M.Sc
COMPUTER SCIENCE
1st SEMESTER

With Effect From 2007 – 08 Admitted Batch

Syllabi

Chairman
Board of Studies
(2005-08)

Dept of Computer Science and Systems Engineering
College of Engineering
Andhra University
Visakhapatnam

MASTER OF SCIENCE IN COMPUTER SCIENCE
Course Structure and Scheme of Examination
1st SEMESTER
With effect from 2007 – 08 admitted batch

Code	Name of the Subject	Periods/ week		Max Marks			Credits
		T	L	Exam	IA	Total	
MSCS1.1	Discrete Mathematical Structures	3	-	70	30	100	3
MSCS1.2	Computer Organization	3	-	70	30	100	3
MSCS1.3	Data Structures and Algorithms	3	-	70	30	100	3
MSCS1.4	Systems Programming	3	-	70	30	100	3
MSCS1.5	Data Communications	3	-	70	30	100	3
MSCS1.6	Computer Organization Lab	-	3	50	50	100	2
MSCS1.7	Data Structures Lab	-	3	50	50	100	2

MSCS 1.1 Discrete Mathematical Structures

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1.INTRODUCTION: 4 Periods
Logic – Propositional Equivalences predicates- and quantifiers – Methods of Proof – Sets- Set Operations- Functions- Algorithms- Complexity of Algorithms- Integers- Applications of Number Theory- Matrices.

2. MATHEMATICAL REASONING, INDUCTION & RECURSION: 4 Periods
Proof Strategy- Recursive and Summation- Mathematical Summations- Mathematical Structures – Recursive Definitions- Recursive Algorithms- Program Correctness.

3 COUNTING TECHNIQUES: 6 Periods
Basics of Counting-Pigeon hole principle permutation and combinations- Binomial Coefficients- Generalized Permutations and Combinations-

4 ADVANCED COUNTING TECHNIQUES 6Periods
Recurrence relations- Solving Recurrence Relations - Divide-and-Conquer Algorithms and Recurrence Relations - Generating Functions- Inclusion Exclusion-Applications of Inclusion- Exclusion.

5 RELATIONS: 3 Periods
Relations- Properties-n-any Relations. Representing Relations-Closures of Relations Equivalence Relations- Partial Orderings.

6 .GRAPHS: . 6 Periods
Introduction- Terminology- Representing- Isomorphism- Connectivity-Euler and Hamilton Paths-Shortest-Path Problems- Planar Graphs- Graph Coloring

7) TREES: 5 Periods
Introduction- Applications of Trees- Traversals- Spanning trees- Minimum Spanning Trees.

8 .BOOLEAN ALGEBRA: 6 Periods
Boolean Functions- Representing of Boolean Functions- Logic Gates- Minimization of circuits.

9 .MODELING COMPUTATION: 8 Periods
Languages and Grammars Finite- State Machines With Output and No Output Language Recognition- Turing Machines.

Text Book:
Discrete Mathematics and its applications (Fifth Edition),Kenneth H. Rosen
Tata McGraw-Hill Publishing Company.

Reference:
Discrete Mathematics for computer Science & Mathematics(Second Edition), J.L.Mott, Abroham Kandel, & T.P.Baker. Prentice-Hall of India(Ltd).

MSCS 1.2 Computer Organization

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1.Digital Logic Circuits (Review) 4 Periods
Digital Computers, Logic Gates, Boolean Algebra, Map Simplification, Combinational Circuits, Flip-Flops, Sequential Circuits

2.Digital Components 5 Periods
Integrated Circuits, Decoders, Multiplexers, Registers, Shift Registers, Binary Counters, Memory Unit

3.Data Representation 4 periods
Data Types, Complements, Fixed-Point Representation, Floating-Point Representation, Other Binary Codes, Error Detection Codes

4.Basic Computer Organisation and Design 9 Periods
Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic

5.Microprogrammed Control 4 periods
Control Memory, Address Sequencing, Microprogram Example

6.Central Processing Unit 10 periods
Introduction, General Register Organisation, Stack Organisation, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC)

7.Input-Output Organisation 6 Periods
Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA)

8.Memory Organisation 6-Periods
Memory Hierarchy, Main Memory, Auxiliary Memory, Associate Memory, Cache Memory

Text Book:
Computer System Architecture (Third Edition),. Morris Mono - Pearson Prentice Hall,2007

MSCS 1.3 Data Structures & Algorithms

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1.Overview of C++ features: 8periods
Concepts of class & objects, I/O Streams, Constructors & Destructors, Parameter passing options, Functions & operator overloading, Inheritance, Virtual functions

2. Stacks & Queues: 10 periods
Introduction to Data Structures, ADT Stack and its implementation in C++, Evaluation of postfix expressions, ADT Queue and its implementation in C++, Generalising a Stack and its implementation using Templates.

3.Linked Lists: 4 periods
Defining & Implementing linked lists with creation, insertion and deletion operations in C++

4. Searching Algorithms 4 periods
Sequential search & Binary search algorithms, Implementation in C++, Estimation of Time complexity in Best, Worst and average cases, Classification of algorithms, Big-OH notation.

5. Sorting Algorithms 6 period
Methodology, Implementation and Algorithm Analysis of Insertion sort, Selection sort, Merge Sort and Quick Sort

6.Trees: 8 periods
Definition and Implementation of ADT Binary tree, AVL Trees, Heaps, Heap sort and Priority Queues.

7.Graphs: 8 periods
Definition of Graph, Representation of Graphs, Graph Traversal methods, Topological sorting, Minimum cost Spanning trees, Implementation of Kruskal's Algorithms, Finding shortest paths in a di-graph

Text Book:
Introduction to Data Structures & Algorithms with C++, GLENN W.ROWE,
Prentice Hall India,2003

Reference Book:
1. Data Structures in C++, N.S.KUTTI & P.Y.PADHYE, Prentice Hall India, 2003
2. Data Structures & Algorithms in C++, Adam Drozdek, Vikas Publishing House, 2002.

MSCS 1.4 Systems Programming

Instruction: 3 Periods/week External Assessment: 70 Marks
Internal Assessment: 30 Marks Time: 3 Hours

1. Introduction to Systems Programming 6 Periods
Machine Structure - Machine and Assembly Language Programming with IBM 360/370 - Grammars -Types of Grammars –Languages, FSM

2.Concepts of Single Pass, Two- Pass and Multi- Pass Assemblers 11 Periods
Design of a Single and Two-Pass Assembler

3.Macros and Macro Processors 11 Periods
Definition - Types of Macros - Macro Instructions - Features of Macro Facility like conditional Macro Expansion - Macros Calls within Macros - Macro Definitions within Macros - Design of Macro Processors: Single - Pass and Two - Pass.

4. Loaders: 11 Periods
Absolute Loader - Relocation Loader - Binders - Dynamic Loading and Linking - Design of Absolute Loader and Direct Linking Loaders.

