database management systems. The student should also acquire insight into difference between the centralized databases and distributed databases, distributed DBMS architecture, query decomposition and data localization, transaction management, distributed concurrency control, client/server architectures and distributed multi-DBMSs.

Course Outcomes

- Key concepts and techniques for centralized databases as well as distributed databases and implementation, such as storage, indexing, query evaluation, query optimization, transaction management, concurrency control, crash recovery, deadlocks and catalog management.
- Analyze and design distributed database systems based on the principles of distributed indexing, query evaluation, data replication, transaction management, concurrency and recovery.
- Demonstrates the principles and techniques for database secure accessing and mining.

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

1. Distributed Database Systems, Chhanda Ray, Pearson.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-III
BIG DATA ANALYTICS

Course Code: GR14F5043
 III Year I Semester

L:4 T:0 P:0 C:4

Course Objectives

- To Understand the fundamentals of big data analytics
- To learn to analyze the big data using intelligent techniques.
- To understand the working and applications of Map Reduce techniques.
- To learn various search and visualization techniques.
- To learn to use various techniques for mining data stream.

Course Outcomes: The students will be able to:

- Work in big data platform
- Analyze the big data analytic techniques for useful business applications.
- Design efficient algorithms for mining the data from large volumes.
- Analyze the HADOOP and Map Reduce technologies associated with big data analytics
- Explore on Big Data applications Using Pig and Hive
- Understand the fundamentals of various big data analysis techniques

Unit-I

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

Unit-II

Mining Data Streams: Introduction To Streams Concepts – Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window - Real time Analytics Platform(RTAP) Applications - Case Studies.

Unit-III

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



Unit-IV

Hadoop Environment: Setting up a Hadoop Cluster - Cluster specification - Cluster Setup and Installation – Hadoop Configuration-Security in Hadoop - Administering Hadoop – HDFS - Monitoring-Maintenance-Hadoop benchmarks-Hadoop in cloud.

Unit-V

Hadoop Frameworks: Applications on Big Data Using Pig and Hive – Data processing operators in Pig – Hive services – HiveQL – Querying Data in Hive - fundamentals of HBase and ZooKeeper - IBM Info Sphere Big Insights and Streams. Visualizations - Visual data analysis techniques, interaction techniques; Systems and applications.

Reference Books

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press (2013)
3. Tom Plunkett, Mark Hornick, "Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop", McGraw-Hill/Osborne Media (2013), Oracle press.
4. Tom White "Hadoop: The Definitive Guide" Third Edit on, O'reilly Media, 2012.
5. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, "Understanding Big
6. Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGrawHill Publishing, 2012
7. AnandRajaraman and Jeffrey David Ullman, "Mining of Masive Datasets", Cambridge University Pres, 2012.



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-III
ETHICAL HACKING

Course Code: GR14F5044
 III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Well founded in programming techniques
- Aware of concepts of computer networks

Course Objectives

- To be aware of the Ethics and legality of a ethical hacker
- To provide students knowledge of different attacks used by hackers
- To give foundation of methods of system attacks, network based attacks on systems, servers and databases

Course Outcomes

- Ability to appreciate the techniques used in real-time attacks
- Develop skills and identify ways of prevent attacks on systems, servers and databases

Unit-I

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

Unit-II

Scanning and Enumeration: Scanning, types of Scanning, CEH Scanning Methodology, Ping Sweep Techniques, Nmap Command Switches, SYN, Stealth, XMAS, NULL, IDLE, and FIN Scans, TCP Communication Flag Types, War-Dialing Techniques, Banner Grabbing and OS Fingerprinting Techniques, Proxy Servers, Anonymizers, HTTP Tunneling Techniques, IP Spoofing



