Price:- 10Rs.

॥ अंतरी पेटवू ज्ञानज्योत ॥

North Maharashtra University, Jalgaon

Syllabus for Forth Year Engineering Degree Course (B.E.)

COMPUTER

w.e.f. July,2001

North Maharashtra University, Jalgaon

B.E. (Computer Engineering) (1998 Course)

Term I

Sr. No	Subject Code	Subject	Teaching Scheme Hours/Week		Examination Scheme				
			Lectures	Practical	Paper Duration Hours	Maximum Marks			
						Paper	TW	PR	OR
1		Elective - I	4	2	3	100	25	-	25
2		Operating Systems	4	2	3	100	25	-	-
3		Data Base Management Systems	4	2	3	100	25	25	-
4		Software Engineering	4	2	3	100	25	25	-
5		Seminar	-	2	-	-	50	-	-
6		Project Work	-	4	-	-	50	-	-
		Total	16	14	-	400	200	50	25
		Grand Total	30		-	675			

Elective - I

1. Object Oriented Modeling and Design.
2. Artificial Intelligence.
3. VLSI Design.

* Each Unit carries 20 Marks.

Term II

Sr. No	Subject Code	Subject	Teaching Scheme Hours/Week		Examination Scheme				
			Lectures	Practical	Paper Duration Hours	Maximum Marks			
						Paper	TW	PR	OR
1		Elective - II	4	2	3	100	25	-	25
2		Computer Networks	4	2	3	100	25	25	-
3		Computer Graphics	4	2	3	100	25	25	-
4		Advanced Microprocessors	4	2	3	100	25	-	-
5		Technical Visit	-	-	-	-	50	-	-
6		Project Work	-	6	-	-	50	-	50
		Total	16	14	-	400	200	50	75
		Grand Total	30		-	725			

Elective - II

1. Component Architectures.
2. Image Processing.
3. Compiler Construction.
4. Advanced Computer Architecture

* Each Unit carries 20 Marks.

Object Oriented Modeling and Design (Elective I)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 Marks

Unit 1:
Review of Object Modeling, New paradigm, object oriented thinking- rethinking, Objects and Classes, Links and association, generalization and Inheritance, Grouping concepts, aggregation, abstract classes, Polymorphism, Metadata, Constraints, Reuse, dynamic modeling, events, states, Operations, Concurrency.

Unit 2:
Importance of modeling, brief overview of Object Modeling Technology (OMT) by Rambaugh, Booch Methodology, USE CASE driven approach (OOSE) by Jackobson, overview of CRC card method by Cunnignhaum.

Unit 3:
Overview of UML: Efforts of standardization / Integration, OMG approval for UML, Scope of UML, Conceptual model of UML, architecture- Metamodel, mechanisms, Unified Software development life cycle, UML diagrams (All). Class diagrams: Terms and Concepts, Relationship diagrams.

Unit 4:
Terms and Concepts of:

- Advanced Class Diagram: Advanced relationship, interface- types and rules, packages, common modeling techniques, modeling groups of elements, modeling architectural views
- Instances and Object diagrams: Modeling concrete/ prototypical instances, links, object interaction
- Collaborations, Use Cases, Interaction diagrams, State Transition Diagrams
- Architectural Modeling: Component Diagram, Deployment Diagram, Pattern and Frame work

Unit 5;
Introduction to Component Technology, Concept of distributed Object Systems; COM and CORBA, Introdution to Object Oriented Database

References:

1. Booch/Rambaugh, Jacobson - UML User Guide, Addison Wesley.
2. Sinon Alhair: UML in a Nutshell.
3. Rambaugh: Object Oriented Modeling and Design, PHI.
4. Booch; Object Oriented Analysis and Design with applications, Addison Wesley.
5. B. Meyer: Object Oriented Software Constructions, PHI.

Web Sites

http://www.rationalrose.com
http://www.citus_links.com

Laboratory Assignments:
Consider representative system such as College Library, Inventory, super Shoppe (or any other topic of sufficient complexity). For this system define as assignment

1. Class Diagram
2. Use Cases
3. Interaction or Activity Diagram
4. State Chart diagram

Using Rational Rose / Object Analyst, or any other tool. (Minimum 4 assignments).

Artificial Intelligence (Elective I)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 marks

Artificial Intelligence (Elective I)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 marks

Unit 1:
Introduction to Artificial Intelligence:
Definition, AI Applications, AI Representation, Properties of internal Representation, Heuristic search techniques, Best files search, Mean and ends analysis, A* and AO* Algorithm
Game Playing:
Minimize search procedure, Alpha-beta cutoffs, Warting for Quiscence, Secondary search.

Unit 2:
Knowledge Representation using Predicate Logic:
Predicate calculus, Predicates and Arguments, ISA hierarchy, Frame notation, Resolution, Natural deduction
Knowledge Representation using Non-monotonic Logic:
TMS (Truth Maintenance System), Statistical and probabilistic reasoning, Fuzzy Logic, Structure knowledge representation, Semantic Net, Frames, Script, Conceptual dependency.

Unit 3:
Planning:
Types of planning, Block world, strips, Implementation using goal stack, Nonlinear planning with goal stacks, Hierarchical planning, List commitment strategy.
Perception:
Action, Robot architecture, Vision, Texture and images, Representing and recognizing scenes, Walzs algorithm, Constraint determination, Trihedral and Nontrihedral figures labeling.

Unit 4:
Learning:
By training neural networks, Introduction to neural networks, Neural net architecture and applications. Natural Language Processing and understanding, Pragmatic, Syntactic, and Semantic analysis, Finite State Machine, RTN, ATN, Understanding sentences.

Unit 5:
Expert System:
Utilization and functionality, architecture of Expert system, Knowledge representation, Two case studies on expert systems

References:

1. Eugene Charniak, Drew McDermott, "Introduction to Artificial Intelligence".
2. Elaine Rich, Kevin Knight, " Artificial Intelligence".
3. Kishan Mehrotra, Sanjay Rawika, and K. Mohan, " Artificial Neural Network".

