UNIVERSITY OF PUNE

BE (COMPUTER ENGINEERING) - 2003 Course

Term I

Subject Code	Subject	Teaching Scheme		Examination Scheme				Mark
		Lect.	Pract	Th	Tw	Pr	Or	
410441	Design & Analysis of Algorithms	04		100		—		100
410442	Operating Systems	03	—	100	—	—	—	100
410443	Object Oriented Modeling & Design	03	02	100	25	—	50	175
410444	Principles of Compiler Design	04	—	100	—	—	—	100
410445	Elective I	04	02	100	25	—	50	175
410446	Computer Laboratory I	—	04	—	50	50	—	100
410447	Project Work	—	02	—	50	—	—	50
	Total	**18**	**10**	**500**	**150**	**50**	**100**	**800**

Term II

Subject Code	Subject	Teaching Scheme		Examination Scheme				Mark
		Lect.	Pract	Th	Tw	Pr	Or	
410448	Networks and Information Security	04	—	100	—	—	—	100
410449	Advanced Computer Architecture and Computing	04	—	100	—	—	—	100
410450	Software Testing and Quality Assurance	04	—	100	—	—	—	100
410451	Elective II	04	02	100	25	—	50	175
410452	Computer Laboratory II	—	04	—	25	50	—	75
410447	Project Work	—	06	—	100	—	50	150
	Total	**16**	**12**	**400**	**200**	**50**	**100**	**700**

Th: Theory **Tw: Term Work** **Pr: Practical** **Or: Oral**

Elective I

1) Image Processing
2) Advanced Databases
3) Artificial Intelligence
4) Multimedia Systems

Elective II

1) Distributed Systems
2) Software Architecture
3) Embedded Systems
4) High Performance Networks

410441 Design and Analysis of Algorithms

Teaching Scheme : **ExaminationScheme :**
Theory:4 Hours/Week **Theory:100 Marks**
Duration: 3 Hrs

Objectives :

- To study and perform analysis of algorithms.
- To study techniques/strategies in design of algorithms

UNIT I :

Introduction :

'O','Ω' and 'θ' asymptotic notations, Average, Best and Worst case analysis of algorithms for Time and Space complexity, Amortized Analysis, Solving Recurrence Equations, Proof Techniques: by Contradiction, by Mathematical Induction.

Priority Queues : Heaps & Heap sort. **(8)**

UNIT II :

Divide And Conquer And Greedy Strategy :

Divide and Conquer: General Strategy, Exponentiation. Binary Search, Quick Sort and Merge Sort. Greedy Method, General Strategy, Knapsack problem, Job sequencing with Deadlines, Optimal merge patterns, Minimal Spanning Trees and Dijkstra's algorithm. **(9)**

UNIT III :

Dynamic Programming :

General Strategy, Multistage graphs, OBST, 0/1 Knapsack, Traveling Salesperson Problem, Flow Shop Scheduling. **(7)**

UNIT IV :

Backtracking & Branch And Bound:

Backtracking: General Strategy, 8 Queen's problem, Graph Coloring, Hamiltonian Cycles, 0/1 Knapsack.

Branch and Bound: General Strategy, 0/1 Knapsack, Traveling Salesperson Problem . **(8)**

UNIT V :

Parallel Algorithms:

Computational Model, Basic Techniques and Algorithms (Complete Binary Tree, Pointer Doubling, Prefix Computation), Selection, Merging, Sorting Networks, Parallel Sorting, Graph Problems (Alternate Algorithm for Transitive Closure, All pairs shortest path) **(8)**

UNIT VI :

NP-Hard And NP-Complete Problems:

Algorithms, Complexity-intractability, Non-Deterministic Polynomial time (NP) Decision problems, Cooks Theorem.

NP-Complete problems- Statisfiability problem, Vertex cover problem.

NP-Hard problems-graph, scheduling, code generation problems, Simplified NP Hard Problems. **(6)**

Text Books :

1. Bressard, "Fundamental of Algorithm." , PHI X 14263
2. Horowitz and Sahani, "Fundamentals of computer Algorithms", Galgotia. 8656

References :

1. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm" PHI 1805
2. A. V. Aho and J.D. Ullman, "Design and Analysis ofAlgorithms", Addison Wesley 13981 RR

410442 Operating Systems

Teaching Scheme
Lectures : 3 Hrs/Week

Examination Scheme
Theory : 100 Marks
Duration: 3 Hrs.

Objectives

- To study advanced concepts of Operating Systems
- To study the comparison of different function of different OSs.

UNIT I :

Process synchronization and inter-processcommunication : Background, Critical section problem, semaphores, classic problems of synchronizations, critical regions, monitors, OS synchronization **(4)**

UNIT II :

Deadlocks : System model, deadlock characterization, methods for handling deadlocks, deadlocks detection, prevention, avoidance and recovery

Protection : Goals of protection, domain of protection, Access matrix, Implementation of Access matrix, Revocation of access rights

Security : Security problem, User authentication, program threats, system threats, securing system and facilities, Intrusion detection and cryptography **(6)**

UNIT III :

System Architecture, User perspective, Operating System Services, Assumptions about hardware, Architecture of UNIX OS, Introduction to System Concepts, Kernel Data Structures, System Administration, Buffer Headers, Structure of the Buffer Pool, Scenarios for retrieval of a Buffer, Reading and Writing Disk Blocks, Advantages and Disadvantages of the Buffer Cache **(8)**

UNIT IV :

Internal Representation of files: Inodes, Structure of a regular file, Directories, Conversion of a path name to an Inode, Super Block, Inode assignment to a new file, Allocation of Disk Blocks, Other file types.

System Calls for the file system: open, read, write, file & Record Locking, lseek, Close, File Creation, Creation of Special Files, Change Directory and Change Root, Change Owner and Change Mode, stat and fstat, pipes, dup, mounting and unmounting file systems, link, unlink, file system abstrations, file system maintenance **(8)**

UNIT V :

Structures of processes: Process States and Transitions, Layout of System Memory, The Context of a process, Saving the context of a process, Manipulation of the process address space, Sleep.

Process control: Process Creation, Signals, Process Termination, Awaiting Process Termination, Invoking other programs, The User ID of a process, Changing the size of a process, The shell, System boot and the init process,

Process Scheduling and Time: Process Scheduling, System calls for time & clock **(8)**

UNIT VI :

Memory Management Policies: Swapping, Demand Paging, a hybrid system with swapping and demand paging **I/O Subsystem:** Driver interfaces, Disk drivers, Terminal drivers, Streams **(4)**

Text Books :

1. William Stallings, "Operating System-Internals and Design Principles ", 5/e, Prentice Hall India, ISBN-81-297-01094-3
2. Silberschatz, Galvin, Gagnes , "Operating System Concepts" 6/e , John Wiley & Sons, ISBN-9971-51-388-9

Reference Books :

1. Dhamdhre D. M.,"Operating Systems - A Concept-Based Approach", Tata McGraw Hill Publications, 2nd Edition-2006
2. Andrew S. Tanenbaum, " Modern Operating Systems", 2/e, Prentice Hall India, ISBN-81-203-2063-8
3. Maurice J. Bach, "The design of the UNIX Operating System", Prentice Hall India, ISBN-81-203-0516 7

410443 Object Oriented Modeling and Design

Teaching Scheme	Examination Scheme
Lectures : 3 Hours /Week	Theory : 100 Marks
Practical : 2 Hours/Week	Term Work : 25 Marks
	Oral : 50 Marks
	Duration : 3 Hrs.

