

**Proposed Syllabus for  
M.Sc. ( Computer Science)  
(To be implemented from year 2008-2009)**

**Course Structure** – The entire course is a Two year and Four semester course. For three semesters there will be four theory courses and one Lab course. The last semester will be Industrial training/Institutional project. The Lab course of semester I and semester III and one theory course each from semester II and Semester III are departmental Elective courses.

Every theory paper is evaluated for 80 marks externally and for 20 marks internally except departmental theory courses which are internally evaluated for 100 marks

The lab course is divided into project work and assignments and the break up is given below for each lab course.

The Industrial Project will be graded . The grades are O, A+, A, B+, B, C+ , C and D. D grade indicates failure

**Semester I :**

- CS11-101-Paper I : Principles of Programming Languages
- CS11-102-Paper II : Object Oriented Software Engineering
- CS11-103-Paper III : Distributed Database Concepts
- CS11-104-Paper IV : Design and Analysis of Algorithms
- CS11-105-Paper V : Lab course (Departmental)

**Semester II :**

- CS12-201-Paper I : Advanced Networking Concepts
- CS12-202-Paper II : UNIX Internals
- CS12-203-Paper III : Software Architecture
- CS12-204-Paper IV :XML .NET Programming (Departmental)
- CS12-110-Paper V : Lab course (University)

**Semester III :**

- CS23-301-Paper I : Software Metrics & Project Management
- CS23-302-Paper II : Mobile Computing
- CS23-303-Paper III : Information System Security
- CS23-304-Paper IV : Elective (Departmental)
- CS23-305-Paper V : Lab course (Departmental)

**Semester IV :**

- CS24-401-Industrial Training project

## CS-11-101: Principles of Programming Languages

- 1. Introduction** [2]
- The Art of Language Design
  - The Programming Language Spectrum
  - Why Study Programming Languages?
  - Compilation and Interpretation
  - Programming Environments
- 2. Non-Imperative Programming Models: Functional, Logic Languages** [12]
- Common LISP**
- Basic LISP Primitives ( FIRST, REST, SETF, CONS, APPEND, LIST, NTHCDR, BUTLAST, LAST, LENTH, REVERSE, ASSOC)
  - Procedure definition and binding, DEFUN, LET
  - Predicates and Conditional, EQUAL, EQ, EQL, =, MEMBER, LISTP, ATOM, NUMBERP, SYMBOLP, NIL, NULL, IF, WHEN, UNLESS, COND, CASE
  - Procedure Abstraction and Recursion
  - Properties and Arrays
- Turbo Prolog**
- Introduction, FACTs, Objects and Predicates, Variables, Using Rules, Controlling execution fail and cut predicates,
  - Input, Output, Recursion, Operators, Compound Objects, Dynamic Databases, Using LISTs, String Operation, String Operation,
- 3. Names, Scopes, and Bindings** [5]
- The Notion of Binding Time
  - Object Lifetime and Storage Management : Stack-Based Allocation, Heap-Based Allocation, Garbage Collection
  - Scope Rules Static and Dynamic Scope, Symbol Tables, Association Lists and Central Reference Tables
  - The Binding of Referencing Environments Subroutine Closures, First- and Second-Class Subroutines, Naming-Related Pitfalls in Language Design, Scope Rules, Separate Compilation
- 4. Control Flow** [6]
- Expression Evaluation : Precedence and Associativity, Assignments, Ordering Within Expressions, Short-Circuit Evaluation

- Structured and Unstructured Flow
- Sequencing
- Selection
  - Short-Circuited Conditions, Case/Switch Statements
- Iteration
  - Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops
- Recursion
  - Iteration and Recursion, Applicative- and Normal-Order Evaluation
- Nondeterminacy

## **5. Data Types [6]**

- Type Systems
  - The Definition of Types, The Classification of Types
- Type Checking
  - Type Equivalence, Type Conversion and Casts, Type Compatibility and Coercion, Type Inference
- Records (Structures) and Variants (Unions)
  - Syntax and Operations , Memory Layout and Its Impact, Variant Records
- Arrays, Strings, Sets, and Lists
- Pointers and Recursive Types
  - Syntax and Operations, Dangling References, Garbage Collection
- Equality Testing and Assignment