Laboratory Assignments:
Assignments based on:
1. A* Algorithm Implementation
2. AO* Algorithm Implementation.
3. Implementation of Unification Algorithm.
4. Hill Climbing Algorithm.
5. Game playing with Min/Max Search.
6. Implementation of Dynamic database.
7. Parsing method implementation.
8. Development of Mini Expert System using Prolog
9. Application development using Neural Network.
10. Development of Intelligent Perception System.

Any six assignments should be framed by concerned staff.

VLSI Design (Elective I)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 marks

Unit 1:

VLSI Design (Elective I)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 marks

Unit 1:
Review of MOS, CMOS logic, Logic gates: NAND, NOR, XOR, Latches and Buffers; DC analysis of inverter / basic gates, Concept of minimum logic levels, noise margin and delay calculations, History of IC design, What is VLSI? Introduction to ASIC, Different types of ASIC, Features of different EDA tools, EPLDs, CPLDs, FPGAs, structural details.

Unit 2:
Overview of design automation approach to digital design - use of Hardware Description languages, Introduction to VHDL and Verilog, Digital System Simulation and Hardware synthesis.Structure of VHDL, Timing and concurrency issues, structural specification of hardware, wiring and component interconnections.

Unit 3:
Definition and Usage of packages and components, Design of a general purpose Test Bench.
Use of Design libraries and library management, Introduction to Library STD_Logic 1164 and multivalued logic, Behavioral descriptions of hardware syntax and Semantics for various forms and constructs.

Unit 4:
Synthesis and design implementation with case studies using download facilities at minimum 4 MHz into CPLD series 5200 0r 4000 series, Architecture of FPGA and CPLD such as Xilinx 9500 series CPLDs and 4000 or Spatorz series FPGAs.

Unit 5:
Information on a complete tool from design entry to place and route, Introduction to various standard tools used for simulation and synthesis.

References:

1. Morris Mano, "Digital Design", 2nd Ed., EEE-PHI.
2. Douglas Perry, "VHDL", 3rd Ed., McGraw Hill.
3. Peter Ashender, "VHDL Cook Book".
4. J. Bhaskar, "VHDL".
5. Neil and Kamran, "Principles of CMOS VLSI design", Addison Wesley.
6. Lal, "Digital System Design using PAL".
7. Blackcalac, "Digital Circuits and Systems".
8. Xilinx Manual.

Laboratory Assignments:
Software: Using Xilinx version 2.1 or similar.
Hardware: Using FPGA and CPLD.

a. Experiments based on Combinational Logic

1. Simulation and Implementation of BCD to 7-segment display decoder.
2. Simulation and Implementation of Magnitude Comparators.
3. Simulation and Implementation of 2-line to 4-line Decoder or Multiplexer and Demultiplexer.
4. Simulation and Implementation of ALU with minimum 4 Arithmetic / Logical operations e.g. IC 74181.

b. Experiments based on Combinational Logic (Any 2)

1. Simulation and Implementation of Latches and Registers with Reset and Clear.
2. Simulation and Implementation of Counters.
3. Simulation and Implementation of Shift Registers.

c. Experiments based on Combinational Logic (Any 2)

1. Simulation and Implementation of functionality of 8253.
2. Simulation and Implementation of functionality of 8255.
3. Simulation and Implementation of a Bit programmable Input / Output e.g. 6821.

Operating Systems

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks

Unit 1

Introduction:

Need of OS, evolution of OS, types of OS like Batch, Time sharing, Multiprogramming, Multitasking, Distributed and Real time.

OS views and concepts, Shell command language, System calls, User view, Functional requirement and structure, Monolithic, Layered, Vertical model

Processes and Processor Management:

Process concept, Interleaved I/O, CPU burst, Process state, OS services for process management, Threading.

Unit 2

Scheduling:

Process Scheduling - long term, middle term and short term scheduler, Scheduling Algorithms and performance evaluation. Inter-process communication and synchronization needs, Mutual exclusion, Semaphores. Critical regions – Monitors, messages for inter-process communication and synchronization, Classical problems in concurrent programming.

Dead lock - Principle, detection, prevention, avoidance and recovery, Banker's Algorithm.

Unit 3:

Process Management in UNIX:

Structure of process, process control, process system calls (No algorithms) - Fork, Join, Exec, System boot and Init process, Shutdown process.

Memory Management:

Types, contiguous and Non-contiguous. Segmentation, Paging concepts. Virtual memory and its management (Allocation, Fetch, Replacement).

Unit 4:

Memory Management in UNIX.

Policies, Swapping and Demand paging.

File Management:

Organization, Concepts, Files and Directories, Hierarchical structure of files and space allocation, File space management, Security issues, Protection mechanism.

File Management in UNIX:

Internal representation of files, Inodes.

Unit 5:

File structures in UNIX:

Structure of various files, directories, superblock, inode assignment to a new file, Allocation of disk blocks, System calls, file creation, pipes.

I/O Management:

I/O problem, I/O hardware, I/O interfaces, Buffer register, Buffer commands and design issues.

Distributed OS:

Fundamental concepts, system modules, Issues in designing Distributed OS in brief.

References:

1. Peterson, Operating System Concepts,
2. Milenkovic, Operating System Concepts and Design, McGraw Hills
3. Bach M. J., The Design of UNIX Operating System, PHI
4. Godbole, Operating Systems,
5. Sinha P.K., Distributed Operating Systems Concepts and Design, PHI
6. Steven, Advanced UNIX Programming
7. Donovan John, Operating Systems, TMH
8. Tannebaum, Operating Systems

Web sites

http://www.linux.com

Laboratory Assignments:

1. Study of various commands in UNIX/Linux.
2. Command Interpreter.
3. CPU Scheduling.
4. Memory Management.
5. Deadlock: Banker's Algorithm.