5. General Model of Compiler 6 Periods
Phase of a Compiler - Detailed Discussion of different Phases

Introduction to Software Tools 3 Periods
Text editors, Interpreters, Program Generators, Debug Monitors.

Text Books**:**
Systems Programming , John J. Donovan
Systems Programming and Operating Systems, D.M.Dhamdhere

MSCS 1.5 Data Communications

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

Data Communication networks and open system standards: **3** Periods
Data communication networks, Standards, ISO reference model.

The Electrical Interface: 9 Periods
Transmission media, Attenuation and distortion sources, Signal types, Signal propagation delay, Public carrier circuits, Physical layer interface standards.

Data Transmission: 12 Periods
Data transmission basics, Asynchronous transmission, Synchronous transmission, Error detection methods, Data compression, Transmission control circuits, Communications control devices.

Protocol basics: 12 Periods
Error Control, Idle RQ, Continuous RQ, Link management.

Data link control protocols: 12 Periods
Application environments, Character-oriented protocols, Bit-oriented protocols.

Text book:
Data Communications, Computer Networks and Open Systems, Fred Halsall- Pearson Education, Low Price Edition, 4th edition, 2001.

Reference:
Data and Computer Communications, Williams Stallings, Prentic-Hall India , Eastern Economy Edition, 6th Edition, 2003

MSCS 1.6. Computer Organisation Laboratory

Practical : 3 Periods/week External Assessment: 50 Marks
Internal Assessment: 50 Marks Time: 3 Hours

Experiments in the following areas

T T L Characteristics
T T L I C Gates
Flip-Flops
Counters
Shift Registers
Multiplexers
Decoders
8085 Assembly Language Programming
PC Architecture
Hardware: Demonstration of Software lab environment (configuration & internal parts of PC)
Software: PC assembler (TASM / MASM) - minimum of 10 problems.

Note:
1 Week for Lab. Instruction, 1 Week for Repeat Expts, 7 Week for expts. 1 to 7
4 Weeks for expt. 9, 3 Weeks for expt. 9

MSCS 1.7 Data Structures Laboratory

Practical : 3 Periods/week External Assessment: 50 Marks
Internal Assessment: 50 Marks Time: 3 Hours

1. Implementation of linked lists with insert, delete, display, reverse function.
2. Array implementation of stack to evaluate a given postfix expression after accepting values of single character operands at run time.
3. Circular array implementation of queue with menu option like insert, delete, display, exit.
4. Construction of a Binary search tree and display in-order, pre-order, and post-order sequences of its nodes.
5. Implementation of ADT Binary Tree with create, insert, delete and find height operations.
6. Implementation of Hash Table.
7. Write C++ program for merge sort.
8. Write C++ program for quick sort.
9. Write C++ program for Heap sort.
10. Implementation of Dijikstra's Algorithm.
11. Implementation of Kruskals Algorithm.
12. Finding topological ordering of the nodes of a graph.

M.Sc
COMPUTER SCIENCE
2nd SEMESTER

With Effect From 2007 – 08 Admitted Batch

Syllabi

Chairman
Board of Studies
(2005–08)

Dept of Computer Science and Systems Engineering
College of Engineering
Andhra University
Visakhapatnam

MASTER OF SCIENCE IN COMPUTER SCIENCE

Course Structure and Scheme of Examination

FIRST YEAR - II SEMESTER

With Effect From 2007-08 admitted batch

Sub. Ref. No.	Name of the Subject	Periods		Max Marks			Credits
		Lec.	Lab.	EA	IA	Total	
MSCS2.1	Computer Networks	3	-	70	30	100	3
MSCS2.2	Computer Graphics	3	-	70	30	100	3
MSCS2.3	Operating Systems	3	-	70	30	100	3
MSCS2.4	Artificial Intelligence	3	-	70	30	100	3
MSCS2.5	Object Oriented Programming	3	-	70	30	100	3
MSCS2.6	Unix Programming Lab	-	3	50	50	100	2
MSCS2.7	Data Communications Lab	-	3	50	50	100	2

MSCS 2.1 COMPUTER NETWORKS

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

LOCAL AREA NETWORKS:
6.1.1Topology, 6.1.2Transmission Media, 6.1.3 Medium Access Control Methods, 6.2.1CSMA/CD Bus, 6.2.2, Token Ring, 6.2.3Token Bus, 6.3Performance, 6.4 Wireless Lans, 6.4.1Wireless Media, 6.5Protocols, 6.5.1MAC Sub Layer Services, 6.5.2LLC Sub Layer, 6.5.3Network Layer, 7.5Bridges, 7.6Transparent Bridges, 7.7 Source Routing Bridges, 7.7.3 Internetworking With Different Types.

WIDE AREA NETWORKS:
8.1Characteristics Of Public Data Networks, 8.1.1Circuit And Packet Switching, 8.1.2Data Grams And Virtual Circuits, 8.2 Packet Switched Data Networks, 8.2.1 Physical Layer, 8.2.2Link Layer, 8.2.3Network Layer, 8.4ISDN, 8.4.1User Interfaces, 8.4.2Network Access Points, 8.4.3Channel Types, 8.4.4 User Network Interface, 8.4.5User Interface Protocols, 8.4.6Signaling Protocols, 8.4.7Frame Relay Services.

INTERNETWORKING:
9.1Internetworking Architecture, 9.2Internetworking Issues, 9.5Internet IP, 9.5.1 Address Structure, 9.5.2Data Grams, 9.5.4Fragmentation And Reassembly, 9.5.5 Routing, 9.5.6Internet Control Message Protocol, 9.6IPv6, 9.6.1Data Gram Structure, 9.6.2 Multicast Support

TRANSPORT PROTOCOL:
11.1User Data Gram Protocol, 11.2TCP, 11.2.1Reliable Stream Service, 11.2.2 Protocol Operations.

APPLICATION SUPPORT PROTOCOL:
12.1Session Layer, 12.1.1Token Concept, 12.2Presentation Layer, 12.4Data Encryption, 12.4.1Terminology, 12.4.2Basic Techniques, 12.4.3 DES, 12.4.4 RSA, 12.4.5 Message Authentication.

TCP/IP APPLICATION PROTOCOLS:
13.1.1Establishing Transport Connection, 13.1.2TELNET, 13.1.3FTP, 13.1.4SMTP, 13.1.5SNMP Worldwide Web (From A. S. Tannenbaum Section 7.6), 14.1Directory Services, 14.1.1 Domain Name System.

Text Book:
Fred Halsall, Data Communications, Computer Networks And Open Systems, Fourth Edition, Addison Wesley [Pearson Education 2000]

Reference Books:
1. Andrew S. Tanenbaum, Computer Networks, Third Edition. PHI, 1999.
2. Larry L Peterson And Bruce S Davie, Computer Networks- A Systems Approach, Second Edition, Harcourt Asia, Pte. Ltd, 2000.