## **6. Subroutines and Control Abstraction [6]**

- Review of Stack Layout
- Calling Sequences
  - Case Study: C on MIPS, In-Line Expansion
- Parameter Passing
  - Parameter Modes, Special-Purpose Parameters, Function Returns
- Generic Subroutines and Modules
- Exception Handling
  - Definition of Exceptions, Exception Propagation, Example: Phrase Level Recovery in a RDP, Implementation of Exceptions
- Coroutines
  - Stack Allocation, Transfer

## **7. Data Abstraction and Object Orientation [6]**

- Encapsulation and Inheritance
  - Modules, Classes, Type Extensions
- Initialization and Finalization
  - Choosing a Constructor, References and Values, Execution Order, Garbage Collection
- Dynamic Method Binding

Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup and Related Concepts

- Multiple Inheritance  
Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance

## 8. Concurrency

[5]

- Concurrent Programming Fundamentals  
Communication and Synchronization, Languages and Libraries, Thread Creation Syntax  
Implementation of Threads
- Shared Memory  
Busy-Wait Synchronization, Scheduler Implementation, Scheduler-Based Synchronization, Implicit Synchronization
- Message Passing  
Naming Communication Partners, Sending and Receiving, RPC

### Text Books:

1. Programming Language Pragmatics by Michael J. Scott; Morgan Kaufmann Publishers, An Imprint of Elsevier, USA
2. Introduction to Turbo Prolog by Carl Townsend
3. LISP 3<sup>rd</sup> edition  
By Patrick Henry Winston & Berthold Klaus Paul Horn (BPB)

### Reference Books

1. Programming languages - Design and implementation  
By Terrence W. Pratt and Marvin V. Zelkowitz.
2. LISP Primer, Colin Allen, and Maneesh Dhagat (Also Available OnLine  
<http://mypage.iu.edu/~colallen/lp/node1.html> )

## CS 11-102 : Object Oriented Software Engineering

- 1. Object Oriented Concepts and Principles [5]**
  - a. What is object Orientation  
Introduction ,object, classes and instance, Polymorphism, Inheritance.
  - b. Object oriented system development  
Introduction  
Function/data methods  
Object oriented analysis  
Object oriented construction  
Object oriented testing
  - c. Identifying the elements of an object model  
Identifying classes and objects  
Specifying the attributes  
Defining operations  
Finalizing the object definition
- 2. Introduction to UML [4]**
- 3. Basic Structural Modeling [4]**

Classes , Relationship, Common mechanism, Diagrams, Class diagram
- 4. Advanced structural Modeling [5]**

Advanced classes, Advanced Relationship, Interface, Types and Roles.  
Packages, Object Diagram
- 5. Basic Behavioral Modeling [5]**

Interactions  
Use cases, Use Case Diagram  
Interaction Diagram  
Activity Diagram  
State chart Diagram
- 6. Object Oriented Analysis [8]**

Iterative Development and the Unified process  
Inception  
Understanding requirements  
Use Case Model From Inception to Elaboration  
Elaboration
- 7. Object Oriented Design [6]**
  - a. The Booch Method, The Coad and Yourdon Method and Jacobson Method and Raumbaugh method

- b. The Generic components of the OO Design model
- c. The system design Process
  - a. Partitioning the analysis modal
  - b. Concurrency and sub system allocation
  - c. The task management component
  - d. The data management component
  - e. The resource management component
  - f. Inter sub system communication
- d. Object design process

**8. Architectural Modeling [6]**

Component , Components Diagram  
 Deployment Diagram  
 Collaboration Diagram

**9. Testing [5]**

- a. Object oriented Testing Strategies
- b. Test case design for OO Software
- c. Inter Class Test case design

**10. case studies [4]**

**References**

1. Software Engineering By Pressman
2. The Unified Modeling Language User Guide By Grady Booch, James Raumbaugh and Ivar Jacobson.
3. Object Oriented Software Engineering By Ivar Jacobson
4. Applying UML and Patterns By Craig Larman