6. Interprocess Communication using Message Queue.
7. Installation of Linux: Workstation as well as Server.
8. Linux System Administration.
9. Web Server Configuration.
10. Mail Server Configuration.

Any 8 assignments on above topics. (Assignment no. 7 compulsory).

Database Management Systems

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Practical: 25 Marks

Unit 1:
Introduction to DBMS:

Basic concepts, Advantages of a DBMS over file processing systems, Data abstraction, Data models and Data independence, Components of a DBMS and overall structure of a DBMS, Data base terminology.

Database Administration Issues: DBA role, Indexes, Data dictionary, Security, Backups, Replication, SQL support for DBA, Commercial RDBMS selection.

Data Modeling

Basic Concepts, Types of data models, E-R data model and Object-Oriented data model, Relational, Network and Hierarchical data models and their comparison, E-R and EER diagramming.

Unit 2:
Relational Model:

Basic concepts, Attributes and Domains, Interaction and extensions of a relation, Concept of integrity and referential constraints, Relational query languages (Relational algebra and relational calculus), Concepts of View and triggers.

Unit 3:
SQL:

Structure of a SQL query, DDL and DML, SQL queries, Set operations, Predicates and Joins, Set membership, Tuple variables, Set comparison, Ordering of Tuples, Aggregate functions, nested queries, Database modification using SQL, Dynamic and embedded SQL and concept of stored procedures, Query optimization.

Unit 4:
Relational Database Design:

Notion of a normalized relations, Normalization using Functional Dependency, Multi-valued dependency and Join dependency, 1NF, 2NF, 3NF, BCNF, 4NF.

Transaction Management:

Basic concept of a transaction, Components of transaction management (Concurrency control and Recovery system), Different concurrent control protocols such as time stamps and locking, Different Recovery system), Different concurrent control protocols such as time stamps and locking, Different crash recovery methods such as log based recovery and shadow paging, Concepts of cascaded aborts, Multi-version concurrency control methods.

Unit 5:
Object Oriented DBMS:

Review of Object Oriented concepts: Objects, Classes, attributes, messages, inheritance and polymorphism etc., Object Schemes, Class-Subclass relationships, interobject relationship, Features of Object-Oriented DBMS and ORDBMS, Concept of OID, Persistence of objects in OODBMS, Physical organization, Object-Oriented queries, Schema modifications, Temporal Databases, Active Databases.

Database Systems Architecture:

Centralized, Client Server System, Parallel systems, Distributed Systems, Web enabled system.

New Applications:

Need for data analysis, Decision support systems, Data warehouse, On-line Analytical processing (OLAP), Data mining concepts, Spatial and Geographic databases, Multimedia databases.

References:

1. Henry F. Korth, Abraham Silberschatz, "Data base System Concept", Third Ed., Mc-Graw Hill Inc., New York.

2. J. Date, "Introduction to Database Management Systems", Sixth Ed, Narosa Publishing House.
3. Groff James R., Paul Weinberg, "LAN times guide to SQL".
4. Bipin Desai, "Introduction to Database Management Systems", Galgotia.

Laboratory Assignments:

1. Creating a sample Database application using conventional file processing mechanisms and 'C' language. The program should provide facilities for retrieving, adding, deleting and modifying records.
2. Prepare an E-R diagram for the given problem definition. Prepare and verify a relational database design using concepts of normalization techniques in appropriate normal form.
3. Creating a sample database file and indexes (for the design made in experiment no. 2) using any client server RDBMS (ORACLE/Sybase) package using SQL DDL, queries. This will include constraints (key referential etc.) to be used while creating tables.
4. SQL DML queries: Use of SQL DML queries to retrieve, insert, delete and update the data base created in experiment no. 3. The queries should involve all SQL features such as aggregate functions, group by, having, order by, sub queries and various SQL operators.
5. Screen Design and Report Generation: Sample forms and report should be generated using Developer 2000 (in case of ORACLE) or through Power – Builder or Visual Basic Front end tools or any prototyping software engineering tool.
6. Case Study of MIS.
7. Prototype of OODBMS/Active Database/Temporal Database in C++.

Software Engineering

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Practical: 25 Marks

Unit 1:

Software and Software Engineering:

What is and Why Software Engineering? Product: evolving role of software, Software characteristics, components, applications, Software crisis and myths, Software engg. Processes, Software development phases and software process models, Prototyping and RAD models, Waterfall, incremental model, spiral model, 4GT model, CASE tools and classifications

Unit 2:

Planning and Management of Software Projects:

(Only basic concepts)

People, Problem and Process, measures, metrics and indicators, metrics for software quality, scoping, Software project estimation, make-buy decisions, software acquisition, Software risks: identification, projection, assessment, monitoring, project scheduling and tracking tasks/work breakdown structures, timeline chart, project plan, CASE tools.

Systems Engineering:

Computer based systems, system engineering hierarchy.

Information Engineering. Information strategy, planning, enterprise modeling, business area analysis, information flow modeling. Product engineering: system analysis, feasibility study, economic and technical feasibility analysis, modeling system architecture diagrams, CASE tools.

Unit 3:

Requirement Analysis:

Communication techniques, FAST, Quality deployment, analysis principles, modeling, partitioning, prototyping, specifications, SRS and SRS reviews analysis models: data modeling, functional modeling and information flow, Data flow diagrams, extensions to real-time systems, behavioral models, mechanisms of structured analysis, ER diagrams, control modeling, data dictionary, CASE tools.

Unit 4:

Design Fundamental:

Software design and software design process, principles and concepts, abstraction, refinement and modularity, software architecture, control hierarchy, partitioning, data structure, information hiding, effective modular design, cohesion, coupling, design module, design document, CASE tools.

Design Methods:

Architectural design and design process, transform and transaction flow, design steps, interface design, procedural design, graphical and tabular design notations.

Software Testing and Testing Strategies:

Software testing fundamentals, test case design, White-box, Black-box testing, control structure testing, strategic approach to testing, strategic issues, unit testing, integrated testing, validation testing, system testing, CASE tools.