MSCS 2.2 COMPUTER GRAPHICS

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1.Vector and Raster Graphic Fundamentals - Line Drawing Algorithms: Simple DDA, Symmetric DDA and Bresenhana's (B11 quadrants). Circle Generator.

2.Different types of Graphical I/O devices and their classification.

3.Co-OrdinateSystems(2D) – Homogeneous Co-Ordinates- Matrix Representation - Windows, Viewports – Windowing transformation

4.Line clipping algorithms - Polygons-Inside Test - Polygon Clipping Algorithm - Scan Conversion algorithms.

5.Two Dimensional Transformations – Matrix Representation - Concatenation of 2D transformations.

6.Display File Segmentation – Compilation – Data Structures used for Implementation Display Files.

7.Three dimensional Transformations – Projections - Viewing Transformation - Curves and Surfaces.

8.GeometricModelsandPictureStructure,

9.DesignofGraphicPackages.

Text Books :

Principles of Interactive Computer Graphics, Newman and Sproull (McGraw Hill)
Computer Graphics, Donald Hearn and M.Pauline Baker (PHI 2nd Edition)

Reference:

Procedural Elements of Computer Graphics, Rogers (McGraw Hill)
Mathematical Elements of Computer Graphics, Rogers (McGraw Hill)

MSCS 2.3 OPERATING SYSTEMS

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1. Overview of Operating System Functions, Batch Processing Systems, Multi programming Operating Systems, Time Sharing Systems.

2. Processor Management
   Jobs, Programs and Processor, Job Scheduling, Process Scheduling, Process Synchronization, Process Communication, Dead Locks, Process Management in Multiprocessor Operating Systems.

3. Storage Management, Contiguous, Noncontiguous Storage Allocation, Virtual Storage implementation using Paging and Segmentation.

4. Information Management

IO Organisation and Physical IOCS, File Organisation, Logical IOCS, File Systems, The Unix File System

5. Concurrent Programming

Implementing Process Precedence, Software Implementation of Critical Section, Evolution of Languages Features for Concurrent Programming, Monitors, Concurrent Programming in ADA

Text Book:

1. Systems Programming and Operating Systems (Part II – Operating Systems), Dhamdhare, 2nd Edition, TMH

2. Applied Operating System Concepts – Avi Silberschatz, Peter Galvin, Grey Gagne

MSCS 2.4 ARTIFICIAL INTELLIGENCE

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1. Introduction to Artificial Intelligence:

Overview of AI – Definition of AI, Relationship between AI Systems and other computing systems, comparison between AI programming and other conventional programming; AI and related fields; Key Issues in AI Research, AI problems- Examples; problem spaces, production systems and characteristics; knowledge – general concepts.

2. Knowledge Representation:

Approaches to knowledge representation, Issues in knowledge representation, Formal systems – basic concepts, Symbolic logics - Syntax and semantics of FOPL, properties of w.f.f, clausal forms, Resolution principle, Examples of Resolution; Structural knowledge – graphs, frames, C.D's and scripts; probabilistic reasoning- Bayesean Networks, Dampster – Shafer theory; Non Monotonic Reasoning – TMS, Model and Temporal logics, Fuzzy sets & Fuzzy logics.

3. **Knowledge organisation and Manipulation:**

Search and control strategies - Examples of research problems, uninformed search techniques, Informed and Heuristic search techniques; Matching Techniques – Structures used in Matching, Measures of matching, partial matching, Fuzzy Matching Algorithms and RETE Matching Algorithm.

4. AI languages:

LISP – Basic list manipulation functions, predicates, Conditionals, Input, output and local variables, Iteration and Recursion in LISP, property lists and Arrays, Prolog – Introduction, facts, questions, variables, conjunctions, syntax of character, Operators, equality, matching, arithmetic expressions; Goals; Back tracking, cut predicates; Input and output operations.

Text Books:

1. Artificial Intelligence by Elaine RICH and Kevin Knight – TMH
2. Introduction to AI & Expert systems by O.W. Patterson – PHI
3. Artificial Intelligence by N J Nilsson HARCOURT ASIA (Pvt) Ltd.

Reference Books:

1. Programming prolog by Clockson & Mellish - Narosa
2. Artificial Intelligence by P.H Winston - AWL

MSCS 2.5 Object Oriented Programming

Instruction: 3 Periods/week External Assessment: 70Marks
Internal Assessment: 30 Marks Time: 3 Hours

1. Object oriented systems development and designing:

   Traditional and object oriented software cycles, Objectives of OOP, Object oriented analysis

2. Object oriented programming in C++:
   Concepts of data abstraction, encapsulation, Introduction to objects, classes and instances, static members, inheritance, polymorphism,
   Overloading and information hiding – function overloading, operator overloading in C++, Memory management – constructors, overloading of constructors, copy constructors, destructors, constructors and information hiding, concepts of file handling in C++.

3. Inheritance and polymorphism in C++:
   Inheritance – derived and base classes, protected member, overriding member function, class hierarchies, multiple inheritance – containership, virtual functions, late binding – pure virtual functions – abstract classes – friend functions, Friend classes, static functions, this pointer, templates- function templates, class templates, Exception Handling in c++

4. Object oriented Mechanism in JAVA:
   Class definition in JAVA, constructors, inheritance, polymorphism in JAVA, Access specifications in JAVA, Interfaces and packages in JAVA, Error & exception handling in JAVA, Building JAVA Applets and Applications in JAVA
   Multi threads in JAVA - introduction to threads, the thread class, the runnable interface, thread states, thread priorities, thread groups & synchronization.
   Graphics in JAVA

Text Books:

1. Object Oriented Programming in C++, Robert Lafore, Galgotia
2. Object Oriented programs, Bala Guruswamy, TMH
3. Programming in JAVA, Dietel & Dietel, AWL
4. Introduction to JAVA programs, Y.Danial Liang, PHI
5. Complete Reference JAVA, 3ed, Peter Naughton & H.Schimalt – TMH

MSCS 2.6 UNIX PROGRAMMING LAB

Instruction: 3 Periods/week External Assessment: 50 Marks
Internal Assessment: 50 Marks Time: 3 Hours

STUDY OF LABORATORY ENVIRONMENT:
Hardware Specifications: system details, network details
Software Specifications: O.S. details, compilers
FAMILIARIZTION OF UNIX COMMANDS and UTILITIES
Simple programs using ***make*** utility
Simple programs to display process group Ids: PID, PPID, GID

SIMPLE UNIX-C PROGRAMS:

Display "Unix Programming Lab." N times using LIBRARY FUNCTION CALLS and user Defined function ***dsply(int)***. N is an integer given through keyboard upon prompting.

Display "Unix Programming Lab." N times using SYSTEM CALLS and user defined function ***dsply(char *)***. N is an integer given through keyboard upon prompting.