## CS11-103: Advanced Database Concepts

- 1. DATABASE TUNING** [6]
  - physical database design & tuning
- 2. INTRODUCTION TO DDBMS** [2]
  - Distributed data processing
  - Distributed database systems (DDBS)
  - Promises of DDBMS
  - Completing factors and problem areas
- 3 DISTRIBUTED DBMS ARCHITECTURE** [4]
  - DBMS standardization
  - Architectural models for DDBMS
  - DDBMS architecture and Global directory issues.
- 4. DISTRIBUTED DATABASE DESIGN** [6]
  - Alternative design strategies
  - Distributed design issues
  - Fragmentation and allocation.
- 5. OVERVIEW OF QUERY PROCESSING** [4]
  - Query processing problems
  - Objectives of query processing
  - Complexity of relational algebra operators
  - Characterization of query processors
  - Layers of query processing
- 6. QUERY DECOMPOSITION & DATA LOCALIZATION** [ 6]
  - Query decomposition
  - Localization of distributed data
- 7. OPTIMIZATION OF DISTRIBUTED QUERIES** [8]
  - Query optimization
  - Centralized query optimization
  - Join ordering in fragment queries.
- 8. TRANSACTION MANAGEMENT** [ 2]
  - Destination of a transaction
  - Problems of transactions

- Type of transactions
- Architecture revisited

## **9 DISTRIBUTED CONCURRENCY CONTROL**

[10]

- Serilizability theory
- Taxonomy of concurrency control mechanisms
- Locking-based concurrencyControl algorithms
- Timestamp-based concurrency control
- Optimistic concurrency control
- Deadlock management
- Relaxed concurrency

## **10.Distributed DBMS reliability**

[8]

- Reliability concepts & measures
- Failures & fault tolerance in distributed systems
- failures in DDBMS
- Local reliability protocols
- Distributed reliability protocols
- Dealing with site failures
- Network partitioning.

## **TEXT BOOK**

1. Principles of Distributed Database Systems; 2nd Edition  
By M. Tamer Ozsü and Patrick Valduriez Publishers: Person Education Asia  
ISBN: 81-7808-375-2
2. Database systems (2<sup>nd</sup> edition) By Raghuramakrishnan and Johannes

## **Referencess**

1. Distributed Database; Principles & Systems By Stefano Ceri and Giuseppe Pelagatti  
Publications: McGraw-Hill International Editions  
ISBN: 0-07-010829-3

## CS-11-104: Design And Analysis of Algorithms

### 1. Analysis

Algorithm definition, space complexity, time complexity, worst case –best case – average case complexity, asymptotic notation(  $O$ ,  $\Omega$ ,  $\theta$  notation), sorting algorithms (insertion sort, heap sort) , sorting in linear time, searching algorithms, recursive algorithms ( Tower of Hanoi , Permutations). [6]

### 2. Design strategies

**Divide and conquer**-control abstraction, binary search, merge sort, Quick sort, Strassen's matrix multiplication [6]

**Greedy method**- knapsack problem, job sequencing with deadlines, optimal storage on tapes, optimal merge patterns, Huffman coding.[6]

**Dynamic programming**- matrix chain multiplication, longest common subsequence, string editing, 0/1 knapsack problem, Traveling salesperson problem. [8]

**Backtracking**- General method, 8 Queen's problem, Sum of subsets problem, graph coloring problem, Hamiltonian cycle. [2]

**Branch and Bound Technique** : FIFO, LIFO, LCBB, TSP problem, 0/1 knapsack problem [4]

### Graph algorithms

BFS, DFS, classification of edges, Topological sort, Strongly connected components, minimal spanning trees, Kruskal and Prim's algorithm, single source shortest paths, Dijkstra's algorithm, Bellman-ford algorithm, all paths shortest paths, Floyd-Marshall algorithm. Flow networks, Ford-Fulkerson method, Maximum bi-partite matching [12]

### Problem classification

Nondeterministic algorithm, The class of P, NP, NP-hard and NP- Complete problems, significance of Cook's theorem [2]

### Text Books

- 1) T. Cormen, C. Leiserson, & R. Rivest, Algorithms, MIT Press, 1990
- 2) Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran, Computer Algorithms ,Galgotia.