Objectives :

- Introduction to Modeling and Design of software, firmware and business processes.
- Introduce UML 2.0 and its diagrams as a modeling tool for large and complex systems.
- Understand the concepts being modeled in UML

UNIT I :

Introduction to OMG Standards : MDA, MOF, XML, CORBA , UML 2.0. UML History, UML 2.0 New Features.

Rational Unified Process emphasizing Inception, Elaboration, Construction, Transition Phases. 4+1 View architecture, Architectural approaches : Use case Centric, Architecture driven, Iterative approach, OO Concepts Review .

UNIT II :

Introduction to UML : UML MetaModel, Extensibility mechanisms like stereotypes, tagged values, constraints and profiles. OCL.Overview of all diagrams in UML 2.0.

UNIT III :

Object diagrams : CRC method, Review of OO concepts.Class diagrams, Classes and Relationships, Interfaces and ports, Templates, Active Objects, Advanced relationships generalization, association, aggregation, dependencies. Composite structure diagrams including composite structures, collaborations.

UNIT IV :

Interaction diagrams : Interaction Overview diagrams including interactions, signals, exceptions, regions, partitions, Sequence diagrams, Communication diagrams.

UNIT V :

State Machine diagrams : States, encapsulation of states, transitions, submachine, state generalization. Timing diagrams, Activity diagrams, Activities, sub activities, signals, exceptions, partitions, regions.

UNIT VI :

Support for modeling Architecture in UML. Package diagrams, Component diagrams, Deployment diagrams. Applications of UML in embedded systems, Web applications, commercial applications.

All diagrams are to be assumed for UML 2.0. For each diagram the need, purpose, Concepts, Notation, Forward Engineering, Reverse Engineering & Application must be considered.

Text Books :

1. Grady Booch, James Rumbaugh, Ivar Jacobson "Unified Modeling Language User Guide", The (2nd Edition) (Addison-Wesley Object Technology Series) (Hardcover)

Reference Books :

1. Joseph Schmuller "SAMS Teach yourself UML in 24 Hours", Third edition.
2. Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third Edition (Paperback) ,Addision Wesley
3. Dan Pilone, Neil Pitman "UML 2.0 in a Nutshell", (In a Nutshell (O'Reilly)) Paperback)
4. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado "UML 2 Toolkit (Paperback) "
5. Jim Arlow, Ila Neustadt "UML 2 and the Unified Process : Practical Object-Oriented Analysis and Design" (2nd Edition) (Addison-Wesley Object Technology Series) (Paperback)
6. Michael Jesse, James A. Schardt "UML 2.0 for dummies "
7. Kendal Scott, Apress "Fast track UML 2.0 "

Object Oriented Modeling and Design Lab

Objectives of the Laboratory:

- To learn how to understand the requirements of a system, its scope.
- To learn good design, good modeling practices, document them and be able to discuss the pros and cons of your designs and models.
- To learn issues in modeling large, complex systems with example hypothetical systems
- To learn concepts, best practices in software, firmware development today and explore UML 2.0 basic and advanced concepts and notation for the same:
- To use UML 2.0 diagrams for modeling different aspects of a system throughout the SDLC lifecycle
- To model an entire system using UML 2.0
- To learn effective use of any CASE TOOL for UML 2.0.

To meet above objectives teachers will help students choose a hypothetical system preferably either a commercial, web to based or embedded system for modeling. The students will try and identify scope of such a system as realistically as possible. Students will learn to draw, discuss different UML 2.0 diagrams, their concepts, notation, advanced notation, forward and reverse engineering aspects. As far as possible draw as many diagrams for one single system, unless they are not applicable for the chosen system in which case other systems may be chosen for specific diagrams.

Any 8 diagrams can be drawn using tool, the other diagrams can be drawn on paper. Optionally one may draw Interaction overview diagrams, timing diagrams, composite structure diagrams and object diagrams for your system as study assignments, paper based assignment or in cases relevant CASE TOOL.

The write-ups for any diagram can include small examples to cover notation that has not been referred to in your submitted diagram. Generally any UML diagram has accompanied document to explain the diagram further. For example use case descriptions, non-functional requirements, scripts, notes, assumptions, project management aspects.

Assignments :

1. Choose a hypothetical system of significant complexity and write an SRS for the same.
2. Draw one or more Use Case diagrams for capturing and representing requirements of the system. Use case diagrams must include template showing description and steps of the Use Case for various scenarios.
3. Draw one or more Package diagram to organize and manage your large and complex systems as well as their complex models.
4. Draw activity diagrams to display either business flows or like flow charts.
5. Draw basic class diagrams to identify and describe key concepts like classes, types in your system and their relationships.

6. Draw advanced class diagrams to depict advanced relationships, other classifiers like interfaces.
7. Draw sequence diagrams or communication diagrams with advanced notation for your system to show objects and their message exchanges.
8. Draw state machine to model the behavior of a single object, specifying the sequence of events that an object goes through during its lifetime in response to events.
9. Draw component diagrams assuming that you will build your system reusing existing components along with a few new components.
10. Draw deployment diagrams to model the runtime architecture of your system.

410444 Principles Of Compiler Design

Teaching Scheme	Examination Scheme
Lectures : 4 hrs/week	Theory : 100 Marks
	Duration : 3 Hrs

Objectives

- To learn and understand the design of a compiler
- To learn and use tools for construction of a compiler

UNIT I:

Introduction to Compiler :

Translator issues, why to write compiler, compilation process in brief, front end and backend model, compiler construction tools, Interpreter and the related issues, Cross compiler, Incremental compiler, Boot strapping, byte code compilers

Lexical Analysis

Review of lexical analysis: alphabet, token, lexical error, Block schematic of lexical analyser, Automatic construction of lexical analyser (LEX), LEX specification and features. **(6)**

UNIT II :

Syntax Analysis

Introduction: Role of parsers, Top down-RD parser, Predictive parsers, LL (k) parsers, Bottom up Parsers - Operator precedence parsers, shift-Reduce: SLR, LR (k), LALR etc. using ambiguous grammars, Error detection and recovery, Automatic construction of parsers (YACC), YACC specifications

Semantic Analysis

Need of semantic analysis, type checking and type conversion **(10)**

UNIT III :

Syntax directed translation

Syntax directed definitions, construction of syntax trees, bottom-up evaluation of S-attributed definitions, L-attributed definitions, Top-down translation, bottom-up evaluation of inherited attributes

Intermediate Code Generation

Intermediate languages, declarations, assignment statements, iterative statements, case statements, arrays, structures, conditional statements, Boolean expressions, back patching, procedure calls, Intermediate code generation using YACC **(10)**

UNIT IV :

Run Time Storage Organisation

Source language issues, Storage organization and allocation strategies for block structured and non block structured languages, Activation record, variable-length data, procedure parameters, nested procedures, access to non-local names, procedure Call and return, static and dynamic scope, Symbol Table organisation and management **(6)**

UNIT V :

Code Generation :

Introduction: Issues in code generation, Target machine description, Basic blocks and flow graphs, next-use representation of basic blocks, Peephole optimisation, DAG Generating code from a DAG, Dynamic programming, Code generator-generator concept. **(8)**

Unit VI :

Code Optimisation

Introduction, Classification of optimisation, Principle sources of optimisation, optimisation of basic blocks, Loops in flow graphs, Optimising transformations: compile time evaluation, common sub-expression elimination, variable propagation, code movement, strength reduction, dead code elimination and loop optimisation, Local optimisation, DAG based local optimisation. Global optimisation: control and data flow analysis, Computing global data flow information: meet over paths, Data flow equations, Data flow analysis, Iterative data flow analysis: Available expressions, live range identification **(8)**

Text Books

1. A V Aho, R. Sethi, J D Ullman, "Compilers: Principles, Techniques, and Tools", Pearson Education, ISBN 81 - 7758 - 590 - 8

Reference Books

1. K. Cooper, L. Torczon, "Engineering a Compiler", Morgan Kaufmann Publishers, ISBN 81 - 8147 - 369 - 8
2. K. Louden, "Compiler Construction: Principles and Practice", Thomson Brookes/Cole (ISE), 2003, ISBN 981 - 243 - 694 - 4
3. J. R. Levine, T. Mason, D. Brown, "Lex & Yacc", O'Reilly, 2000, ISBN 81-7366 - 062 - X
4. S. Chattopadhyay, "Compiler Design", Prentice-Hall of India, 2005, ISBN 81-203-2725-X.