Unit 5:

Object Oriented (OO) Software Engineering:

Planning: OO paradigms and concepts, identifying elements of object model. Object oriented analysis (OOA) and OOD: Conventional Vs. OO, generic components of OO Analysis model, OOA process, Object- relationship model, Object-behavior model. Human Computer interface (HCI) components, Object design process, design patterns, CASE tools.

Unified Modeling Language (UML):

Different Methods: Rambaugh/Booch/Jackobsons, need for standardization, Diagramming in UML (Use Case, Class, Interaction, State diagrams), CASE tools.

Software Quality Assurance:

Software quality concepts, Software quality assurance (SQA) and approaches, Software Reliability, SQA plan, ISO 9000 and SEI standards for software, Software configuration management (SCM), base lines, Scan process, Version control, change control, SCM audits, CASE tools.

References:

1. Roger Pressman: "Software Engineering, A practitioner's Approach", 5th Ed., TMH.
2. Sommerville: "Software Engineering", TMH.
3. Pankaj Jalota: "Software Engineering", Narosa Publishing House.
4. Martin Fowler: "UML Distilled", Addison Wesley.
5. Rambaugh: "OO modeling and design", PHI.
6. Pfleeger S. L.: "Software Engineering".

Web sites

http://www.rationalrose.com
Visit the Carnige Mellan University web site

Laboratory Assignments:

For a given problem definition, perform Object-Oriented analysis and Design activities. Make use of Object-oriented analysis and design tools (such as Rational Rose, Object Analyst, etc.) and provide an object-oriented design. (Minimum 4 assignments.)

Seminar

Teaching Scheme:
Practical: 2 Hrs/Week

Examination Scheme:
Term work: 50 Marks

Each student will select a topic in the area of Computer Engineering and Technology preferably keeping track with recent technological trends and development. The topic must be selected in consultation with the institute guide.

Each student will make a seminar presentation in the term making use of audio/visual aids for a duration of 15-20 minutes and submit the seminar report in the form of a bound journal (1 copy) duly signed by the guide and Head of Department. Attendance at seminars for all students is compulsory.

The seminar will be assessed internally by a panel of staff members from the institute during the presentation.

Project Work

Teaching Scheme:
Practical: 4 Hrs/Week (Term I)
6 Hrs/Week (Term II)

Examination Scheme:
Term work: 50 Marks (Term I)
50 Marks (Term II)
Oral: 50 Marks
(At the end of Term II)

The student will undertake one project over the academic year, which will involve the design of a system or subsystem in the area of Computer Engineering.

1. If the project chosen is a hardware project, it will involve designing a system / subsystem or upgrading an existing system. The design must be implemented into a working model with necessary software interfacing and a user manual.

OR

2. If the project chosen is in the area of pure Software Application, it must involve the detailed Software Design Specifications, Data Structure Layout, File Design, Testing with complete documentation and user interface, with life cycle testing and as an executable package.

The project will be undertaken preferably by a group of at least 4 students (Maximum 5 students) who will jointly work and implement the project. The group will select a project with the approval of the guide and submit the name of the project with the synopsis of not more than 2 to 3 pages not after than the 2nd week of August in the academic year. A preliminary report by the group must be submitted and certified at the end of Semester I.

The group will submit at the end of Semester II.

1. The workable project.
2. Project Report in the form of bound journal complete in all respects - 1 copy for the institute and 1 copy for each student in the group for certification.

The term work will be assessed by the examiners in consultation with the guide. Oral examination will be based on the project work completed by the candidate.

The project report will contain the following details:

1. Problem definition and requirements specification, acceptance test procedure (ATP).
2. System definition - requirement analysis.
3. System design.
4. System implementation - code documentation, data flow diagram / algorithm.
5. Test results and procedure - test report as per ATP.
6. Platform choice, use.
7. Appendix - tools used, references.

Documentation will use UML approach with Presentation, Category, Use Case, Class diagrams, etc.

Component Architectures (Elective II)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 Marks

Unit 1:
Component Architecture: Introduction
Component Object Model (COM) Introduction:
COM as better C++, Software distribution, Dynamic Linking, Separating Interface from implementation, Run time polymorphism, Introduction to DCOM.
Interfaces in COM/DCOM:
Introduction to interfaces, Interface Definition Language (IDL), Interfaces and IDL, Using COM interface pointers, Optimizing query interface, Code sharing and reuse.

Unit 2:
Classes and Objects in COM/DCOM:
Introduction, Classes and Servers, Optimizations, Classes and IDL, Class Emulation, Query interface types and properties, Object services and Dynamic Composition, Apartments; Cross apartments access, Life cycle management.

Unit 3:
Distributed COM:
Fundamental programming architecture of DCOM; Parallel processing, Advantages of distributed computing.
Threading Models and Apartments: Apartments, Apartments interaction, Implementing multithreaded local components.
Facilities: Connection points and type information, connectable objects.
Remoting: DLL surrogates and Executable components.

Unit 4:
CORBA:
Objectives, History of CORBA, OMG and CORBA, Overview of CORBA, CORBA components, Architectural features, Method Invocations: Static and Dynamic, ORB.
CORBA Services:
Services for object naming, object life cycle, Event, Transaction service features, concurrency control services, Persistent Object service and CORBA security service.
CORBA Facilities, CORBA Domains

Unit 5:
Enterprise Java Beans:

Introduction to distributed Computing: Mainframe-Transaction Monitors, Two-tier and Three-tier architecture, EJB's role in Middle-tier.

Java Beans and Enterprise Java Beans, Architecture of EJB, Session and Entity Java beans

Comparison of EJB, MTS and CORBA.

HTML, XML & XSL: Introduction, Limitations of HTML.

Java RMI, JDBC: Introduction only.

ASP: Introduction only.