### References

- 1) A. Aho, J. Hopcroft, & J. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974
- 2) Donald Knuth, The Art of Computer Programming (3 vols., various editions, 1973-81), Addison Wesley

## CS12 -201 Advanced Networking

- 1. Review of Basic Concepts [3]**
  - Network Architecture – Protocol Hierarchies, Layered model, services, interface
  - Reference Models
  - Underlying Technologies

LAN's (Ethernet, Token Ring, Wireless) , Point-to-Point WAN's , Switched WAN's (X.25, Frame Relay, ATM), Connecting devices , Addressing (Physical, Network, Transport)
- 2. The Internet Layer Protocols [4]**
  - IP- Datagram, fragmentation and reassembly,
  - ICMP- types of messages, error reporting, ICMP Package
- 3. Bootstrap and Autoconfiguration [2]**

BOOTP and DHCP
- 4. Routing Protocols [4]**
  - Interior and Exterior Routing – RIP, OSPF, BGP
  - Multicast Routing- Unicast , Multicast and Broadcast, Multicasting, Multicast trees,
- 5. The Transport Layer [10]**
  - The transport Service- Services provided, Service primitives, Sockets
  - Process-to-Process Communication – Port addresses
  - Elements of transport protocols – addressing, connection establishment, connection release, flow control and buffering, multiplexing, crash recovery
  - UDP – Introduction, Remote Procedure Call,
  - TCP – Service model, Protocol, frame format, connection establishment, release, connection management, Silly Window Syndrome-Nagle's algorithm, error control, congestion control, state transition diagram
- 6. Sockets and Client-Server Model [2]**
  1. Client-Server Model – Concurrency, Processes,

2. Socket Interface – Sockets, Byte Ordering, Socket System Calls, Connectionless and Connection Oriented applications
3. Implementation of Sockets ( C/Java etc)

## 7. The Application Layer [14]

1. DNS, Telnet and Rlogin, FTP, TFTP, SNMP, SMTP, World Wide Web( Client and Server Side, cookies, wireless web), Java and the Internet, Multimedia (streaming audio, Internet Radio, voice over IP - RTP , video standards), Real time traffic over the internet

## 8. Mobile IP [4]

2. Mobility, routing and addressing, characteristics, operation, foreign agent discovery , registration and communication, two crossing problem, communication with computers on the home front.

## 9. Introduction to Network Security [4]

3. Cryptography, Symmetric key algorithms, Public Key algorithms, Digital Signatures, Certificates, IPSec , Firewalls, Virtual Private networks, Network Address Translation, Authentication Protocols, Social Issues

### Texts

1. Behrouz Forouzan , *TCP/IP Protocol Suite*, Second Edition, Tata McGraw Hill
2. Andrew S Tanenbaum , *Computer Networks*, Fourth Edition, Prentice Hall
3. Douglas E. Comer , *Internetworking with TCP/IP*, Vol. 1, Principles, Protocols and Architecture Fifth Edition, Prentice Hall, 2000, ISBN 0-13-018380-6.
4. William Stallings, *Data and Computer Communications* , Seventh Edition, Pearson Education

### Supplementary useful references:

1. Douglas E. Comer , *Internetworking with TCP/IP, Vol. 2, Design , Implementation and Internals*, Prentice Hall Publisher.

2. *Internetworking with TCP/IP, Vol. 3, Client-server Programming and Applications* by Douglas E. Comer, Prentice Hall Publisher. (Excellent reference for distributed programming over TCP/IP networks)

3. Richard Stevens , *TCP/IP Illustrated*, Vol. 1. by, Addison Wesley (A very practical book with lots of useful network diagnostic tools and programs.)
4. Craig Hunt , *TCP/IP Network Administration* O'Reilly & Associates, Inc. (A must for network and system administrators dealing with internetworking.)
5. L. Peterson and B. Davie. Morgan , *Computer Networks: A Systems Approach* by Kaufmann Publishers Inc., 1996.
6. J. Kurose, K. Ross ``Computer Networking: A Top-Down Approach Featuring the Internet" Addison-Wesley, '00

## **CS-204 – General Lab IIB (Assignments)(Networking)**

### **Lab Assignments**

1. Design a LAN with a given set of requirements. The design should include topology, hardware and software requirements like cable, connectors, hubs/switches/bridges, interface cards along with a budget for the LAN. (The course instructor should give the requirements to the students) (**Compulsory**)

**Minimum two from the following. Assignments can be done in groups of (2/3).**

- 1.Design and Implement a GUI or text based network monitoring tool to record network statistics like packets sent and received, percentage errors, desktop grabbing, remote monitoring.
- 2.Design a simple wireless communication system between two computers.

### **ICMP :**

1. Define struct called stateinformation that contains all the local information that an ICMP module needs to access or modify.
2. Define a struct called icmppseudoheader that can hold entries for header needed to calculate the checksum for ICMP. Using this declaration, write a function called ICMPchsum to calculate checksum field for an ICMP packet. The function takes two arguments
  - i) pointer to ICMP packet
  - ii) pointer to header.