410446 Computer Laboratory-I

Teaching Scheme
Practical: 4 hrs/week

Examination Scheme
Practical: 50 Marks
Term work: 50 Marks

Objectives

- To learn and use compiler writing tools
- To understand and implement algorithms used by an operating system

PART I : Assignments related to Compilers

LEX

1. Assignment to understand basic syntax of LEX specifications, built-in functions and Variables
2. Implement a lexical analyser for a subset of C using LEX Implementation, should support error handling.

YACC

1. Assignment to understand basic syntax of YACC specifications, built-in functions and Variables
2. Write an ambiguous CFG to recognise an infix expression and implement a parser that recognises the infix expression using YACC. Provide the details of all conflicting entries in the parser table generated by LEX and YACC and how they have been resolved.

Syntax Directed Translation

1. Write an attributed translation grammar to recognise declarations of simple variables, for, assignment, if, if-else statements as per syntax of C or Pascal and generate equivalent three address code for the given input made up of constructs mentioned above using LEX and YACC. Write a code to store the identifiers from the input in a symbol table and also to record other relevant information about the identifiers. Display all records stored in the symbol table.

Laboratory Project

1. For a small subset of C with essential programming constructs, write a compiler using LEX and YACC. (To be carried out in a group of 4 to 6 students).

PART II: Assignments related to Operating Systems

1. Study of UNIX Commands
2. Shell Programming and AWK Programming with suitable application and use of advanced filters, AWK Report Generation.
3. Using fork system call creates child process, suspend it using wait system call and transfer it into the zombie state.
4. Client - Server communication using following IPC mechanism
    1. Unnamed pipe
    2. Named pipe
    3. Semaphore
5. File management using low level file access system calls such as write, read, open, lseek, fstat
6. Implement an Alarm clock application using signals
7. Simultaneous execution of two threads
8. Write & insert a module in Linux Kernel

PART III : Design And Analysis Of Algorithms Laboratory

Minimum 6 assignments based on the following topics. (Atleast 1 assignment must be performed on each of the strategies mentioned in the theory syllabus)

1. Recursive and iterative(non recursive) algorithm for specific problem and their complexity measures(comparison expected).
2. Quick Sort/ Merge Sort implementations using divide and conquer approach. Time complexity measure is to be obtained.
3. Minimal spanning Trees/ Job scheduling as an example of Greedy approach
4. Finding shortest path for multistage graph problem. (single source shortest path and all pairs shortest path.)
5. OBST/Flow Shop Scheduling as an example of dynamic programming.
6. 0/1 knapsack's problem using Dynamic Programming, Backtracking and Branch & Bound Strategies.
7. 8-Queens problem/ Graph coloring problem : general backtracking method and recursive back tracking method and their comparison for space and time complexity.
8. A complete LC branch and bound algorithm for job sequencing with dead lines problem. Use fixed tuple size formulation.
9. Algorithm implementation for 'Traveling salesman' problem using -
   (a) Dynamic programming approach.
   (b) Branch & Bound approach.
10. Simulation/ Implementation of any Parallel Algorithms.

Reference Books :

1. Brian W. Kernighan, Rob Pike, "The UNIX Programming Environment" Prentice Hall India, ISBN: 81-203-0499-3
2. W. Richard Stevens, "UNIX Network Programming", 2/e Volume 2, Prentice Hall India, ISBN: 81-203-2062-X
3. Neil Matthew, Richard Stones, "Beginning Linux Programming", 3/e, WROX Publication, ISBN: 81-265-0484-6
4. Daniel P. Bovet, Marco Cesati, "Understanding the Linux Kernel, Second Edition", 2/e, ISBN: 0-596-00213-0
5. Richard L. Peterson, "The Complete Reference, Linux", 5/e, Tata McGraw Hill, ISBN: 0-07-052489-7

410445 Advanced Databases

Teaching Scheme :	**Examination Scheme :**
Lecturers : 4 Hrs. / Week	**Theory : 100 Marks**
Practical : 2 Hrs / Week	**Term Work : 25 Mark**
	Oral : 50 Mark
	Duration : 3 Hrs.

Objectives :

- To learn and understand advances in Database System Implementations
- To learn and understand various database architectures and applications.

UNIT I :

Parallel databases

Introduction, Parallel database architecture, I/O parallelism, Inter-query and Intra-query parallelism, Inter-operational and Intra-operational parallelism, Design of parallel systems

UNIT II :

Distributed Databases

Introduction, DDBMS architectures, Homogeneous and Heterogeneous Databases, Distributed data storage, Distributed transactions, Commit protocols, Concurrency control in distributed databases, Availability, Distributed query processing , Directory systems

UNIT III :

Web based systems

Overview of client server architecture, Databases and web architecture, N-tier architecture, Business logic - SOAP XML - Introduction, XML DTD's, Domain specific DTD's , Querying XML data

UNIT IV :

Data Warehousing

Introduction to Data warehousing, architecture, Dimensional data modeling- star, snowflake schemas, fact constellation, OLAP and data cubes, Operations on cubes, Data preprocessing - need for preprocessing, data cleaning, data integration and transformation, data reduction

UNIT V :

Data Mining

Introduction to data mining, Introduction to machine learning, descriptive and predictive data mining, outlier analysis, clustering - k means algorithm, classification - decision tree, association rules - apriori algorithm, Introduction to text mining, Baysian classifiers.

UNIT VI :

Information Retrieval

Information retrieval - overview , Relevance ranking using terms and hyperlinks, synonyms, homonyms, ontologies, Indexing of documents, measuring retrieval effectiveness , web search engines, Information retrieval and structured data .

List of assignments

1. **ORDBMS -** Implement system using composite, multivalued attributes, inheritance
2. Directory systems - address book using LDAP
3. Web based system using ASP / JSP
4. Building cubes and OLAP analysis
5. Data mining algorithms
6. Case Study :
   a) any one from open source (eg : Postgres SQL , MySQL)
   b) Any one from Oracle, SQL Server, DB2

Text Books :

1. Abraham Silberschatz, Henry Korth, S. Sudarshan, "Database system concepts", 5th Edition , McGraw Hill International Edition
2. Jiawei han, Micheline Kamber, "Data Mining : Concepts and systems" , Morgan Kaufmann publishers

Reference Books

1. Rob Coronel, Database systems : "Design implementation and management", 4th Edition, Thomson Learning Press
2. Raghu Ramkrishnan, Johannes Gehrke, "Database Management Systems", Second Edition, McGraw Hill International Edition

410445 Artificial Intelligence

Teaching Scheme	Examination Scheme
Lectures : 4Hrs/Week	Theory : 100 marks
Practical. : 2Hrs/Week	Term Work : 25 Marks
	Oral : 50 Marks
	Duration : 3 Hrs

Objectives

- To understand the concepts of Artificial Intelligence
- To Learn and Understand the knowledge representation techniques for knowledgebase
- To Learn and Understand the fundamentals of Neural Network

UNIT I :

Introduction

Definition, What is A.I ? Foundation of A.I., History, Intelligent Agents, Agent Architecture, A.I. Application-(E Commerce, & Medicine), A.I. Representation, Properties of internal representation . Future of A.I ,Production System, Issue in design of search programs

Logic Programming

Introduction , Logic, Logic Programming, Forward and Backward resoning, Forward and Backward chaining rules **(8)**

UNIT II :

Heuristic search techniques

Heuristic search, Hill Climbing, Best First Search, mean and end analysis, Constraint Satisfaction, A* and AO* Algorithm.