References:

1. Dale Rogerson, "Inside COM", Microsoft Press.
2. Guy Eden, Henry Eden, "DCOM", Microsoft Press.
3. Mowbrey, "Inside CORBA", Addison Wesley.
4. Mowbrey, Zahavi, "Essential CORBA", Addison Wesley.
5. John Siegle, "CORBA Fundamentals and Programming", John Wiley and Sons.
6. Orfali, "The essential Distributed Object Survival Guide".
7. Valesky, "Enterprise Java Beans", Addison Wesley.
8. Bruce Eckel, "Thinking in Java".
9. Naughton, Schildt, "The Complete Reference Java 2", TMH.
10. "Mastering ASP", BPB.

Web sites

http://www.msdn.microsoft.com
http://java.sun.com
http://www.vcdj.com

Laboratory Assignments:

1. Build a COM Component in VC++ and client in VB.
2. Same as 1 only client in VJ++.
3. Same as 1 only component is build in VB and client in VC++
4. Using Java socket programming APIs, write a simple echo server.
5. Assignment on Java Applet.
6. Accessing VB component using ASP (Active Server Pages).
7. Accessing a Database using JDBC.

Image Processing (Elective II)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 Marks

Unit 1:
INTRODUCTION

Digital Image Representation, Elements of Digital Image Processing System, Elements of Visual Perception, Image Model, Sampling and Quantization – Uniform and Non-Uniform, Relationships between Pixels – Neighbors, Connectivity, Distance, Arithmetic/Logic Operations, Imaging Geometry – Basic Transformations, Perspective Transformations, Camera Model, Camera Calibration, Stereo Imaging.

Unit 2:
IMAGE TRANSFORMATION

Fourier Transform, Discrete Fourier Transform, Properties of 2D Fourier Transform, Fast Fourier Transform, FFT Algorithm and Implementation, Separable Image Transforms – Walsh Transform, Hadamard Transform, Hotelling Transform, Hough Transform.

Unit 3:
IMAGE ENHANCEMENT

Spatial-Domain Method and Frequency-Domain Method, Histogram Modification Techniques, Local Enhancement.

Image Smoothing – Averaging, Median Filtering, Lowpass Filtering, Ideal Filter, Butterworth Filter, Averaging of Multiple Images.

Image Sharpening – Differentiation, Highpass Filtering, Ideal Filter, Butterworth Filter.

Pseudo-Color Image Processing.

Unit 4:
IMAGE RESTORATION

Degradation Model, Diagonalization of Circulant and Block Circulant Matrices, Algebraic Approach to Restoration, Restoration in the Spatial Domain, Geometric Transformations – Spatial Transformation, Gray – level Interpolation.

IMAGE ENCODING

Fidelity Criteria, Encoding Process, Entropy, Huffman Code, Shift Codes, B-Codes, Error-Free Encoding, T-Algorithm, LML rule, IA rule, IP Algorithm, 1D/2D Run Length Encoding, Coding Considerations.

Unit 5:

IMAGE SEGMENTATION

Detection of Discontinuities, Point/Line/Edge Detection, Combined Detection, Edge Linking and Boundary Detection.

Thresholding – Global/Optimal Thresholding, Region Oriented Segmentation, Region Splitting and Merging.

IMAGE STANDARDS

JPEG and MPEG Standards, Graphics image file Formats – BMP, TIFF, PCX, GIF.

References:

1. Rafael Gonzalwis and Paul Wintz: "Digital Image Processing".
2. Philip: "Image Processing"
3. Anil K. Jain:, "Fundamentals of Digital Image Processing", PHI.
4. Sid Ahmed, "Image processing, Theory, Algorithm and architecture", McGraw Hill.

Laboratory Assignments:

1. Study of various Image File Format e.g. BMP, TIFF, PCX, GIF, JPEG for typical images:
   a. B/W images to develop a C Program to open a B/W image in one of the above formats and save the pixels in text files.
   b. Image of Constant intensity.
2. To develop a Program to study image attributes like Grey levels and Histogram, Grey level operations through look up tables.
3. To develop a Program to implement general purpose M*N masking operation (2D convolution). Program should be able to scale the output to save the same as an image.
4. To develop a Program to implement general purpose M*N 2D FFT. Program should be able to scale the output to save the same as an image.
5. To develop a Program to implement image enhancement technique like Noise filtering, Contrast Stretching, Edge Crispening.
6. To Study image Restoration techniques like deblurring, morphology operators like dilate and erode.
7. To develop a Program for Image Feature Extraction techniques like Edge detection, geometric moments: statistical parameters, Hough Transform.
8. To develop a program to implement a image compress technique using: .
   a. Transform method – DCT
   b. Huffman/RLE coding
   c. Quantization
9. Develop software to convert color image in BMP format into B/W image in BMP format.
10. Demonstrate usage of frame grabber card to capture image, video source and save it as standard file.

Staff member should frame minimum eight assignments based on the above topics. All image processing algorithms should be applied to standard image file format and program development should be on Windows Platform.

Compiler Construction (Elective II)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Oral: 25 marks

Unit 1:

Introduction to Compiler:

Translation issues: why to write compiler, compilation process in brief, front end and back end model, compiler construction tools, Interpreter and related issues, Cross compiler, Incremental compiler, boot strapping.

Lexical Analysis:

Introduction to Lexical Analysis - alphabet, token, lexical error, Block schematic of lexical analyzer, Explanation of NFA, DFA, Conversion from NFA to DFA, R.E. to optimized DFA, time vs. space complexity.

Construction of Lexical Analyzer - hand coding of Lexical Analyzer, I/O buffering, Lexical look ahead issue, comment handling, error recovery, Time and Space complexity. Automatic construction of Lexical Analyzer (LEX), LEX specification details.

Unit 2:

Syntax Analysis:

Introduction - Role of parsers and issues of separating lexical and syntax analysis, Types of grammar, CFG introduction, Expressing language through CFG.

Basic concepts in parsing - leftmost derivation, rightmost derivation, derivation tree, sentence, sentential form, language, derivation, parse tree, Ambiguous grammar.