### **BootStrap and Autoconfiguration**

- 1.Create a header file to include all constants that you think are needed to implement BOOTP algorithm in C. ( use #define directive)
2. Create a header file to include all constants that you think are needed to implement DHCP algorithm in C. ( use #define directive)

### **Routing Protocols**

1. Write a C code to implement RIP

### **Transport Layer**

1. Write a program to simulate main module of TCP

### **Socket and Client Server Model**

1. Design a client server application for solving roots of a quadratic equation by making use of appropriate API's

### **Application Layer**

1. Implement FTP / TFTP / SNMP / SMTP
2. Design a chat application with the following features: Single/ Multiline messages, emoticons, single as well as multiple windows, conversation with single and multiple people.
3. Design and implement a firewall for your network. It should have the following facilities: blocking specific IP address, pages with specific content.

## CS12 –202 Unix Internals

- 1. Introduction to the kernel** **2 lectures**
  - Architecture of the Unix operating system
  - Introduction to system concept
  - Kernel data structure
  - System administration
  
- 2. The buffer cache** **3 lectures**
  - Buffer header
  - Structure of buffer pool
  - Buffer retrieval
  - Reading and writing disks block
  - Advantages and disadvantages
  
- 3. Internal representation of files** **5 lectures**
  - Inodes
  - Structure of a regular file
  - Directories
  - Conversion of pathname to an inode
  - Super block
  - Inode assignment to a new file
  - Allocation of disk block
  
- 4. System calls for the file system** **10 lectures**
  - Open
  - Read
  - Write
  - File and record blocking
  - Adjusting the position of file I/O – lseek
  - Close
  - File Creation
  - Creation of Special Files
  - Change directory and chage root
  - Change owner and change mode
  - Stat and fstat
  - Pipes
  - Dup
  - Mounting and unmounting file systems
  - Link
  - Unlink
  - File system naintenance

- 5. The structure of processes** **7 lectures**
- Process states and transitions
  - Layout of system memory
  - The context of a process
  - Saving the context of a process
  - Manipulating of a process address space
  - sleep
- 6. Process control** **10 lectures**
- Process creation
  - Signals
  - Process termination
  - Awaiting Process termination
  - Invoking other programs
  - The user-id of a process
  - Changing the size of a process
  - The shell
  - System boot and init process
- 7. Process scheduling and time** **4 lectures**
- Process scheduling
  - System calls for time
  - clock
- 8. Memory management policies** **5 lectures**
- Swapping
  - Demand paging
  - Hybrid system with swapping and demand paging
- 9. The I/o subsystems** **4 lectures**
- Driver interfaces
  - Disk driver
  - Terminal drivers

## **CS-204 – General Lab IIB (Assignments)(UNIX Internals)**

### **Lab Assignments**

1. Design a directory structure that improves the efficiency of searching for pathnames by avoiding the linear search
2. Implement free disk block list with a bitmap instead of linked list

3. Design a scheme that reduces the number of directory searches for file names by caching frequently used names.
  4. Redesign getblk and brelse where the kernel follows a FIFO scheme instead of LRU.
  5. Design algorithm for allocating and freeing memory page and page tables
  6. Many process can sleep on an address but the kernel may want to wakeup selected processes that receive a signal assuming that the signal mechanism can identify the particular processes, remodify the wakeup algorithm so that only one process is woken up on a sleep address instead of all the processes.
  7. Implement a new new system call newpgrp(PID, ngrp), that resets the process group of another process identified by the process ID PID to ngrp
  - 8 Implement a new system call nowait(PID) where PID identifies a child of the process issuing the call when issuing the callthe process informs the kernel that it will never wait for the child process to exit, so that the kernel can immediately cleanup the child process slot when the child dies
  9. Any one assignment on windows internal (Compulsory)
- (Note : These are the sample assignments. More assignments can be given. Each student is supposed to do a minimum of 5 assignments)

## CS12-203 Software Architecture

1. The Big Picture – How it all fits in? [2]
  - UML → The Notation
  - Process → Unified Process / Rational Unified Process inception, elaboration, construction, transition.
  - How various components fit in the life cycle
  - The artifacts at end of each process / discipline
  
2. Software Architecture: [2]
  - What Software Architecture is and what it isn't.
  - Why is architecture important?
  - Architectural structures and views
  
3. Architectural Styles : [3]
  - Architectural Styles
  - Pipes and Filters
  - Data Abstraction and Object – Oriented Organization
  - Event-Based, Implicit Invocation
  - Layered Systems
  - Repositories
  - Interpreters
  - Other familiar Architectures
  - Heterogeneous Architectures.
  