Write "Unix Programming Lab." N times in a file: outfile.txt in current directory using LIBRARY FUNCTION CALLS and user defined function ***writefile(int)***. N is an integer given through a file infile.txt.

Write "Unix Programming Lab." N times in a file: outfile.txt in current directory using SYSTEM CALLS and user defined function ***writefile(char *)***. N is an integer given through a file infile.txt.

PROGRAMS using system calls that provides some error checking

1.Checking error numbers with externally declared integer **errno** and using **perror** library function

2.Display all of the available system error messages in a numbered two-columns-per-line-format.

3.Write your own error messaging function that is called when a file manipulation failure occurs. The function should provide a more descriptive, user-friendly interface than **perror**. It might be helpful to examine the header file<sys/errno.h>and the manual page entry for Intro in section 2(i.e., man –s2 Intro) prior to start.

4. Display process group ID information.

5. Displaying system limits like Max size of argv,Max #Child Processes,etc using sysconf

PROGRAMS using Processes:
1.Chain of processes

2.Fan of Processes

3.Write a program that determines by trial and error the number of files a process can have simultaneously open. Be sure to remove (investigate the unlink system call) any files that you generate.

4.Predict what will happen when a process forks a child process and the child process issues a chdir system call – will the current directory for the parent be changed as well? Write a Program that substantiates your answer.

PROGRAMS using COMMAND LINE ARGUMENTS

PROGRAMS for Simple Shell and Complex Shell with cd command, editor command, etc.)

PROGRAMS for Primitive Communications: Lock Files, Signal and Signal management Calls

PROGRAMS using Pipes: Unnamed Pipes, Named Pipes

PROGRAMS using Message Queues: Creating a Message Queue, A Client-Server Message Queue

PROGRAMS using Semaphores: Creating and Accessing Semaphore Sets, Semahore Operations

PROGRAMS Using Shared Memory: Creating Shared Memory Segment, Using a File as Shared memory

PROGRAMS using RPCs: Executing Remote Commands in a C program

MSCS 2.7 DATA COMMUNIATIONS LAB

Instruction: 3 Periods/week External Assessment: 50 Marks
Internal Assessment: 50 Marks Time: 3 Hours

DATA COMMUNICATIONS EXPERIMENTS

1.1 PC-to-PC COMMUNICATIONS UNDER **DOS** WITH **NULL MODEM**
a) Using Serial Ports and RS-232 C Cable Connection
b) Using Paralell Ports and Parallel Cable Connection

1.2 PC-to-PC COMMUNICATIONS UNDER **DOS** WITH **MODEM** and **4-LINE EXCHANGE** Using Communication Software: COMIT or XTALK

1.3 PC-to-PC COMMUNICATIONS UNDER **WIN 98's** DIRECT CABLE CONNECTION with NULL MODEM
a) Using Serial Ports and RS-232 C Cable Connection
b) Using Paralell Ports and Parallel Cable Connection

1.4 PC-to-PC COMMUNICATIONS UNDER **WIN 98's DIAL-UP** NETWORKING WITH **MODEM** and **4-LINE EXCHANGE**

1.5 PC-to-PC COMMUNICATIONS UNDER **WIN 98's HYPER TERMINAL** WITH **MODEM** and **4-LINE EXCHANGE**

1.6 PC as **T**ERMINAL using Terminal Emulator Software to Connect 8085/8086 µP. trainer

1.7 **INERNET** CONNECTION SET-UP USING **DIAL-UP** NETWORKING

LAN EXPERIMENTS:

2.1 **THIN ETHERNET** LAN WITH BUS TOPOLOGY with a minimum of two systems
a) Windows Peer-to-Peer Network b) Windows NT Client-Server Network

2.2 **THIN ETHERNET** LAN WITH STAR TOPOLOGY with a minimum of two systems
a) Windows Peer-to-Peer Network b) Windows NT Client-Server Network

2.3 **THICK ETHERNET** LAN WITH BUS TOPOLOGY with a minimum of two systems
a) Windows Peer-to-Peer Network b) Windows NT Client-Server Network

2.4 **THIN ETHERNET** LAN WITH BUS TOPOLOGY with a minimum of two systems
a) Novell Peer-to-Peer Network b) Novell Client-Server Network

2.5 **THIN ETHERNET** LAN WITH STAR TOPOLOGY with a minimum of two systems
a) Novell Peer-to-Peer Network b) Novell Client-Server Network

2.6 **TERMINAL NETWORK** WITH **UNIX/LINUX SERVER** and one or two Terminals

2.7 **TERMINAL NETWORK** WITH **UNIX/LINUX SERVER, Terminal Server,** and one or two terminals

M.Sc
COMPUTER SCIENCE
3rd SEMESTER

With Effect from 2007 – 08 Admitted Batch

Syllabi

Chairman
Board of Studies
(2005-08)

Dept of Computer Science and Systems Engineering
College of Engineering
Andhra University
Visakhapatnam

MASTER OF SCIENCE IN COMPUTER SCIENCE

Course Structure and Scheme of Examination

III SEMESTER

With Effect From 2003-04 admitted batch

Sub. Ref. No.	Name of the Subject	Periods		Max Marks			Credits
		Lec.	Lab.	EA	IA	Total	
MSCS3.1	Object Oriented Software Engineering	3	-	70	30	100	3
MSCS3.2	RDBMS	3	-	70	30	100	3
MSCS3.3	Network security	3	-	70	30	100	3
MSCS3.4	Theory of Computation	3	-	70	30	100	3
MSCS3.5	Elective	3	-	70	30	100	3
MSCS3.6	RDBMS Lab	-	3	50	50	100	2
MSCS3.7	Visual Programming Lab	-	3	50	50	100	2

Electives:

1. Embedded Systems
2. Data Warehousing & Data Mining
3. Bioinformatics
4. Image Processing

MSCS3.1 OBJECT ORIENTED SOFTWARE ENGINEERING

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

1. Software & Software Engineering
   The nature of software, software engineering and as branch of engineering profession, stakeholders in software engineering, software quality, software engineering projects,
2. Developing requirements
   Domain analysis, software project's starting point, problem definition and scope, What is requirement?, type of requirements, gathering and analyzing of requirements, requirements document types, reviewing, managing change in requirements,
3. Modeling with classes
   UML, essentials of UML class diagrams, associations and multiplicity, generalization, instance diagrams,
4. Using design patterns
   Pattern introduction, the abstraction-occurrence pattern, general hierarchical pattern, the play-role pattern, the singleton pattern, the observer pattern, the delegation pattern, the adaptor pattern, the façade pattern, the immutable pattern, the read-only interface pattern and the proxy pattern.
5. Focusing on users and their tasks
   User-centred design, characteristics of users, developing use case models of systems, the basics of user interface design, usability principles, evaluation users interfaces
6. Modeling interactions and behavior
   Interaction diagrams, state diagrams, activity diagrams
7. Architect ring and designing software
   The process of design, principles leading to good design, techniques for making good design decisions, software architecture, writing a good design document
8. Testing and inspecting to ensure high quality
   Basic definitions of defect, error and failure, effective and efficient testing, defects in ordinary and numerical algorithms, defects in timing and coordination, defects in handling stress and unusual situations, documentation defects, writing formal test cases and test plans, strategies for testing large software, inspections, quality assurance in general
9. Managing the software process
   Project management, software process model, cost estimation, building software engineering teams, project scheduling and tracking, contents of a project plan