Techniques, Enumeration, Null Sessions, SNMP Enumeration, Windows 2000 DNS Zone Transfer, Steps Involved in Performing Enumeration. System Hacking: Understanding Password-Cracking Techniques, Understanding the Lan Manager Hash Cracking Windows 2000 Passwords, Redirecting the SMB Logon to the Attacker SMB Redirection, SMB Relay MITM Attacks and Countermeasures NetBIOS DoS Attacks, Password-Cracking Countermeasures, Understanding Different Types of Passwords Passive Online Attacks, Active Online Attacks, Offline Attacks Non electronic Attacks, Understanding Keyloggers and Other Spyware Technologies Understand Escalating Privileges, Executing Applications, Buffer Overflows, Understanding Rootkits Planting Rootkits on Windows 2000 and XP Machines, Rootkit Embedded TCP/IP Stack Rootkit Countermeasures, Understanding How to Hide Files, NTFS File Streaming NTFS Stream Countermeasures, Understanding Steganography Technologies, Understanding How to Cover Your Tracks and Erase Evidence, Disabling Auditing, Clearing the Event Log.

Unit-III

Trojans, Backdoors, Viruses, and Worms: Trojans and Backdoors, Overt and Covert Channels, Types of Trojans, Reverse-Connecting Trojans, Netcat Trojan ,Indications of a Trojan Attack, Wrapping, Trojan Construction Kit and Trojan Makers , Countermeasure Techniques in reverting Trojans, Trojan-Evading Techniques, System File Verification Subobjective to Trojan Countermeasures Viruses and Worms, Difference between a Virus and a Worm ,Types of Viruses, Understand Antivirus Evasion Techniques, Understand Virus Detection Methods. Sniffers: Protocols Susceptible to Sniffing, Active and Passive Sniffing, ARP Poisoning, Ethereal Capture and Display Filters, MAC Flooding, DNS Spoofing Techniques, Sniffing Countermeasures

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

Unit-IV

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



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

Unit-V

Linux Hacking: Linux Basics, Compile a Linux Kernel, GCC Compilation Commands, Install Linux Kernel Modules, Linux Hardening Methods Penetration Testing Methodologies: Security Assessments, Penetration Testing Methodologies, Penetration Testing Steps, Pen-Test Legal Framework , Automated Penetration Testing Tools , Pen-Test Deliverables

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-IV
DESIGN PATTERNS

Course Code: GR14F5045
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites: The student should be aware of imperative programming and details of deployment of applications.

Course Objectives

- Analyze a software development problem and express its essence succinctly and precisely.
- Design a module structure to solve a problem, and evaluate alternatives.
- Implement a module so that it executes efficiently and correctly.
- Appreciate engineering issues in the development of software, such as the importance of addressing the user's concerns, working with limited resources, maintainability, dependability, and division of labor.

Course Outcomes

- Identify key entities and relationships in the problem domain; write succinct textual descriptions of problems in the style of a user manual;
- Design programs consisting of a collection of modules exploiting well-known design patterns and devise appropriate module specifications and express them informally in terms of pre- and post- conditions.

Unit-I

Introduction: What Is a Design Pattern?, Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

Unit-II

A Case Study: Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary. **Creational Patterns** Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

Unit-III

Structural Pattern Part-I: Adapter, Bridge, Composite.

Structural Pattern Part-II: Decorator, açade, Flyweight, Proxy.



Unit-IV

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

Behavioral Patterns Part-II: Mediator, Memento, Observer.

Unit-V

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

Text Book

1. Design Patterns By Erich Gamma, Pearson Education

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-IV
INFORMATION RETRIEVAL SYSTEMS

Course Code: GR14F5046
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Logical thinking, Problem Solving techniques
- Should have completed a course on Data Structures and Database Management Systems.

Course Objectives

- To enable student understand the concepts of Information Retrieval Systems.
- To familiarize the techniques of retrievals from large collections of data.
- To analyze the retrieval systems for web search tasks.

Course Outcomes

- Should have acquired a good understanding on fundamentals of information retrieval techniques
- Should be able to apply the IR techniques on large text collections.

Unit-I

Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital libraries and Data Warehouses.

Information Retrieval System Capabilities - Search, Browse, Miscellaneous.

Unit-II

Cataloging and Indexing: Objectives, Indexing Process, Automatic Indexing, Information Extraction.

Data Structures: Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure.

Automatic Indexing: Classes of automatic indexing, Statistical indexing, Natural language, Concept indexing, Hypertext linkages

Unit-III

Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters.