Game playing

Minmax search procedure, Alpha beta cutoffs, waiting for Quiescence, Secondary search. **(7)**

UNIT III :

Knowledge Representation

Basic of Knowledge representation, Knowledge representation Paradigrams, Propositional Logic, Inference Rules in Propositional Logic, Knowledge representation using Predicate logic, Predicate Calculus, Predicate and arguments, ISA hierarchy, Frame notation , Resolution , Natural Dedication

Knowledge Representation using non monotonic logic:

TMS (Truth Maintenance System), statistical and probabilistic reasoning, fuzzy logic, structured knowledge representation, semantic net, Frames, Script, Conceptual dependency. **(10)**

UNIT IV :

Learning:

What is Learning? Types of Learning (Rote , Direct instruction Analogy, Induction, Deduction)

Planning:

Block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, least commitment strategy. **(7)**

UNIT V :

Advance AI Topics

Natural Language Processing

Introduction, Steps in NLP , Syntactic Processing , ATN, RTN, Semantic analysis, Discourse & Pragmatic Processing

Perception Perception, Action, Robot Architecture. **(8)**

UNIT VI :

Neural Networks:

Introduction to neural networks and perception-qualitative Analysis. Neural net architecture and applications.

Expert system:

Utilization and functionality, architecture of expert system, knowledge representation, two case studies on expert systems. **(8)**

Text Books

1. Eugene, Charniak, Drew Mcdermott: "Introduction to artificial intelligence."

2. Elaine Rich and Kerin Knight: "Artificial Intelligence."

3. Kishen Mehrotra, Sanjay Rawika, K Mohan; "Artificial Neural Network."

Reference Book

1. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Prentice Hall, 2nd Edition.
2. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley, 1990.
3. Herbert A. Simon , "The Sciences of the Artificial ", MIT Press, 3rd Edition (2nd Printing),1998.
4. Tim Jones "Artificial Intelligence Application Programming" M. Dreamtech Publication

Laboratory work:

Assignment based on:

1. Implement 8 puzzle problem using A* algorithm
2. Implement AO* algorithm for Tower of Hanoi
3. Implementation of Unification Algorithm.
4. Implementation of Truth Maintenance System using Prolog
5. Implementation of Min/MAX search procedure for game Playing
6. Parsing Method Implementation using Prolog.
7. Development of mini expert system using Prolog / Expert System Shell " Vidwan"

   Designed By NCST Mumbai.

Staff should frame any six assignments on above topics.

410445 Image Processing

Teaching Scheme
Lectures: 4 Hours / Week
Practical: 2 Hours / Week

Examination Scheme
Theory: 100 Marks
T/W: 25 Marks
Oral: 50 Marks
Duration:3 Hrs.

Objective:

- Introducing The Concept Of Image In Digital Form
- Analysis Of Digitized Images.
- Processing And Restoration Of Images

UNIT I :

Introduction

Scenes And Images, Application Of Image Processing, Image Processing System (Hardware, Software) , Vector Algebra, Orthogonal Transform, Fuzzy Sets And Properties

UNIT II :

Image Formation And Digitization

Geometric Model, Photometric Model, Sampling, Digitization , Elements Of Digital Geometry, Image Properties , Representation.

UNIT III :

Image Processing

Image Enhancement - Contrast Intensification, Smoothening, Sharpening Image Restoration — Square Error Restoration Techniques, Singular Value Decomposition, Homomorphic Filtering. Image Compressing - Basic - Lossy Compression, Loss-Less Compression

UNIT IV :

Segmentation : Region Extraction, Thresholding, Texture Based, Histogram Analysis

Edge Detection : Preventive Operators, Edge Filtering Masks

Feature Extraction : Representation, Topological Attributes, Geometric Attributes, Spatial Features

Recognition : Deterministic, Statistical ,Fuzzy

UNIT V :

Color image processing and Morphology

Color Models, Pseudo coloring And Color Displays, Spatial Interpolation, Binary Morphology -Dilation, Crosion Thinning, Thickening And Pruning, Gray Level Morphology

UNIT -VI :

Application Of Image Processing

Image Processing in Multimedia , Medical Images Capture And Processing, Satellite Imagery, Stereography, Water Marking

Text Books :

1. B. Chanda, D.Datta Mujumdar, "Digital Image Processing And Analysis", PHI , 5th Reprint ISBN-81-203-1618-5
2. R.C. Gonzalez, R.R. Woods, "Digital Image Processing Person Education ", ISBN - 81-7808-629-8

Reference:

1. William Pratt, "Digital Image Processing", John Willey & Sons Inc. , ISBN-9-814- 12620-9
2. Anil Jain, "Fundamentals Of Digital Image Processing", Anil Jain PHI, ISBN-81-203-0929-4

Image Processing Lab

Assignments for Image Processing

1. Develop C / C++ code to create a simple image and save the same as bitmap image in .bmp file.
2. Develop C / C++ code to perform basic image enhancement operations like; sharring, smoothening using filtering masks
3. Develop C/C++ code to implement image compression (any one algorithm)
4. Implement gray scale thresholding to blur an image.
5. Using derivative filtering technique implement an algorithm for edge detection and further thinning the edge.
6. **Mini Project**: Instructor can give captured image from any of the application area and a group of students will be assigned a set of IP operations to analyze and extract all the features of the given image.

   The results (output) may be viewed and compared using any standard package.

   Students will submit the term work in the form of journal. The journal will contain minimum 5 assignments and a mini project. The instructor will frame atleast one assignment on each of the areas specified above. Oral examination will be based on the term work submitted.

410445 Multimedia Systems

Teaching Scheme	Examination Scheme
Lectures : 4 Hrs/Week	Theory : 100 marks
Practical. : 2 Hrs/Week	Term Work : 25 Marks
	Oral : 50 Marks
	Duration : 3 Hrs.

Learning Objectives :

- Learn key concepts of Multimedia Systems.
- Learn to design multimedia projects independently.

UNIT I :

Introduction:

What is multimedia, Goals and objectives, characteristics of multimedia presentation, multimedia applications, Multimedia building blocks, multimedia and internet, multimedia architecture, multimedia authoring tools, overview of multimedia software tools, multimedia Document Architecture (MHEG, SGML, ODA,OMF etc.) **(6)**

UNIT II :

Multimedia Operating System:

Resource Management, Process management, processor scheduling algorithms, multimedia file system and algorithms.

Multimedia DBMS

Characteristics of multimedia DBMS, data structures and examples of multimedia structures, operations on data, integration in database model. **(5)**

UNIT III :

Digital Image Processing:

Basic Image fundamentals, Image data types, image file formats (GIF, BMP, TIFF, JPEG, PCX etc), Image acquisition, storage processing, Communication, and display, Image enhancement: Enhancement by point processing,

Spatial filtering, Color image processing, Image compression techniques: Lossy, Lossless, Hybrid methods, Lossless: Shannon- Fano algorithm, Arithmetic coding. Lossy : Vector quantization, fractal compression technique. Hybrid: JPEG-DCT **(10)**

UNIT IV :

Multimedia Audio:

Basic sound concepts, audio file formats, RIFF, VOC, WAV, MIDI, AVI, and AVO.