Text Book: Object-Oriented Software Engineering Practical software development using UML and Java by Timothy C. Lethbridge & Robert Langaniere , Tata Mcgraw-Hill Co

MSCS3.2 RELATIOINAL DATA BASE MANAGEMENT SYSTEMS

Instruction : 3 Periods/Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

Introduction to Organization of Databases – Components of DBMS – Data Models – Entity Relationship Model

Basic file Systems: Introduction – Secondary Storage Devices – Files and Buffer Management – File Organization
Sequential File Organization – Indexed Sequential File Organization –Creation and Manipulation of Indexed Sequential File – Hashing – key to address transfer – Overflow Management in Hashed Files.

B-Tree based Indexed File organization – Secondary Indexes:
Organization and usage – File Organization based on Dynamic Hashing with Deferred Splitting – Linear Splitting.

Relational Data Model - Relational Algebra- ISBL – Relational Calculus – The Domain Calculus system - SQL.

Relational Database Design: Integrity Constraints
Functional Dependency – Logical implication of Dependencies – Normal Forms – Decomposition of Relational Schemes – Design Procedures

Security: Introduction – Access control – Crypto-systems – Statistical Database Security.
Concurrency control and Data base Recovery: Transaction – Data Base System Architecture – Serializability – Locking – Non-Locking Schedules – Data Base Recovery.

Text Book:
Data Base Management Systems
Arun K Majumdar and Pritimoy Bhattacharya
Tata-McGrahill Publishing Co Ltd 1996

MSCS3.3 NETWORK SECURITY

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

Introduction:
Attacks, services, mechanisms-security attacks-security services-Model for network security-Internet standards.

Conventional encryption and message confidentiality**:**
Conventional encryption principles-conventional encryption algorithms-cipher block modes of operations-location of encryption devices-key distribution

Public Key cryptography and authentication:
Approaches to message authentication-Secure Hash Functions and HMAC-Public Key Cryptography Principles_ Public Key Cryptography Algorithms-Digital signatures-Key management

Authentication & E mail Security:
Kerberos-X.509 Directory Authentication Services-PGP-S/MIME.

IP Security:
IP security overview-IP Security Architecture-Authentication Header-Encapsulating Security Pay load-Combing Security Associations-Key Management.

Web Security:
Web Security Requirements-SSL and Transport Layer Security-SET-
Network Management Security.

System Security:
Intruders-viruses-related threats-Fire Design principles-Trusted Systems

Text Book:
Network Security Essentials Applications and Standards, by William Stallings
Pearson Education Asia, New Delhi

Reference Books:
1) Network Security: Private Communication in a Public World, Kaufman
Pearson Education Asia, New Delhi.
2) Cryptography and Network Security, by William Stallings
Pearson Education Asia, New Delhi.

MSCS3.4 THEORY OF COMPUTATION

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

1. Finite Automate & Regular Expression
   Basic concepts of Finite stats systems. Deterministic and non Deterministic Finite Automation, Regular Expressions, Relationship between Regular expression & Finite Automate Minimization of Finite Automation Mealy & Moore Machines.
2. Regular sets to Regular Grammars.
   Basic Definition of Formal Language and Grammars chomiskian Hierarchy of Languages and Automata. Regular Sets and Regular Grammars, closure proportion of Regular sets, Pumping lemma for Regular sets, Decision Algorithms for Regular sets, Myhell _Nerode theory & Organization of Finite Automata.
3. Context Free Languages & pushdown Automation, context free grammars, single ficatum of context free grammars, Normal forms. Pumping lemma for CFL, closure proportion of CFL, Push down automata, Language accepted by PDA, Relation between CFL & PDA
4. Computability & Recursion
   Basic definition of computable and non-computable functions, primitive Recursive, Recursive and partial Recursive functions, RICE theorem and Greibach theorem, PCP and un decidability

5. Turing Machines

Turing Machine Models, Organization and Representation of Turing Machines, Computable Languages and Functions of Turing Machines. Techniques for Construction of Turing Machines, Universal Turing Machines, Halting Problem, Modifications of Turing Machines.

Text Books:

1. Introduction To Automate Theory, Languages & Computation by J.E Hopcraft & JD Ullman, Narosa Publications.

Reference Books:

1. Mathematical theory of computation By Mannaz
2. Theory of Computer Science by KLP Mishra & N.Chandra Sekharan, PHI
3. Mathematical Foundations of Computer Science by BECKMAN
4. Introduction to Languages & Theory of Computation By J.C. Martin, TMH

MSCS3.5 EMBEDDED SYSTEMS (Elective)

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

1. Examples of Embedded Systems, Hardware Fundamentals,

2. Microprocessors and Microcontrollers, The 8051 Architecture

3. 8051 Assembly Language Programming: Moving Data, Arithmetic and Logical Operations, Jump and Call Instructions

4. Interrupts and Survey of Software Architectures

5. Introduction to Real-Time Operating Systems

6. Operating System Services

7. Basic Design Using a Real-Time Operating System

8. Embedded Software Development Tools

TEXT BOOKS:

1. The 8051 Microcontroller, Architecture, Programming, & Applications, by Kenneth J.Ayala, Penram International Publishing(India), Second Edition1996 (Ch.1,Ch3.,Ch5,Ch6,Ch7,Ch8)
2. An Embedded Software Primer , David E. Simon, Pearson Education Inc., 1999 (Ch.2, Ch.3, Ch.4,Ch5,Ch.6,Ch.7,Ch.8,Ch9)

REFERENCE BOOKS:

1. Embedded Systems, Architecture, Programming and Design, by Raj Kamal TMH, 2003
2. Embedded Real Time Systems Programming, by Sriram V Iyer and Pankaj Gupta, TMH, 2004

MSCS3.5 DATA WAREHOUSING (Elective)

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

1. Introduction to Data Ware housing.
2. Data warehousing Environment: Architecture perspectives.
3. Modeling and Design Techniques for the Central Data Warehouse.
4. Multi – Dimensional Data Modeling.
5. Data Warehouse Usage
6. Populating Data Warehouse Environment.

Text Book:

An Introduction to Building the Data Warehouse by IBM, Prentice Hall of India.

Reference Books :

1. Data Warehousing in the Real world by Sam Anahory & Murray Addison Wesley.
2. Building the Data warehouse by W.H Inmon , 3rd Edition John Wiley & Sons Incorporation.