Representation of CFG - Tree, Backous Naur Form, recursive rules, etc., Precedence and associativity, Disambiguous grammar.

Parsing technique - Top down: - Rd parser, Predictive LL (k) parser, Bottom up: - Shift-Reduce, SLR, LR (k), LALR, etc. (I/P, O/P, data structures required, block schematic, algorithm, limitations, efficiency to be covered about all above parsers).

Automatic construction of parser (YACC), YACC specification file details, error detection and recovery in YACC.

Unit 3:

Static Semantics and Intermediate Code Generation:

Need of semantic analysis, declaration processing, type analysis - unification algorithm, polymorphic algorithm, overloading function and operation, type conversion, equivalence of type, limitation of CFG, augmenting CFG with attributes, Attribute grammar - S-attribute grammar, with attribute evaluation, Integrating attribute evaluation with parsing, S-attributed definition for name, scope analysis, and its interaction with symbol table management, Symbol Table (ST) organization for block structured and non block structured languages, ST management, L-attributed definition for declaration processing, Comparison of different intermediate code forms, Intermediate code generation for declaration, assignment, iterative statements, case statements, arrays, structures, conditional statements, Boolean expressions, procedure / function definition and call (Usage of YACC utility to generate intermediate code).

Unit 4:

Code Generation:

Introduction - issues in code generation, target machine description, (may be taken as 8085 / 8086), operand description, partial result handling, register description, local and global register allocation, and assignment, Dynamic programming - principle and algorithm, Peephole optimization, unreachable code, time vs. space complexity trade off, Sethi Ullman algorithm for expression trees, Aho Johnson algorithm, different models of memory used in compiler (e.g. small, tiny in Turbo C), Code generator concept.

Unit 5:

Code Optimization:

Introduction, Principle sources of optimization, machine dependent optimization, machine independent optimization, Optimizing transformation, compile time evaluation, Common expression evaluation, value propagation, code movement and loop invariant computation, strength reduction, loop test replacement, dead code elimination.

Local optimization - DAG based local optimization, Introduction to global data flow analysis, control flow analysis - concepts and definition, data flow analysis - definition and concepts, formulation of data flow problem, solving data flow problem, performing data flow analysis, Representing data flow analysis, Computer data flow analysis, meet over paths, forward and backward data flow programming, computing global data flow information, Iterative data flow analysis, Interprocedural analysis and optimization.

Run time storage organization:

Storage allocation strategies, static, dynamic storage allocation, allocation strategies for block structured and non block structured languages, OS support required for I/O statements (e.g. printf, scanf), and memory allocation, deallocation related statements (e.g. new, malloc).

References:

1. Aho A.V., R. Sethi and J.D. Ullman, "Compiler Principle, Techniques and Tools", Addison Wesley.
2. Barret, Couch, "Compiler Construction Theory and Practice", Computer Science series, Asian Student Edition.
3. Dhamdhere D.M., "Compiler Construction Principle and Practice", McMillan India.
4. Gres D., ""Compiler Construction for Digital Computer", Wiley.
5. Holub A.J., "Compiler Design in C", Prentice Hall.
6. Tremblay, Sorenson, "Theory and Practice of Compilers".
7. "LEX and YACC", O'relly.
8. Muchnik, "Advanced Compiler Design and Implementation".

Laboratory Assignments:

1. Calculator (text or graphics) using LEX and YACC or Document Editor (find, replace, macro) using LEX and YACC, or similar kind of assignment using LEX and YACC.
2. Lexical analyzer for a subset of Pascal / C using LEX.
3. Syntax analyzer along with intermediate code generation (triple, quad) for a subset of Pascal / C using LEX and YACC.

4. Interpreter for an expression grammar.
5. Pascal to C converter using LEX and YACC.
6. Code generator for a Hypothetical Machine.
7. Any 2 optimization techniques on intermediate code generation
   - Constant expression evaluation.
   - Local copy propagation
   - Common sub expression elimination.
   - Loop invariant code movement.
8. Pseudo language compiler (for 8085 machine) using LEX and YACC.

Assignments 2, 3, 6, 7 are compulsory, and any two from the remaining.

Advanced Computer Architecture (Elective II)

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks

Unit I:

Introduction to parallel processing: -

Evolution of computer system, parallelism in uniprocessor system, parallel computer structures, Architectural classification schemes, parallel processing application, addressing schemes for main memory, S-access memory organisation and c-access memory organisation.

Unit II:

Principles of pipelining and vector processing: -

Principles of linear pipelining, classification of pipeline processors, General pipelines & reservation tables, design of pipelined instruction units, arithmetic pipelines, Multifunction and array pipelines, Internal forwarding & Register tagging, Hazard detection & resolution, study of Cray-1, characteristics of vector processor, Introduction to VLIW processor.

Unit III:

Array processors:-

SIMD computer organisation, masking and data routing mechanisms, Inter PE communication, static & dynamic networks, Mesh connected Illiac network, cube interconnection network, Barrel shifter & data manipulator, shuffle exchange and omega network, study of illiac-IV system and BSP system.

SIMD matrix multiplication, comparison of SIMD matrix multiplication with SISD matrix multiplication, parallel sorting on array processors, Associative search algorithms.

Unit IV:

Multiprocessor Architecure

Loosely coupled multiprocessors, tightly coupled multiprocessors, processor characteristics for multiprocessing, Interconnection network, time shared or common buses, crossbar switch, multiparty memories, classification of multiprocessor operating system, software requirements for multiprocessors, operating system requirements.

Parallel algorithms for multiprocessors:-

Classification of parallel algorithms, synchronized parallel algorithms, Asynchronized parallel algorithms, study of Cray X-mp system.

Unit V:

Multiprocessing control and algorithms: -

Interprocessing communication mechanism, process synchronization mechanism, synchronization with semaphores, conditional Sections, system deadlock and protection, deadlock problems, deadlock prevention avoidance, deadlock detection & recovery.