MPEG-I, MPEG-II, MPEG-IV, MPEG Audio and Video Compression Techniques. **(8)**

UNIT V :

Virtual Reality and Multimedia:

Concept, VR devices: Hand Gloves, Head mounted tracking system, VR chair, CCD, VCR, 3D Sound system, Head mounted display. Virtual Objects - Basics of VRML. **(5)**

UNIT VI :

Operating System Support to Multimedia:

Multimedia function calls, Windows support for sound, animations, movies, music and MIDI controls. Introduction to UNIX support to Multimedia.

Introduction to Multimedia networks and applications: Quality of multimedia data transmission, Multimedia over IP, Multimedia over ATM Networks. **(8)**

Text Book

1. Judith Jeffcoate "Multimedia in Practice":, PHI.
2. Ralf Steinmetz and Klara Nahrstedt "Multimedia Computing, Communication and Applications", Pearson Education.

References :

1. Ze-Nian Li Marks S. Drew "Fundamentals of Multimedia", Pearson Education.
2. Nigel Chapman and Jenny Chapman. Wiley "Digital Multimedia"
3. A. K. Jain "Fundamentals of Digital Image Processing", PHI
4. Gonzalez, Woods, "Digital Image Processing" Addison Wesley
5. Mark Nelson "Data Compression Book ", BPB.
6. Ranjan Parekh ,"Principals of Multimedia ",TMH.

Multimedia systems Lab

Assignments :

1. Write a menu Driven program
   a) Create, Edit .VOC file and convert it to .WAV file format
   b) Create, Edit .WAV file and convert it to .VOC file format.
2. Write a program to develop animated clip
3. Write a tool to create presentation slide with audio & video effects
4. Using VRML generate any one virtual scene
   1. Coffee House
   2. Building Model
   3. Garden Model
5. Implement Edge Detection Algorithm on any Image
6. Implement arithmetic coding technique on any image.
7. Implement JPEG-.DCT compression Technique

410447 Project Work

Teching Scheme	Examination Scheme
Practical : 2 Hours/Week	Term Work : 150 Marks
	Oral : 50 Marks

Objectives :

1) To allow the students to select a project in the area of Computer Engineering or Information Technology of his/her choice, study the feasibility of it and plan the project properly for two terms of the year.

   The project will be undertaken preferably by a group of maximum 4 students and minimum 3 students who will jointly work and implement the project. The group will select a project with the approval of project coordinator /guide and submit the name of the project with synopsis of not more than 2 to 3 pages not later than the second week of July in the academic year.

Term I

Teaching Scheme	Examination Scheme
Practical : 2 Hours/Week	Term Work : 50 Marks

1) A preliminary project report consisting of a **problem definition, literature survey, platform choice, SRS(System Requirement Specification) Document in specific format and high level design document**

at the end of term-I.

2) The project work will be assessed by internal examiner (preferably the guide) having four or more years of experience and an external examiner from other college having five or more years of experience.
3) The pair of examiners will go through the preliminary report prepared by the group and observe the presentation prepared by them. Every group member must take part in the presentation. The examiners can ask some questions based on report and presentation to judge the understanding of the topic of every group member and will award the marks according to their performance.

Term II

Teching Scheme	**Examination Scheme**
Practical : 6 Hours/Week	**Term Work : 100 Marks**
	Oral : 50 Marks

1) A project report consisting of a preliminary report prepared in term-I, detailed design (all necessary UML diagrams) document, User Interface design, Test cases and test results generated by available testing tool, conclusions, appendix (if necessary), glossary, tools used and references at the end of term-II.
2) The project work will be assessed by internal examiner (preferably the guide) and an external examiner from the industry having five or more years of experience.
3) The pair of examiners will go through the project report prepared by the group and observe the presentation prepared by them. They will ensure taht every member has contributed in the project work.
4) The pair of examiners will conduct a oral examination based on the project work they have undertaken during

the entire academic year.

410448 Networks And Information Security

Teaching Scheme	**Examination Scheme**
Theory : 4 Hrs/ Week	**Theory:100 Marks**
Practical : 2 Hrs/Week	**Duration : 3 Hrs.**

UNIT I :

Introduction

Need of security, attributes of security, authentication, access control, confidentiality, authorization, integrity, non-reproduction and cryptography, Vulnerabilities in OSI model, layers, Types of attacks, DOS, IP spoofing, man-in-the-middle, attack, replay, DNS poisoning, Information security lifecycle, multilevel model of security, Worms, viruses, Trojans, one time passwords, single sign on, use of Bioinformatics in security.

UNIT II :

Public Key Cryptography

Principles, RSA, ECC, DSA, key management, Kerberos, Elliptical curve Cryptography, X.509, diffie-helleman, key exchange, attacks, message authentication and hash functions, Hash algorithms, digital signatures.

UNIT III :

Secret Key Cryptography

DES, triple DES, AES, IDEA, key distribution, attacks.

UNIT-IV :

Virtual Private Network

Need, types of VPN, VPN supported systems, tunneling and tunneling protocols, PPTP, L2TP, IPSec protocol suite, IKE, ESP, AH.

UNIT V :

Network Periphery Security

Router and security, firewalls, packet filters, DMZ, application level gateways, IPS, types, OS hardening, VLAN, wireless LAN, WEP, Honey pot.

UNIT VI :

Web And Email Security

Security services, web security considerations, SSL and TLS, SET, PEM and S/MIME, PGP, smart cards, application security using smart cards, Kerberos, electronic commerce attacks, micro payments, unsmart cards, E-cache.

Text Books:

1. William Stallings "Cryptography and network security, principles and practices",Pearson

Reference Books:

1. Charlie Kaufman, Radia Perlman and mike speciner "Network security, private communication in a public world"
2. Christopher M. King, Curtis patton and RSA press "Security architecture, design deployment and operations".
3. Stephen northcatt, leny zeltser, et al "INSIDE NETWORK Perimeter Security" Pearson Education Asia.
4. Robert Bragge, Mark Rhodes, Heith straggberg "Network Security - the complete reference", Tata McGraw Hill Publication

410419 Advanced Computer Architecture and Computing

Teaching Scheme	Examination Scheme
Theory : 4 Hrs/Week	Theory : 100 Marks
	Duration : 3 Hours

UNIT I :

Overview of Parallel Processing and Pipelining Processing

Necessity of high performance, Constraints of conventional architecture, Parallelism in uniprocessor system, Evolution of parallel processors, future trends, Architectural Classification, Applications of parallel processing, Instruction level Parallelism and Thread Level Parallelism, Explicitly Parallel Instruction Computing (EPIC) Architecture, Case study of Intel Itanium Processor

Principles of scalable performance : Performance Metrics and Measures, Speedup Performance Laws. **(6)**

UNIT II :

Pipeline Architecture

Principles and implementation of Pipelining, Classification of pipelining processors, General pipelining reservation table, Design aspect of Arithmetic and Instruction pipelining, Pipelining hazards and resolving techniques, Data buffering techniques, Job sequencing and Collision, Advanced pipelining techniques, loop unrolling techniques, out of order execution, software scheduling, trace scheduling, Predicated execution, Speculative loading, Register Stack Engine, Software pipelining, VLIW (Very Long Instruction Word) processor, Case study: Superscalar Architecture- Pentium, Ultra SPARC. **(8)**

UNIT III :

Vector and Array Processor

Basic vector architecture, Issues in Vector Processing, Vector performance modeling, vectorizers and optimizers, Case study: Cray Arch.

SIMD Computer Organization Masking and Data network mechanism, Inter PE Communication, Interconnection networks of SIMD, Static Vs Dynamic network, cube hyper cube and Mesh Interconnection network.