MSCS3.5 BIOINFORMATICS (Elective)

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

Motivation and Expectation**:**
Students are expected to know the fundamentals of Engineering in Medicine and biology, which is emerging as an interesting field. Students are expected to use The Internet extensively to understand the subject.

1. Introduction:

Definitions, Sequencing, Biological sequence/structure, Genome Projects, Pattern recognition an prediction, Folding problem, Sequence Analysis, Homology and Analogy.

2. Protein Information Resources

Biological databases, Primary sequence databases, Protein Sequence databases, Secondary databases, Protein pattern databases, and Structure classification databases**.**

3. Genome Information Resources

DNA sequence databases, specialized genomic resources

4. DNA Sequence analysis

Importance of DNA analysis, Gene structure and DNA sequences, Features of DNA sequence analysis, EST (Expressed Sequence Tag) searches, Gene hunting, Profile of a cell, EST analysis, Effects of EST data on DNA databases

5. Pair wise alignment techniques

Database searching, Alphabets and complexity, Algorithm and programs, Comparing two sequences, sub-sequences, Identity and similarity, The Dotplot, Local and global similarity, different alignment techniques, Dynamic Programming, Pair wise database searching.

6. Multiple sequence alignment

Definition and Goal, The consensus, computational complexity, Manual methods, Simultaneous methods, Progressive methods, Databases of Multiple alignments and searching

7. Secondary database searching

Importance and need of secondary database searches, secondary database structure and building a sequence search protocol

8. Analysis packages

Analysis package structure, commercial databases, commercial software, comprehensive packages, packages specializing in DNA analysis, Intranet Packages, Internet Packages.

Text Books:

1. Introduction to Bioinformatics, by T K Attwood & D J Parry-Smith
Addison Wesley Longman

Reference:

1. Bioinformatics- A Beginner's Guide, Jean-Michel Claveriw, Cerdric Notredame, WILEY DreamTech India Pvt. Ltd

2. Sequence Analysis in A Nutshell by Scott Markel & Darryl Leon O'REILLY

MSCS3.5 IMAGE PROCESSING (Elective)

Instruction : 3 Periods /Week **Sessional Marks : 30**
Univ. Exam : 3 Hours **Univ. Exam Marks:70**

1. Fundamentals of Image Processing

Image Acquisition, Image Model, Sampling, Quantization, Relationship between pixels, distance measures, connectivity , Image Geometry, Photographic film. Histogram: Definition, decision of contrast basing on histogram, operations basing on histograms like image stretching, image sliding, Image classification. Definition and Algorithm of Histogram equalization.

2. Image Transforms:-

A detail discussion on Fourier Transform, DFT,FFT, properties
A brief discussion on WALSH Transform , WFT, HADAMARD Transform, DCT.

3. Image Enhancement: (by SPATIAL Domain Methods)
   a Arithmetic and logical operations, pixel or point operations, size operations,
   b. Smoothing filters-Mean, Median, Mode filters – Comparative study
   c.. Edge enhancement filters – Directorial filters, Sobel, Laplacian, Robert, KIRSCH Homogeneity & DIFF Filters, prewitt filter, Contrast Based edge enhancement techniques. – Comparative study
   d. Low Pass filters, High Pass filters, sharpening filters. – Comparative Study
   e. Comparative study of all filters
   f. Color image processing.

4. Image enhancement : (By FREQUENCY Domain Methods)

Design of Low pass, High pass, EDGE Enhancement, smoothening filters in Frequency Domain. Butter worth filter, Homomorphic filters in Frequency Domain Advantages of filters in frequency domain, comparative study of filters in frequency domain and spatial domain.

5. Image compression: Definition, A brief discussion on – Run length encoding, contour coding, Huffman code, compression due to change in domain, compression due to quantization Compression at the time of image transmission. Brief discussion on:- Image Compression standards.

6. Image Segmentation: Definition, characteristics of segmentation.
Detection of Discontinuities, Thresholding Pixel based segmentation method.
Region based segmentation methods – segmentation by pixel aggregation, segmentation by sub region aggregation, histogram based segmentation, spilt and merge technique. Use of motion in segmentation (spatial domain technique only)

7. Morphology:-
   Dilation, Erosion, Opening, closing, Hit-and-Miss transform, Boundary extraction, Region filling, connected components, thinning, Thickening, skeletons , Pruning Extensions to Gray – Scale Images

   Application of Morphology in I.P

Text Book:

Digital Image Processing , by Rafael C. Gonzalez and Richard E. Woods Addision Wesley

Reference books:

1. Fundamentals of Electronic Image Processing by Arthyr –R – Weeks, Jr. (PHI)
2. Image processing, Analysis, and Machine vision by Milan Sonka vaclan Halavac Roger Boyle, Vikas Publishing House.

MSCS 3.6 RDBMS LAB

Lab : 3 Periods /Week **Sessional Marks : 50**
Univ. Exam : 3 Hours **Univ. Exam Marks:50**

File Structures

Reading a stream of fields, record structures and its length indicators, Mixing of numbers and characters, Use of a hex dump, Retrieving records by keys using sequential search, direct access

Indexing and indexed sequential files

Index file, inverted file operations, usage of B and B^{++} trees

Hashing files

Hashing functions, algorithms, record distribution and collision resolution by progressive over flow, Extendable hashing and hashing performance

Above will be implemented in C^{++}

RDBMS

Each student is assigned with a problem. The student is to develop a logical and physical database design for the problem.

A.The logical design performs the following tasks:
1. Map the ER/EER diagrams to a relational schema. Be sure to underline all primary keys, include all necessary foreign keys and indicate referential integrity constraints.
2. Identify the functional dependencies in each relation
3. Normalize to the highest normal form possible

B. Perform physical design based above logical design using Oracle on Windows platform

C. Perform DML and DLL using PL/SQL

Reference books:

1. Oracle SQL and PL/SQL handbook by John Adolph Palinski
Pearson Education
2. Oracle PL/SQL Programming, by Steven Feuerstein
O'Reilly Publishers

MSCS 3.7 VISUAL PROGRAMMING LAB

Lab : 3 Periods/Week **Sessional Marks : 50**
Univ. Exam : 3 Hours **Univ. Exam Marks:50**

Experiments using java AWT/swing (JFC)

Reading Data From Key Board
Handling Buttons, Labels, Text Fields, Text Areas, Scroll Bar
Handling Check Boxes, Radio, List Box, Sliders
Handling Menu
Handling Swing Components Like Progress Bars
Handling Databases Using JDBC Native Driver

Experiments using VC++

Reading Data From Key Board
Handling Buttons, Labels, Text Fields
Handling Check Boxes, Radio, List Box, Sliders.
Handling Menu. Tool Bars
File Handling
Internet Programming
Creative Active X Controls

Reference books:

1. VC++, by Steven Holzner, BPB publisher

2. JAVA tutorial, Person Education.

M.Sc
COMPUTER SCIENCE
4th SEMESTER

With Effect from 2007 – 08 Admitted Batch

Chairman
Board of Studies
(2005-08)

Dept of Computer Science and Systems Engineering
College of Engineering
Andhra University
Visakhapatnam

M.Sc (COMPUTER SCIENCE)
With Effect From 2007-08 admitted batch
Scheme of Examination

IV SEMESTER

SL.NO.	NAME OF THE SUBJECT	EXTERNAL EVALUATION
MSc(CS)	Project Work	**100 Marks**

Credits:13

==========
COVER / TITLE PAGE :
==========

T I T L E

The Thesis report submitted in partial fulfillment
of the requirements for the award of degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE

By

STUDENT NAME with regd. No.