4. Patterns : [4]
  - What is a Pattern & Design Pattern.
  - What makes a Pattern. (GOF)
  - Describing Design Patterns.
  - Pattern Categories & Relationships between Patterns.
  - Organizing the Catalog.
  - Patterns and Software Architecture.
  
5. Study of Design Patterns : [186]
  - Creational Patterns-singleton, factory method, abstract factory
  - Structural Patterns-adapter, decorator, facade
  - Behavioral Patterns-iterator, observer, strategy, command and state (study of intent, applicability, participants, structure, collaboration and consequences)
  - GRASP(General Responsibility Assignment Software Patterns : Patterns for Assigning Responsibilities
    - Expert, Creator, High Cohesion, Low Coupling, Controller, Polymorphism, Pure Fabrication, Indirection, Don't Talk to Strangers.

## 6. Study of Frameworks :

[6]

- Frameworks as reusable chunks of architecture,
- The framework lifecycle, development using frameworks,
- Struts for Identify the MVC (Separation of layers)
- Configuration
- Declarative error handling
- Validation Framework
- Interaction with web application
- Case Study
- Use of Front controller & Service to worker patterns.
- Web Architectures
- Available
  - Baracudda, Webworks, Velocity, Struts etc.
- Selection of proper framework
- Comparing Frameworks.
- Advantages of Struts.

## 7. Components:

(5)

- Development using components, composition, components as units of deployment, different approaches to components (e.g. OMG, Microsoft, Sun), developing components.

## 8. Case Study ( struts)

(5)

- Take a Framework and find Patterns in the Frame work.
- Benefits of Patterns in the choosen Framework
- How Pattern interact in the selected Framework.

## Reference Books

- Design Patterns – Elements of Reusable Object-oriented Software By E. Gamma, Richard Helm, Ralph Johnson , John Vlissides (GoF)
- Struts By Chuck Canvass.
- Pattern – Oriented Software Architecture (POSA) Volume 1.  
By : Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal.
- Software Architecture in Practice. By Len Bass, Paul Clements, Rick Kazman.
- Applying UML and Patterns By Craig Larman.
- Software Architecture- Perspectives on an emerging discipline by Mary shaw and David Garlan

## CS-12-204-XML.NET programming Syllabus

1. Introduction to XML
2. How is XML used?
3. Rules of XML
4. XML Syntax
5. XML Declarations
6. XML tags
7. XML Document
  - Elements
  - Tags and attributes
  - Entity references
  - Comments
  - Processing instructions
  - CDATA sections
8. Well Formed XML Documents
9. XML DTD's
10. XML DOM
11. Using XML Parser
12. XSL

### .NET Syllabus

- Chp 1. The philosophy of .Net
  - Introducing building blocks of the .Net Platform
  - Overview of .Net Assemblies
  - Role of CIL
  - The role of .NET type metadata
  - Assembly Manifest
  - Understanding CTS, CLS, CLR
- Chp2 The C# Programming language
  - System.Console Class
  - Method Parameter modifiers
  - Value Types and Reference types
  - Boxing and Unboxing Operations
  - .Net Enumerations
  - System.Object
  - System Data Types
  - System.String Data Type
  - .Net Array types
- Chp3 Object-Oriented Programming with C#.ul style="list-style-type: none;">- C# Class Type
- C#'s Encapsulation services
- C#'s Inheritance support
- Programming for Containment/Delegation
- C#'s Polymorphic support