User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, Weighted searches of Boolean systems, Searching the Internet and



hypertext **Information Visualization:** Introduction, Cognition and perception, Information visualization technologies.

Unit-IV

Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems.

Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example – TREC results.

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-IV
HUMAN COMPUTER INTERACTION

Course Code: GR14F5033
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Learner must have basic knowledge about user Interface.
- Should be well familiar with the concepts and terminology of computer.

Course Objectives

- The objectives and Importance of user Interface
- Importance of Screen design and graphical user interface
- Interaction between the Software tools and Interaction Devices.
- Understand Windows and its Components.

Course Outcomes

- Should be able to work with various graphical user interfaces.
- Should be able to understand technological consideration in interface design.
- Should be able to understand Multimedia, colors, uses problems.

Unit-I

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

Unit-II

Design process: Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, and understanding business Functions.

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



Unit-III

Windows – New and Navigation schemes selection of window, selection of devices based and screen based controls.

Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

Unit-IV

Software tools – Specification methods, interface – Building Tools.

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
ELECTIVE-IV
SEMANTIC WEB AND SOCIAL NETWORKS

Course Code: GR14F5048
III Year I Semester

L:4 T:0 P:0 C:4

Prerequisites

- Web Search techniques
- Data mining techniques & Applications

Course Objectives

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

Course Outcomes

- Will be able to build Semantic Web applications with features of Social Networking
- Will be able to distinguish Semantic Web applications from other Web applications.

Unit-I

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

Unit-II

Ontologies and their role in the semantic web: Ontologies Languages for the Semantic Web –Resource Description Framework(RDF) / RDF Schema, Ontology Web Language(OWL),UML,XML/XML Schema.

Ontology Engineering, Constructing Ontology, Ontology Development Tools, Ontology Methods, Ontology Sharing and Merging, Ontology Libraries and Ontology Mapping,

Unit-III

Logic, Rule and Inference Engines: Semantic Web applications and services, Semantic Search, e-learning, Semantic Bioinformatics, Knowledge Base



Unit-IV

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

Unit-V

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

Teaching Methodologies

1. White Board and Marker
2. Power Point Presentations

Text Books

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

Reference Books

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



**GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

MOBILE APPLICATIONS DEVELOPMENT LAB

Course Code: GR14F5049
III Year I Semester

L:4 T:0 P:2 C:2

Prerequisites

- Knowledge of Java programming.
- Knowledge of inheritance, event handling and listeners.

Course Objectives

- To realize the differences between developing conventional applications and mobile applications.
- To learn programming skills in J2ME and Android
- To study about micro browser based applications to access the Internet using Sun Java
- Toolkit and Eclipse
- 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.

Course Outcomes

Upon Completion of the course, the students should be able to:

- Develop useful mobile applications for the current scenario in mobile computing and pervasive computing.
- Use Local Storage on device.
- Develop networked application.
- Use all the features of J2ME and Android
- Port(Deploy) application on mobile device.

Week-1& 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)

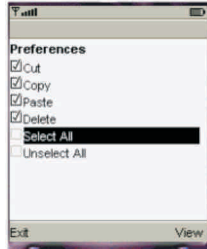
Event Handling.

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



Input checking

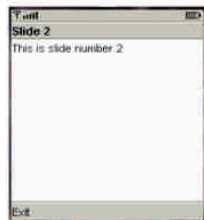
Create an MIDP application which examine, that a phone number, which a user has entered is in the given format.

- * 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&4 Threads & High Level UI:

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.



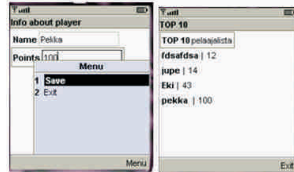
High-level UI

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.

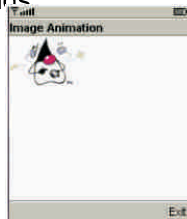




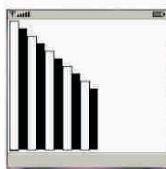
Create a MIDP application, where the user can enter player name and points. The program saves the information to the record using RMS at MIDP device. Program should also print out the top 10 player list to the end user. You can use this class in your game if you made own class for saving and reading record sets.