Parallel Algorithms For Array Processors: Matrix Multiplication, Sorting, FFT **(8)**

UNIT IV :

Multiprocessor Architecture

Loosely and Tightly coupled multiprocessors, Processor characteristics of multiprocessors, Inter Processor communication network, Time shared bus, Crossbar switch, Multiport Memory Model, Memory contention and arbitration techniques, Cache coherency and bus snooping, Massively Parallel Processors (MPP), COW's and NOW's Cluster and Network of Work Stations), Chip Multiprocessing (CMP), Case Study of IBM Power4 Processor

Inter Processor Communication and Synchronization (8)

UNIT V :

Multithreaded Architecture

Multithreaded processors, Latency hiding techniques, Principles of multithreading, Issues and solutions.

Parallel Programming Techniques

Message passing program development, Synchronous and asynchronous message passing , Message passing parallel programming, Shared Memory Programming, Data Parallel Programming **(7)**

UNIT VI:

Parallel Software Issues

a) Parallel algorithms for multiprocessors, classification of parallel algorithms, performance of parallel algorithms

b) Operating systems for multiprocessors systems, Message passing libraries for parallel programming interface, PVM (in distributed memory system), Message Passing Interfaces (MPI), PThreds (in shared memory system)

c) Parallel Programming Languages : Fortan 90, Occam, C-Linda, CCC etc.

d) Issues towards cluster computing. Introduction to Neuro Computing and Grid Computing **(8)**

Text Books

1. Kai Hwang, Faye A. Briggs, "Computer Architecture and Parallel Processing" McGrawhill international Edition

2. Kai Hwang, "Advanced Computer Architecture", Tata McGrawhill Edition

References

1. V.Rajaraman, L Sivaram Murthy, "Parallel Computers", PHI.
2. William Stallings, "Computer Organization and Architecture, Designing for performance" Prentice Hall, Sixth edition
3. Kai Hwang, Scalable Parallel Computing
4. Harrold Stone, High performance computer Architecture
5. Richard Y. Kain , Advanced Computer Architecture
6. <http://www.intel.com/products/processor> (for Intel Itanium Processor)
7. For IBM Power 4 Processor
   a. http://www.ibm.com/servers/eserver/pseries/hardware/whitepapers/power/ppc_arch.html
   b. <http://www.ibm.com/servers/eserver/pseries/hardware/whitepapers/power/ppc_arch_2.html>

410450 Software Testing And Quality Assurance

Teaching Scheme :
Theory : 04 Hrs./week

Examination Scheme :
Theory : 100 Marks
Duration: 3 Hrs.

Objectives :

- To introduce Software Measurement concepts
- To introduce Software Testing Process
- To emphasis on Software Testing strategies
- To introduce Software Quality management principles & metrics.

UNIT I :
Principles of Measurement:

Representation Theory of Measurement, Measurement and models, Measurement Scales, Classification of Software Measures, Determining what to measure, Applying Framework, Software Measurement Validation, Four principles of Investigation, Planning Formal Experiments, What is a good data, How to define/collect data, How to Store and Extract data.

UNIT II :
Internal Product attributes Measurement

Size : Aspects of software size, length, reuse, functionality, complexity

Structure : Types of structural measures, control-flow structures, Modularity and information flow attributes, Object-oriented metrics, Data structure, Difficulties with general complexity measures, Halstead's Software Science.

UNIT III :

Software Measurement Programs and Principles of Testing :

What is a metric plan?, Goal-Question-Metric model, Measurement tools, Measurement in small, Measurement in Large systems.

Defects : Origins of Defects, Defect Classes, Defect repository and Test Design, Developer/Tester support for Defect Repository

Test Case Design I [White-Box] : Test Adequacy criteria, Static testing by humans, Static analysis tools, Structural Testing, Code Complexity testing, Mutation Testing

Test Case Design II [Black-Box] : Test case Design Criteria, Requirement based testing, Positive and negative testing, Boundary Value analysis, Equivalence Partitioning, State-based or Graph-based Testing, Compatibility Testing, User Documentation Testing, Domain Testing

UNIT IV :

Software testing :

Test plan, Management, Execution and Reporting, GUI testing, Validation testing, Integration testing, System and Acceptance testing, Scenario testing, Regression testing, Specification-based testing, Performance Testing, Ad hoc Testing, Usability and Accessibility Testing, Software Test Automation.

UNIT V :

Software Quality metrics and tools :

Quality concepts, Software Quality Assurance, Six Sigma principles, Malcolm Baldridge Assessment, ISO 9000, Edward Deming's principles, Total Quality Management, Product Quality Metrics, In process Quality Metrics,

Software maintenance, Ishikawa's 7 basic tools, Checklists, Pareto diagrams, Histogram, Run Charts, Scatter diagrams, Control chart, Cause Effect diagram. Defect Removal Effectiveness & Process Maturity Level.

UNIT VI :

Software Maintenance :

Problem Reporting: Customer side Preliminary activities, Defects reported by Internal Customers, Logistics and Tooling, Challenges and Best Practices.

Problem Resolution: Overview of Problem Resolution, Categorizing and Identifying problem, Making the Fix and Testing it, Challenges and Best Practices.

Fix Distribution: Overview of Fix Distribution, Choosing method of Fix Distribution, Composing Fixes, Preparing and Testing Shipment unit.

Text Books :

1. Fenton, Fleeger, " Software Metrics: A Rigourous and Practical Approach", Thomson, ISBN 981-240-385-X
2. Stephen H. Kan, "Metrics & Models in Software Quality Engineering", Pearson Education, ISBN 81-297-0175-8

Reference Books :

1. Ramesh, Bhattiprolu, " Software Maintenance", Tata McGraw Hill, ISBN 0-07-048345-0
2. Desikan, Ramesh, "Software Testing : Principles and Practices", Pearson Education, ISBN 81-7758-121-X
3. Burnstein, "Practical Software Testing", Springer International Edition, ISBN 81-8128-089-X

410451 Distributed Systems

Teaching scheme :
Theory : 4 Hrs/ Week.
Practical : 2 Hrs/Week

Examination Scheme :
Theory: 100 Marks.
Term work : 25 Marks
Oral : 50 Marks
Duration : 3 Hrs

UNIT I :

Introduction

Introduction to distributed Systems, examples of distributed systems, characteristics, goals, hardware and software concepts, design issues, resource sharing and the web, challenges.

System Models: Introduction, Architectural Model, Fundamental Models, and Client Server Models.

UNIT II :

Communication

Inter process Communication: Message oriented Communication, Stream Oriented Communication
Layered Protocols: Lower Level, Transport Level and Higher-level Protocols.

Distributed Objects: RPC & LRPC, Remote Method Invocation, Events and Notifications.

UNIT III :

Distributed File Systems

Distributed File Systems: SUN NFS, CODA, Other DS, Comparisons.
Name Services: Name Entities, Locating Mobile Entities, Removing unreferenced entities, Case Studies: DNS Directory, Global Name Service, X 500.

UNIT IV :

Synchronization

Time and Global States: Clock Synchronization, Logical clocks, global state.

Co-ordination: Election Algorithms, mutual exclusion, Distributed Transaction.

UNIT V :

Fault Tolerance

Process Resilience, Reliable client server communication, Reliable group communication, Distributed Commit and Recovery.

UNIT VI :

Case Studies

Case Studies on CORBA, Grid and Clusters.

Text Books :

1. Andrew S. Tanenbaum & Maarten van Steen,Distributed Systems, "Principles and Paradigms", Publisher: PHI.
2. George Coulouris, Jean Dollimore & Tim Kindberg, "Distributed Systems - Concepts and Design", Publisher: Pearson (LPE)

Reference Book :

1. Pradeep K. Sinha "Distributed Operating Systems Concepts and Design", Publication: PHI.

Distributed Systems Lab.