< University/College Emblem>

< College Address>

CERTIFICATE FOR STUDENTS WHO HAD DONE PROJECT IN THE DEPARTMENT

CERTIFICATE

This is to certify that it is a bonafide work done by Mr./Ms.Mrs._____ during the year 200 - 200 in partial fulfillment of the requirements for the award of degree of Master of Science in Computer Science at the **<Name of the Dept. and College Address>.** This work is not submitted to any University for the award of any Degree / Diploma.

Name, Designation and Signature of INTERNAL GUIDE

HEAD OF THE DEPARTMENT

CERTIFICATE FOR STUDENTS WHO HAD DONE PROJECT IN THE INDUSTRY / ORGANISATION

This is to certify that it is a bonafide record of the Dissertation work entitled

"_____" done by STUDENT NAME , a student of M.Sc.(CS) at the **<Name of the Dept. and College Address>.** during the period 200 - 200 in partial fulfillment of the requirements for the Award of Degree of M.Sc.(CS). This work is not submitted to any University for the award of any Degree / Diploma. This work is carried out in (NAME OF THE ORGANISATION) with complete address.

INTERNAL GUIDE

HEAD OF THE DEPARTMENT

CERTIFICATE FROM INDUSTRY

To Date:

Head of the Department
College Address

CERTIFICATE OF PROJECT COMPLETION

This is to certify that _____
Has completed the project in our organization as per the particulars given below.

PERIOD :

PROJECT TITLE :

SOFTWARE TOOLS USED :

SIGNATURE AND STAMP

INDUSTRY / ORGANISATION
OFFICE SEAL

FORMAT OF M.Sc.(CS) DISSERTATION

The dissertation should be in the following format. Otherwise , the submission is rejected.

- Cover / Title Page
- 1st Page=Cover Page
- Certificate(Dept.)
- Submit to the H.O.D
- Acknowledgements
- Abstract
- Table of Contents with page numbers
- Rest of the Dissertation : follow the guidelines given
- References

  REFERENCES EXAMPLE:

  1. Dias, F.J.O. , "Truth-table verification of an iterative logic array, " IEEE Tans. On Computers, Vol. C-25, PP 605-613 , June 1976.
  2. "Signature analysis, " Hewiett-Packard Journal , Vol. 28, No. 9, May 1977.

  (1) (2) are to shown in text.
- Bibliography

  BIBLIOGRAPHY EXAMPLE:

  1. (Author) (Text Book) (Publisher), Year
- Appendix

PAGE FORMAT:	Paper Size	……	A4
	Left Margin	……	1 ½''
	Right Margin	……	1''
	Top Margin	……	1''
	Bottom Margin	……	1''
	Line Space	……	1 ½"

Font-Times New Roman 12
Page Numbers at the Bottom Centre
3 Hard Bound copies are to be submitted

Students must have regular interaction with the project guide. Progress is to be submitted through guide every two months to the Department. Project submission is not allowed for those students who fail to give progress report ON TIME (every two months)

Starting Day of Instruction 4th Sem. M.Sc.(CS).	} {	
Last Day of Instruction 4th Sem M.Sc.(CS).	} {	As specified
Last Day of M.Sc.(CS) Project Submission	} {	by H.O.D.
Commencement of M.Sc.(CS) Project Examination	} {	

NOTE: The internal guide must be available during Viva-Voce Examination of the concerned student(s).

RESENTATION MATERIAL FOR PROJECT VIVA:

Each student has to attend viva with not less than 20 and not more than 20 PPT slides covering major key points of the work. Results of the Project work should be demonstration on a PC. Soft copy of the code must be kept in the folder attached to the last cover page. Projects without code cannot be accepted.

M.Sc.(CS) PROJECT GUIDE LINES

The purpose of this note is to describe how to organize the written Dissertation submitted as partial fulfillment of your M.Sc.(CS) Degree.

The distinguishing mark of a dissertation is an original contribution to knowledge. The dissertation is a formal document whose sole purpose is to prove that you have made an original contribution to knowledge. Failure to prove that you have made such a contribution generally leads to failure.

To this end, your dissertation mush show two important things :

1. You have identified a worthwhile problem, which has not been previously solved.
2. You have answered the question.

Your contribution to knowledge generally lies in your solution or answer.

The sole purpose of the dissertation is to prove that you have made an original and useful contribution to knowledge. The examiners need answers to the following questions:

- What is this student's research question?
- Is it a good question? (has it been answered before> is it a useful question to work on?)
- Did the student convince me that the question was adequately answered?
- Has the student made an adequate contribution to knowledge?

To prove the originality and value of your contribution, you must present a through review of the existing literature on the subject, and on closely related subjects. Then, by making direct reference to your literature review, you must demonstrate that your question.

(a) Has not been previously answered, and
(b) Is worth answering

Describing how you answered the question is usually easier to write about, since you have been intimately involved in the details over the course of your studies.

A Generic Dissertation Skeleton

1. INTRODUCTION

This is a general introduction to what the dissertation is all about-it is not just a description of the contents of each section. Briefly summarize the question (You will be stating the question in detail later), some of the reasons why it is a worthwhile question, and perhaps give an overview of you main results. This is a birds-eye view of the answers to the main questions answered in the dissertation (see above).

2. BACKGROUND INFORMATION

A brief section giving background information may be necessary, especially if your work spans two or more traditional fields. That means your readers may not have any experience with some of the material needed to follow your dissertation, so you need to give it to them. A different title than that given above is usually better; e.g., “A Brief Review of Frammis Algebra”.

3. REVIEW OF THE STATE OF THE ART

Here you review the state of the art relevant to your dissertation. Again, a different title is probably appropriate; e.g., “State of the Art in Zylon Algotithm”. The idea is to present (critical analysis a little bit later) the major ideas in the state of the art right up to, but not including, your own personal brilliant ideas.

You organize this section by idea, and not by author or by publication. For example if there have been three important main approachs to Zylon Algorithms to date, you might organize subsections around these three approaches, if necessary:

3.1. Interactive Approximate of Zylons
3.2. Statistical Weighting of Zylons
3.3. Graph-theoretic Approaches to Zylon Manipulation

4. PROBLEM STATEMENT

Engineering dissertation tend to refer to a “problem to be solved.