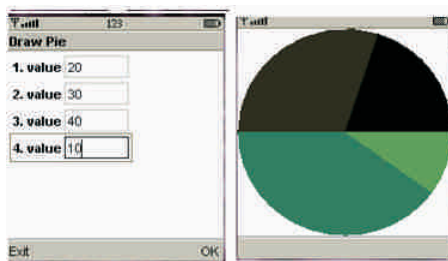
Week-5&6 : Working on Drawing and Images Create a slide show which has three slides, which includes pictures at PNG format. Program should change to the new slide other 5 seconds.



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



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-7&8 : Authentication with a Web Server

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.

Login to HTTP Server from a J2ME Program This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server. Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server.

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

Week-9 & 10: Web Application using J2ME The following should be carried out With respect to the given set of application domains: (Assume that the Server is connected to the well-maintained database of the given domain. Mobile Client is to be connected to the Server and fetch the required data value/information)

- Students Marks Enquiry
- Town/City Movie Enquiry
- Railway/Road/Air (For example PNR) Enquiry/Status
- Sports (say, Cricket) Update
- Town/City Weather Update
- Public Exams (say Intermediate or SSC)/ Entrance (Say EAMCET) Results Enquiry Divide Student into Batches and suggest them to design database according to their domains and render information according the requests.

Android Programs:

Write an Android application program that accepts a name from the user and displays the hello name to the user in response as output using Eclipse.

Write an Android application program that demonstrates the following:

- (i) LinearLayout
- (ii) RelativeLayout
- (iii) TableLayout
- (iv) GridView layout

Write an Android application program that demonstrates intent in mobile application development.

Text Book



1. Beginning J2ME, 3rd edition, Jonathan Konvickalath.
2. J2ME in a Nutshell, Kim Topley, O'Reilly Media
3. Programming Android, O'Reilly Media

Reference Book

1. Developing Mobile Java Applications, Michael Juntao Yuan, Pearson Education



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

SOFTWARE TESTING AND UML LAB

Course Code: GR14F5050
III Year I Semester

L:0 T:0 P:2 C:2

Prerequisites

- Software Testing Lab
- Familiarity with framing test cases
- Knowledge on various software engineering principles
- UML Lab
- Knowledge of object oriented concepts
- Knowledge of structured programming
- Students must demonstrate familiarity with several different areas of knowledge and several different modes of inquiry.

Course Objectives

- Software Testing Lab
- To learn to use the following (or Similar) automated testing tools to automate testing:
 - Win Runner/QTP for functional testing.
 - LoadRunner for Load/Stress testing.
 - Test Director for test management.
- JUnit, HTML Unit, CPP Unit.
- UML Lab
- Understand how UML supports the entire OOAD process.
- Become familiar with all phases of OOAD.

Course Outcomes

- Software Testing Lab
- By the completion of the lab, the student should be able to use automated testing tools like WinRunnerm Test Director, JUnit etc
- UML Lab
- By the completion of the lab, the student should be able to take up any case and do analysis, understand problem space and model the solution using different diagrams with respect to different user requirements.
- Should be able to apply object-oriented concepts and modeling techniques to simulate real-life situations

Unified Modeling Language Lab

Students are divided into batches of 5 each and each batch has to draw the following



diagrams using UML for an ATM system whose description is given below.
UML diagrams to be developed are:

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

Description for an ATM System:

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.) The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

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

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



The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.) If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

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

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

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

Software Testing Lab :

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 he causes for its failure and write down the possible reasons for its failure.
3. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
4. Write the test cases for any known application (e.g. Banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Study of any testing tool (e.g. Win runner)
7. Study of any web testing tool (e.g. Selenium)
8. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
9. Study of any test management tool (e.g. Test Director)
10. Study of any open source-testing tool (e.g. Test Link)
11. Take a mini project (e.g. University admission, Placement Portal) and execute it.



During the Life cycle of the mini project create the various testing documents* and final test report document.

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