Assignments :

1. (a) Implement RPC.
   (b) Implement RMI having following components : Interface, Implementation, stub procedures and client-server.
2. Simulate logical clock synchronization using any method.
3. Write a program to simulate the Election algorithms (both ring and bus topology).
4. Study assignment on NFS : This should emphasize the usage of Pathname resolution and mounting.
5. Implementation of Conflation Algorithm.
6. Implementation of single pass algorithm for clustering.
7. Implementation of Boolean/Search strategy.

Note : Students can implement the algorithms in any programming Language.

410451 Software Architecture

Teaching Scheme :
Theory : 4 Hours /Week

Examination Scheme:
Theory : 100 Marks
Term Work : 25 Marks
Oral : 50 Marks
Duration : 3 Hrs.

Objectives

- Introduction to Software architecture as a discipline
- Introduction to current architecture approaches.
- Introduction to software Architecture strategies.

UNIT I :

Architecture Business Cycle, What is software architecture, why is software architecture important. Documenting software architectures.

UNIT II :

Understanding quality attributes, architecture and quality attributes, achieving quality attributes.

UNIT III :

Design Patterns: history, principles and expectations. Ways of using patterns. Study of a number of representative patterns like Singleton, Factory, Adaptor, Façade, Proxy, Pipes And Filter, Event Listener, Iterator, Observer, Mediator.

UNIT IV :

Types of Middleware, Application servers, Introduction to Java EE, Introduction to Java EE technologies like JMS, JDBC, JAX-WS, JXTA. EJB 3.0 Architecture, Entity, Session, Message beans.

UNIT V :

Introduction to three tier and N-Tier Web Architectures, XML, Client side technologies DHTML, Java Applets, Active X controls, Server side technologies JSP, JSF, Java Servlets.

UNIT VI :

Components, Interfaces, DLL Servers, Introduction to .NET architecture, .NET assemblies, .NET Remoting,. NET Web Services

Reference Books :

1. Len Bass, Paul Clements <http://www.amazon.com/exec/obidos/search-handle-url/index=books&field-author-exact=Paul%20Clements&rank=-relevance%2C%2Bavailability%2C-daterank/104-7094751-4143148>, Rick Kazman, "Software Architecture in Practice", Second Edition ,Hardcover
2. Eric J. Braude, "Software Design: From Programming to Architecture" ,ISBN: 0-471-20459-5,©2004
3. Dale Rogerson , "Inside Com (Microsoft Programming Series) ", Paperback
4. James L. Weaver, Kevin Mukhar, "Beginning J2EE 1.4: From Novice to Professional (Apress Beginner Series) (Paperback) ", James P. Crume (Publisher)
5. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides ,"Design Patterns: Elements of Reusable Object-Oriented Software" (Addison-Wesley Professional Computing Series) (Hardcover)

Software Architecture Lab

Objectives :

- Understand various non-functional requirements that lead to need for good Architecture and good Design.
- Explore various design patterns and learn to implement them
- Explore some representative architectural styles
- Understand how to design and partly implement Client Server systems using components and web services

- Implement and study some MIDDLEWARE based clients and components in LAN and WEB world.
- Explore implementations of some of the underlying technologies of distributed client server applications

Faculty can set assignments based on Microsoft AND / OR Java world AND /OR Open source based languages, platforms, Middleware, APIs.

Assignments

Part A: Design Patterns

1. Study any Four patterns and SUBMIT a design pattern specification for two patterns in a standard format along with their UML diagrams. The specification must include
   Problem/ Issue, Audience/ Context, Forces, Solution. Discussion/ Consequences/ Implementation, Related Patterns, Example Instances, References.
2. Implement an ITERATOR or OBSERVER pattern in language of your choice and submit it along with a write-up with its specification.
3. Implement any one other pattern in language of your choice and submit it along with a write-up with its specification.

Part B : Architectural

1. Study and submit a report for any of the MVC based Frameworks
2. Implement the pipes and filters architectural style
3. Implement Keyword in context KWIC problem using any one architectyral style

4. Case study of any website or any other large system and its architecture for fault tolerance, scalability, performance, response time, transaction management and other quality attributes

PART C : Middleware and Web services

1. Implement a representative paper design of a hypothetical system using components, Interfaces, its deployment issues with UML 2.0
2. Submit a IDL definition for a few components and interfaces from above system
3. Implement a client and set of components in C++ and COM as DLL servers

   OR

   Implement a client and set of components in C++ and COM as EXE servers
4. Explore and Implement one of the Advanced JEE technology

   Java RMI OR JAVA SOCKETS (with concurrency and threads) OR JAVA based XML processing
5. Implement a sample EJB based application.

PART D : Web Environment

1. Develop dynamic and interactive web clients with use for some of the technologies listed here DHTML, XML, Scripting, Applets or Active X controls, Client side caching.
2. Do server side programming using either CGI scripts or ASP/JSP with database handling and components on server side with security aspects covered.

410451 Embedded System

Teaching Scheme :
Theory : 4 Hours /Week
Practical : 2 Hrs / Week

Examination Scheme :
Theory: 100 Marks
Term Work : 25 Mark
Oral : 50 Mark
Duration : 3 Hrs.

UNIT I :

Introduction to Embedded System

Components of Embedded System & it's Classification, Characteristic of embedded system . Review of Microprocessors & Microcontrollers. Introduction to embedded processor, Digital Signal Processor, Application Specific System Processor, Multiprocessor systems using General Purpose Processor . CISC and RISC Processor architectures and exemplary instruction set, Exemplary ARM Processor.

UNIT II:

System Hardware:

Management of Power Supply , Clocking Unit, Real Time Clock and Timers, Reset Circuitry and Watchdog Timer. Structural Units of Processor. Processor and Memory Selection, Memory Map Of Embedded System DMA , Interfacing Processors , Memories and I/O. Example architecture of Mototrala-68HC/08, AVR AT9052313,Evaluation Board Concept.

UNIT III :

I/O interfacing :

I/O devices, ADC / DAC , Optical Devices such as LED / LCD Display devices ,Opto Isolator , Relay & stepper motor , Timers & counting devices, serial communication using I2 C, CAN , RS232, & USB Device drivers & interrupt servicing inter process communication & synchronization Multiple Processes, Multiple Task, threads & routines , Os & Embedded System model

UNIT IV :

Programming concepts, Embedded System Programming C & C++ :

Software development cycle, Assemble lang., High Level Lang. C program Elements, Micros & Function, Data types, Data structures, Modifiers, Statements, Loops & Pointers, Queues & Stacks, List & Order List, Embedded System Programming In C++ & Java. C Program Compliers & Cross Compliers, In circuit emulator.

UNIT V :

Real Time Operating Systems

Real Time & embedded System Os, Interrupt Routines in RTOS environment, RTOS Task Scheduling models, Interrupt Latency & Response time, Strategy for synchronization between the processes , ISR, OS functions & tasks for resource management Embedded Linux, internals : Linux Kernel for device drivers & embedded system , OS security issue , Mobile OS. RTOS

UNIT VI :

Programming Tools and Applications of Embedded Systems :

MicroC/OS-II, VxWorks .

Case Study of coding for Vending machine system using MUCOSRTOS , Case study coding for send application layer byte streams on A TCP/IP Network Using RTOS Vx works , Case study of an embedded System for an adapting Cruise control System in a car, Case study in Embedded system for Smart Card.

Text Books

1. Rajkamal, "Embedded System Architecture Programming Design" Tata Graw Hill Publication
2. Dr. K. V. K. K. Prasad "Embedded / real time System : Concepts, Design, & Programming - Black Book" Dreamtech Press Publication
3. Dr. K. V. K. K. Prasad, Gupta Dass, Verma "Programming for Embedded system " Wiley - Dreamtech India Pvt. Ltd.