- A concise statement of the question that you dissertation tackles
- Justification, by direct reference to section3, that you question is previously unanswered.
- Discussion of why it is worthwhile to answer this question.

Item 2 above is where you analyze information which you presented in Section 3. For example, may be your problem is to “develop a Zylon algorithm capable of handling very large scale problems in reasonable time” (you would further describe what you mean by “large scale” and “reasonable time” in the problem statement). Now in your analysis of the art ou would show how each class of current approaches fails (i.e. can handle only small problems, o taks too much time). In the last part of this section you would explain why having a large scale fast Xylon algorithm is useful; e.g., by describing applications where it can be used.

5. DESCRIBING HOW YOU SOLVED THE PROBLEM

This part of the dissertation is much more free-form. It may have one or several sections and subsections. But it all has only one purpose: to convince the examiners that you solved the problem that you set for yourself in Section 4. So show what you did that is relevant to solving the problem: if there were blind all eye and dead ends, do not include these.

6. CONCLUSIONS

You generally cover three things in the Conclusions section, and each of these usually merits a separate subsection:

a) Conclusions
b) Summary of Contributions
c) Future Research

7. REFERENCES

The list of references is closely tied to the review of the state of the art given in section 3. most examines scan your list of references looking for the important for the important works in the field, so make sure they are listed and referred to in section 3. Truth be known, most examiners also look for their own publications if they are in the topic area of the dissertation, so list these too. Besides, reading your examiner's papers usually gives you a clue as to the type of questions they are likely to ask.

All references given must be referred to in the main body of the dissertation. Note the difference from a Bibliography, which may include works that are not directly referenced in the dissertation. Organize the list of references either alphabetically by author surname (preferred), or b order of citation in the dissertation.

8. APPENDICES

What goes in the appendices? Any material which impedes the smooth development of your presentation, but which is important to justify the results of a dissertation. Generally it is material that is of too nitty-gritty a level of detail for inclusion in the main body of the dissertation, by which should be available for perusal by the examiners to convince them sufficiently. Examples include program listings, immense tables of data, lengthy mathematical proofs or derivations, etc.,

A NOTE ON COMPUTER PROGRAMS AND OTHER PROTOTYPE

The purpose of your dissertation is to clearly document an original contribution to knowledge. You may develop computer programs, prototypes, or other tools as a means of providing your points, but remember, the dissertation is not about the tool, it is about the contribution to knowledge. Tools such as computer programs are fine and useful products, but you can't get an advanced degree just for the tool. You must use the tool to demonstrate that you have made an original contribution to knowledge; e.g., through its use, or ideas it embodies.

HOW TO WRITE AN ABSTRACT

ABSTRACT

Because on-line search databases typically contain only abstracts, it is vital to write a complete but concise description of your work to notice potential readers into obtaining a copy of the full paper. This article describes how to write a good computer architecture abstract for both conference and journal papers. Writers should follow a checklist consisting of motivation, problem statement, approach, results, and conclusions. Following this checklist should increase the chance of people taking the time to obtain and read your complete paper.

INTRODUCTION

Now that the use of on-line publication databases is prevalent, writing a really good abstract has become even more important than it was a decade ago. Abstracts have always served the function of "selling" your work, But now, instead of merely convincing the reader to keep reading the rest of the attached paper, an abstract must convince the reader to leave the comfort of an office and go hunt down a copy- of the article from a library (or worse , obtain one after a long wait through inter-library loan). In a business context , an "executive summary" is often the only piece of a report read by the people who matter ; and it should be similar in contest if not tone to a journal paper abstract.

CHECKLIST : PARTS OF AN ABSTRACT

Despite the fact that an abstract is quite belief , it must do almost as much work as the multipage paper that follows it. In a computer architecture paper, this means that it should in most cases include the following sections. Each section is typically a single sentence , although there is room for creativity. In particular , the parts may be merged or spread among a set of sentences. Use the following as a checklist for your next abstract.

- MOTIVATION

  Why do we are about the problem and the results? If the problem is not obviously "interesting" it might be better to put motivation first; but if your work is incremental progress on a problem that is widely recognized as important, then it is probably better to put the problem statement first to indicate which piece of the larger problem you are breaking off to work on. This section should include the importance of your work, the difficulty of the area , and the impact it might have if successful.

- PROBLEM STATEMENT

  What problem are you trying to solve ? What is the scope of your work (a, generalized approach , or for a specific situation)? Be careful not to use too much jargon. In some cases it is appropriate to put the problem statement before the motivation, but usually this only works if most readers already understand I why the problem is important.

- RESULTS

  What's answer? Specifically , most good computer architecture papers conclude that something is so many percent faster , cheaper, smaller, or otherwise better that something else. Put the result there, in numbers. Avoid vague, hand-waving results such as "very", "small", or "significant" . If you must be vague, you are only given license to do so when you can talk about orders-of-magnitude improvement . There is a tension here in that you should not provide numbers that can be easily misinterpreted, but on the other hand you do not have room for all the caveats.

- CONCLUSIONS

  What are the implications of your answer? Is it going to change the world (unlikely), be a significant "win", be a nice hack , or simply serve as a road sign indicating that this path is a waste of time (all of the previous results are useful). Are , your results general, potentially generalizable, or specific to a particular case?

OTHER CONSIDERATIONS

An abstract must be a fully self-contained , capsule description of the paper. It can't assume (or attempt to provoke) the reader into flipping through looking for an explanation of what is meant by some vague statement. It must make sense all by itself. Some points to consider include:

* Meet the word count limitation. If your abstract runs too long, either it will be rejected or someone will take a chainsaw to it to get it down to size. Your purposes will be better served by doing the difficulty task of cutting yourself, rather that leaving it to someone else who might, be more incrested in meeting size restrictions than an representing your efforts
in the best possible manner. An abstract word limit of 150 to 200 words is common.

- Any major restrictions or limitations on the results should be stated, if only by using "weasel-words" such as "might" , "could", "may" and "seem".

- Think of a half-dozen search phrases and keywords that people looking for your work might use. Be sure that those exact phrases appears in your abstract , so that they will turn up at the top of a search result listing.

- Usually the context of a paper is set by the publication it appears in (for example , IEEE computer magazines articles are generally about computer technology). But , if you r paper appears in a some what un-traditional venue, be sure to include in the problem statement the domain or topic area that it is really applicable to.

- Some publications request "keywords" . These have two purposes. They are used to facilitate keyword index searches. Which are greatly reduced in importance now that on-line abstract text searching is commonly used. However , they are also used to assign papers to review committees or editors, which can be extremely important to your fate. So make sure that the keyword's you pick make assigning your paper to a review category obvious (for example, if there is a list of conference topics, use your chosen topic area as one of the keyword tuples).

(A typical specimen of table of contents)
<Font Style Times New Roman>

TABLE OF CONTENTS