Data Flow Computer: -

Control flow versus data flow computers, data flow graphs & languages, advantages & potential problems, static data flow computers, systolic array archiktures

List of Experiments: - [Use C^{++} as a programming language]

1. Study of Cray-1 system.
2. Study of Illiac -IV system.
3. An O(n3) algorithm of SISD matrix multiplication.
4. Odd-even transposition sort algorithm.
5. Batcher's odd-even merge sort.
6. M (j,2) sorting algorithm.
7. M (j,k) sorting algorithm.

The term work will be based on above experiments. The term work marks will be based on performance in theory and practical, having a weightage of 40% and 60% respectively.

Books:-

1) Computer Architecture and parallel process - by kai Hwang and faye A. Briggs, MCGRAW-HILL editions.
2) Advanced Computer Architecture- By Kai Hwang, MCGRAW-HILL editions.
3) Parallel processing for super computers & AI.-By Kai Hwang & Degroot.
4) High performance computer Architecture- By Harrold stone.
5) Advanced computer Architecture- By Harrold Stone.

Computer Networks

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Practical: 25 marks

Unit 1:

Categories of Networks - Data, Voice, Video, Multimedia, Internet, Intranet, Repeaters, Bridges, Routers, Gateways, Servers, Transmission media and its specifications. Network softwares, Protocols and Services. Switching - Switch hierarchy, crossbar switches, space division switches, blocking / non-blocking switches, Circuit switching.

Data Link Layer:

Design Issues - services provided to the network layer, Framing, Error control, Flow control, Elementary Data Link protocols - Unrestricted Simplex protocol, Simplex Stop-and-Wait protocol, Simplex protocol for a noisy channel, Sliding window protocols - One bit, Go back n, selective repeat.

Unit 2:

Medium Access Sublayer:

Channel allocation problem - Static and Dynamic channel allocation in LAN/MAN.
Multiple Access Protocols - ALOHA, CSMA/CD, Collision free protocol, Wireless LAN protocols, IEEE standard 802.3, 802.4, 802.5, and their comparison.
Bridges - Transparent, Spanning Tree, Source routing, Remote Bridges.
High Speed LANs - FDDI, Fast Ethernet.

Unit 3:

Network Layer:

Design issues, Routing algorithms - shortest path, Flooding, Flow-based, Distance vector, Link state Routing, Routing for mobile hosts, Congestion Control Algorithms - Principles, Prevention Policies, Traffic Shaping.

Transport Layer:

Transport Services, Elements of Transport Protocols - Addressing, establishing / Releasing a connection, Flow control and buffering.

Unit 4:

TCP/IP:

Internetworking, Virtual circuits, connectionless internetworking, Tunneling, Internetwork Routing, Fragmentation, firewalls, TCP/IP Reference Model, Network Layer in the Internet - IP Protocol, IP addresses, Subnets, Masking, Internet Control Protocols - ICMP, ARP, RARP, Internet Multicasting, Mobile IP, CIDR, IPV4, IPV6, Internet Transport Protocols - TCP Service Model, TCP protocol, TCP Segment Header, TCP connection Management, TCP transmission policy, TCP Congestion Control, UDP, Data Link Layer in the Internet - SLIP, PPP.

Unit 5:

ATM:

Narrowband ISDN - ISDN Services, System Architecture, ISDN Interface, Broad Band ISDN and ATM, B-ISDN ATM Reference Model, Transmission in ATM networks, ATM switches, Data Link Layer in ATM, Network Layer in ATM networks - cell formats, connection setup, Routing and Switching, Traffic shaping and Policies, Congestion Control, ATM LANs, ATM AAL Layer Protocols - AAL 1, AAL 2, AAL 3/4, AAL 5, their comparison.

Internet and Applications:

DNS, Electronic Mail, World Wide Web, DHCP, FTP, HTTP.

References:

1. Andrew S. Tanenbaum, "Computer Networks", 3rd Ed., PHI.
2. Behrouz A. Forouzan, "TCP/IP Protocol Suite", McGraw Hill.
3. ED Taylor, "Networking Handbook", TMH.
4. Behrouz A. Forouzan, "Introduction to Data Communication and Networking", TMH.
5. Douglas E. Comer and David L. Stevens "Internetworking with TCP/IP - Vol. I, II, and III", PHI.

Laboratory Assignments:

1. Implementation of Data Link Layer Protocols, e.g. stop and wait protocol, sliding window protocol, and pipelining protocol.
2. Implementation of Network Layer functions, e.g. writing software for Routing Algorithms and Flow Control Algorithm.
3. TCP/IP socket programming.
4. Win socket programming.
5. Writing TCP/IP application like TELNET, Ping, FTP, Remote Execution.
6. Implementation of Network Security Algorithm, e.g. Data Encryption Standard and Ciphers.
7. Implementation of Application Layer, e.g. E-Mail.
8. Configuration of any two of the following:
   Router, DNS, Proxy Server, Web Server, Mail Server.
9. Data compression and Decompression (RLE, Huffman's coding)
10. Case study of existing networks, Study of network components and resources.
11. Implementation of ALOHA / CSMA/CD protocols.

Total 8 assignments to be completed with assignment no.1, 2, 3, and 8 compulsory.

Computer Graphics

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks
Practical: 25 Marks

Unit 1:

Basic Concepts:

Introduction to Computer Graphics, types of graphics devices, Display file structure, Display file Interpreter, Display Processors, Graphics file format – BMP, TIFF, PCX, GIF.

Interactive Graphics:

Graphics Standards, Graphics hardware, CRT display and controller, Interlaced and Non-interlaced display, Vector Scan and Raster Scan, Display Adapter – VGA, SVGA, Bios Video support, Graphics device drivers, Display buffers, Plotters, Digitizer, Scanners, Light pen.

Line and Circle Generation:

Line generation – DDA and Bresenham's Algorithm, Thick line segments, Antialising of lines, Circle generation - DDA and Bresenham's Algorithm, Character generation: Stroke Principle, Starburst Principle, Bit map Method.