Reference Book :

1. Raj Kamal , "Microcontrollers , Architecture, Programming , Interface & System Design" Pearson Education
2. Sriram Iyer , Pankaj Gupta,"Embedded Real time Systems Programming", Tata Mc Graw Hill.
3. Tammy Nergaard "Embedded Systems Architecture - A Comprehensive Guide For Engineering & Programming", Elesevier Publication

Embedded System Lab

Perform Any 10 Experiments from the following list :

1. Study of Evaluation Board - Controlling Hardware & Software
2. Develop 8 Channel Data Acquisition System to Acquire Date from 8 Channel , Convert it into Digital Format & transmit to PC .
3. Write a C Program Which Generates Packets of 32 Bits . First bit of packet indicates whether the packet is control packet or data packet
4. Using Microcontroller development Board Develop the software which read input from the switches
5. Using Microcontroller development Board Develop the software for serial communication
6. Write a program to control the relay which can be used in Process control system
7. Develop an Embedded system that takes Analog voice signal as input & converts it into digital format & send this data to PC.
8. Write a Shell Script that display the no. of readable , writable & executable files in specified Directory.
9. Write a C Program that takes string input from keyboard & Displays the Length of string - use Multi Threading for message Queue or Shared Memory
10. Write a program that demonstrates the communication between two Processes. One process running as Linux Process & other as RTL Linux Process
11. Write a program that demonstrates multithreading in RTL Linux.
12. Write a program to control an Application RTL Linux.

410451 High Performance Networks

Teaching Scheme :
Theory : 4 Hrs/ week
Practical : 2 Hrs/week
Marks

Examination Scheme
Theory : 100 Marks
Term Work : 25
Oral : 50 Marks
Duration : 3 Hrs.

Prerequisite :

- Conceptual knowledge of Digital transmission and line coding
- Communication networks (Circuit switching and packet Switching)
- Conceptual overview of X.25
- Concepts of OSI model and Local area Network

UNIT I :

Gigabit Ethernet:

Business Drivers and need of it,Architecture and Overview of Gigabit Ethernet,Gigabit Ethernet Media Access Control (Half Duplex operation Vs Full Duplex Operation),Gigabit Ethernet Physical layer (1000Base X and 1000Base T),Applications of Gigabit Ethernet, Ethernet summery Migration from 10Mbps to Gigabit Ethernet,Network Design using Gigabit Ethernet a case study

UNIT II :

Integrated Services Digital Network:

Conceptual view of ISDN and ISDN standards, ISDN Interfaces and functions (transmission structure, U-N Configuration, Protocol architecture etc),ISDN Data Link layer (LAPD protocol, terminal adoption, I.465/v.120),ISDN Network layer (Overview, basic call control using Q.931) ISDN services, Conceptual overview of Signaling System Number 7.

Frame Relay :

Frame relay Vs X.25,Frame relay Protocols and services (protocol architecture, Frame Mode call control),LAPF protocol (Both Core and Control),Concept of DLCI and its significance, Frame Relay Congestion control, Need, Congestion Control frame Work, Network use of CIR, and DE bit, Congestion Notification (FECN,BECN and CLLM)

UNIT III :

Broadband ISDN (B-ISDN) :

Driving forces and need, B-ISDN standards and services, B-ISDN Functional Architecture, B-ISDN Transmission structure, B-ISDN protocol architecture, SONET/SDH and comparison with other available standards

ATM :

Overview, ATM protocol architecture, Virtual Channels and Virtual Path Switching, Detail Functionality of ATM Layer(Cell structure , HEC, Cell Delineation etc),ATM Adoption layer (need, different types and comparison),ATM traffic and Congestion control ,Requirements, ATM service categories ,ATM traffic descriptors, ATM QOS parameters, Classical IP over ATM,ATM in LAN environment (LANE)

UNIT IV :

ADSL and DSL Technologies :

Background and technological capabilities, Standards and associations, Architecture, Conceptual overview of VDSL, Deployment Case study, Market status and future

UNIT V :

MPLS & RSVP:

MPLS,RSVP, Integrated & differential Services

UNIT VI :

WiFi & WiMax

Text Books :

1. William Stallings, "ISDN and Broadband ISDN with Frame Relay and ATM" 4th edition (Pearson Education)
2. Rich Seifert ,"Gigabit Ethernet"(Addison Wesley Inc.)

Reference Books :

1. Sumit kasera and Pankaj Sethi ,"ATM Networks Concepts and protocols" Tata McGraw Hill Publication.

High Performance Networks Lab

Laboratory Assignments :

1. Case studies of implementation of Gigabit Ethernet including the components , costings etc.
2. Implementation of ISDN (U-N interface, Router configuration, Costing etc.)
3. Frame relay service availability - A user's perspective
4. Comparative study of different technologies
5. Industrial visit to understand the market scenario (typically ISP site would help)

410452 Computer Laboratory -II

Teaching Scheme :
Practical :4 Hrs/Week

Examination Scheme :
Term Work : 25 Marks
Practical : 50 Marks

PART I : Network And Information Security

1. Write a program to find an IP address of a remote system.
2. Write a program to detect a Remote Firewall.
3. Install a Proxy server and configure an application Gateway.
4. Install, Configure and study a Intrusion detection system (IDS).
5. Perform a risk assessment of your personal computer (Assume that you have been using your computer to write your throughput: You'll need to identify assets, vulnerabilities, likelihood of incidents, possible counter measures/controls, and a disaster plan.)
6. Implementation of MD5 hashing technique.
7. Implementation of Rijndael algorithm using 64-bit key.
8. Implementation of Diffie-hellman algorithm.
9. Implementation of email security using PGP/GPG(create yourself a 1024-bit gpg key. Use your name and email address for your key label. Use gpg to verify the signature on this assignment.)

10. Design an experiment to estimate the amount of time to
    - Generate key pair (RSA)
    - Encrypt n bit message (RSA)
    - Decrypt n bit message (RSA)

    As function of key size. Experiment with different n-bit messages. Summarize your conclusion.
11. The first in generating encrypt/decrypt keys in RSA is to generate two prime numbers p and q. Now suppose p is not a prime but rather the product of two primes p1 and p2.
    - Will this pose a problem in any way?
    - If so, how/why and how often?

PART II : Software Testing & Quality Assurance

Objectives:

- To introduce Software Testing Process
- To emphasis on Software Testing strategies

Assignment 1 : Prepare System Requirement Specification, Use Case Diagrams, Sequence Diagrams, State chart Diagrams for target system (System Under Construction) demonstrating Static and Dynamic behavior of the system. Prepare Class Diagram demonstrating System Design. Narrate necessary supporting documentation.

Assignment 2 : Implement the System under construction and narrate Test Plan for the same. Identify Test cases, Test Procedures, Test log, Test Oracle for the System under test. The Test case Scenarios should correspond to the relevant Use case Scenario.

Assignment 3 : Perform Unit testing especially indicating the traced Independent data paths, Control paths and Error handling paths. Prepare control flow graphs for the unit under test. Compute the Cyclomatic complexity of the unit. Record the Test criteria and Test conditions along with Test results.

Assignment 4 : Perform Black box Testing for all the units contained in the architectural segments. Perform Regression Testing / GUI Testing of the System under construction.

Assignment 5 : Prepare System Usage Specification outline. Specify Usage patterns of the system and indicate it using Run charts / Histograms.

Note: The instructor is required to frame five assignments covering all the details as mentioned above at suitable length with relevant use of case tools.

P.U.P.—500+5–11–2006 (382Compu) [PC-1]