Unit 2:

Polygons:

Types, representations, entering Polygons, Polygon Filling: Seed fill, Edge fill, Scan conversion Algorithm. Scan conversion: real time scan conversion, Solid area scan conversion, Run length encoding, Cell encoding.

Segments:

Concepts, Segment table, Segment Creation, Deletion, Renaming, Image Transformation.

Unit 3:

3D Geometry:

2D transformations Primitives and Concepts – Translation, Rotation, Rotation about an arbitrary points, Scaling and Shearing, 3D transformations, Rotation about an arbitrary axis, 3D viewing transformations, Concept of Parallel and Perspective Projections, Viewing Parameters, 3D clipping, Mid-Point subdivision Algorithm.

Unit 4:

Windowing and Clipping:

Viewing Transformation, 2D clipping, Sutherland-Cohen subdivision line clipping Algorithm, Midpoint subdivision Algorithm, Generalized clipping, Cyrus-Beck Algorithm, Interior and Exterior clipping, Polygon Clipping, Sutherland-Hodgman Algorithm.

Hidden surfaces and Lines:

Back-face removal Algorithm, Hidden line Methods, Z-Buffer, Warnock and Painters Algorithm, Floating Horizon.

Unit 5:

Light, Color and Shading:

Diffused Illumination, Point Source Illumination, Shading Algorithm, Color models – RGB, HVS, CYM etc., Illuminating back faces, Transparency, Reflection and Shadows.

Curves and Fractals:

Curve generation, Interpolation, Interpolating Algorithms, Interpolating Polygons, B-Splines and Corner, Bezier curves, Fractals, Fractal surfaces and lines.

Graphical User Interface:

Concepts of X-Windows, Concept of Client-Server Model, Protocols, Message Passing (only Gui related concepts), Motif – widget, gadget structure (only GUI concepts), Concepts of MS Windows, OpenGL: Why 3D?, Why OpenGL?, OpenGL and Animation.

Graphics Standard: Introduction to Graphics kernel system with basic Primitives.

Graphics Applications: Scientific and Engineering Applications, Business Applications, Application concept in Animation and Simulation.

References:

1. David F. Rogers, "Procedural Elements for Computer Graphics", Mc-Graw Hill International Editions.
2. Steven Harrington, "Computer Graphics A Programming Approach", Mc-Graw Hill International Editions.
3. Foley, Vandam, Feiner, Hughes, "Computer Graphics Principles and Practice", Addison Wesley.
4. Rao, Prasad, "Graphical user Interface (GUI) with X-Windows and MOTIF", New Age International limited Publisher.
5. Charles Petzold, "Programming Windows 95", Microsoft Publication.
6. Ron Fosner, "Open GL"

Laboratory Assignments:

1. Line/Circle drawing.
2. Polygon Filling.
3. 3D Transformation.
4. Segmentation.
5. Projections
6. Animation.
7. Windowing and Clipping Algorithm.
8. Polygon Clipping Algorithm.
9. Hidden line and Surfaces.
10. Curves and Fractals.
11. Programming of Display device drivers for various cards using Windows environment.
12. Study assignment on any latest GUI applications or Mini Project.

Programs must be designed using device independent graphics principles. Assignment number 11 and 12 is compulsory. Staff member should frame minimum eight assignments based on the above topics.

Advanced Microprocessors

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

Examination Scheme:
Theory: 100 Marks
Term work: 25 Marks

Unit 1:

80386 architecture, Functional diagram, support for pipelining, dynamic bus sizing, 80386 DX/SX differences, Programming model of 80386, register model, data types and addressing modes, Instruction set of 80386. Bus cycles with 16 and 32 bit data bus with timing, state diagrams, interrupt acknowledge, HOLD, HALT, RESET cycles.

Unit 2:

Memory management - Segmentation and Paging, 80386 modes - Real, Protected and V86 modes, Debugging support.

Unit 3:

Protection mechanism and privilege levels in protected & V86 modes, Privileged instructions, Page level protection mechanism, Multitasking support, task switching, task gates.

Unit 4:

Exceptions, Faults, Traps, Aborts, Interrupt handling, IDT, Interrupt, Trap gates. Interrupt handling in Real, Protected, and V86 mode, Coprocessor interface, 80387 interconnection and features.

Unit 5:

16 and 32 bit memory interfacing, I/O address space, I/O protection mechanism, Pentium family architecture and features.

References:

1. Tribel, "80386/486/Pentium Hardware and Software", PHI
2. Ross P. Nelson, "The 80386 Book", Microsoft Press.
3. Lance Leventhal, "Lance Leventhal's 80386 programming Guide", Bantom Books.
4. "Intel 80386, Programmer's Reference Manual", Intel Corporation.
5. Tyrley, "80386 advanced programmer manual",
6. Brey, "The Intel Microprocessors 8086/8088, 80186, 80286, 80386, and 80486, Architecture, Programming and Interfacing, PHI.
7. Moore, Papas, "The 80386 Programmer's Reference Manual.
8. Hans Peter, "The Indispensable Pentium", Addison Wesley.

Web sites

http://www.intel.com

Laboratory Assignments:

1. Assembly language programming for 80386/80387. Simple programs for instruction sets e.g. 32-bit calculator, Integer arithmetic, displaying time and date, etc. (At least 6 programs).
2. Study of 386, 486, Pentium, P-III motherboard (Any one).
   - i. Layout of motherboard and minimal peripherals.
   - ii. Study of CMOS setup.
   - iii. Study of cable set in computer system.
   - iv. Installation of peripherals.
3. PC diagnostics using Diagnostic Tools.
4. Paper work for 386/Pentium based microcomputer system design.

Technical Visit

Examination Scheme:
Term work: 50 Marks

During seventh and eighth term, every student shall visit minimum three industries or research organizations in the field of Computer Engineering arranged by the college and accompanied by teachers. The report of technical visits shall be submitted by every student in the end of eighth term which shall be evaluated by the concerned teachers through internal viva-voce.