- Chp4 Understanding Object Lifetime
  - Understanding Generations
  - The System.GC type
  - Building finalizable objects
  - Building disposable objects
- Chp5 Exception Handling
  - Role of .NET exception handling
  - Configuring the state of Exception
  - System Level Exceptions
  - Application level Exceptions
- Chp6 Interfaces and Collections
  - Implementing interface in C#
  - Interfaces as parameters
  - Arrays of Interface type
  - Building Interface Hierarchies
- Introducing .NET Assemblies
  - Role, Format of .NET Assembly
  - Single-File, Multiple-File Assemblies
  - Private Assemblies
  - Shared Assemblies
- Chp7 Type Reflection, Late Binding, and Attribute-based programming
  - Necessity of Type Metadata
  - Understanding Reflection
  - Building custom metadata viewer
  - Understanding Late Binding
  - Understanding Attributed programming
- Chp8 Building multithreaded applications
  - Role of Thread Synchronization
  - The Asynchronous nature of delegates
  - The System.Threading.Thread Class
- Chp9 The System.IO Namespace
- Chp10. System.Windows.Forms
- Chp11. Database Access with ADO.NET
  - ADO.NET Data providers
  - The System.Data Types
  - Understanding Connected layer of ADO.NET
  - Understanding the Disconnected layer of ADO.NET
- Chp12. ASP.NET Web Pages and Web Controls
- Chp13. ASP.NET 2.0 Web Applications.

Reference:

1. Pro C# 2005 and the .NET 2.0 Platform – Andrew Troelson
2. CLR via C# - Jeffery Richter

## CS-23-301 Software Metrics & Project Management

1. What is Project? What is Project management? Project phases and project life cycle, organizational structure, Qualities of Project Manager. [4]
2. Project Management Components.  
Project Integration Management-Project plan development and execution, change controls, configuration management. [6]
3. Scope Management-Strategic planning ,scope planning,definition ,verification and control. [4]
4. Time management- Activity planning, schedule development and control. [2]
5. Cost Management- Cost estimation and Control. [2]
6. Quality Management- Quality planning and assurance. [2]
7. Human Resource Management- Organizational planning , staff acquisition. [2]
8. Communication Management-Information distribution , reporting. [2]
9. Risk Management-Risk identification,Quantification and control. [2]
10. Procurement Management- Solicitation, contract administration. [2]
11. Software Metrics- The scope of software metrics, software metrics data collection, analyzing software data, measuring size, structure, external attributes. [6]
12. Software Reliability- Measurement and prediction, resource measurement, productivity, teams and tools. [6]
13. Planning a measurement program.  
What is metrics plan?: Developing goals, questions and metrics.  
Where and When: Mapping measures to activities.  
How: Measurement tools.  
Who: Measurers , analyst, tools revision plans. [4]
14. Quality Standards – CMM, PSP/TSP [4]

### References

1. Information Technology Project Management  
By -Kathy Schwalbe.
2. Software Metrics A rigorous and practical approach  
By – Norman Fenton, Shari Lawrence Pfleeger.
- 3: Software Engineering By- Roger Presman.

# CS23-302: MOBILE COMPUTING

## Prerequisites

- Concepts of multiplexing and modulation
- Concepts of Networking
- Conversant with OS internals
- Familiar with event handling
- Web browsers
- Create and Compile Java Programs
- Brief History of wireless communication

## Objectives

- To familiarize the students with the buzz words and technology of mobile communication
- Understand the GSM architecture
- Understand the issues relating to Wireless applications

### 1. Introduction to Mobile Computing

**Lectures 2**

- i. Introduction and need for Mobile computing
- ii. Mobility and portability
- iii. Mobile and Wireless devices
- iv. Applications
- v. Brief History of wireless communication

**Book1: Mobile Comm. By Jochen Schiller**

### 2. Wireless Transmission

**Lectures 3**

- i. General Concepts of multiplexing and modulation
- ii. Spread Spectrum
- iii. Cellular Systems
- iv. Cellular Phone Array
- v. Mobile Phone Technologies (1G, 2G, 2.5G, 3G)

**Book1: Mobile Comm. By Jochen Schiller**

### 3. Medium Access Control Layer

**Lectures 4**

- i. Why specialized MAC?
  - hidden and exposed terminals
  - near and far terminals
- ii. General Concepts and comparison of SDMA, FDMA, TDMA , CDMA

**Book1: Mobile Comm. By Jochen Sciller**

### 4. Global System for Mobile Comm.

**Lectures 9**

- i. Mobile Services (Bearer, Tele-and-supplementary services)
- ii. System Architecture
  - Radio subsystem
  - Network and switching subsystem
  - Operation subsystem
- iii. Protocols
  - Localization and calling
  - Handover
- iv. Value Added Services
  - SMS  
Architecture, Mobile Originated and Mobile Terminated procedures
  - Cell Broadcast Service  
Architecture, Message Transfer Procedure
  - MMS  
Architecture, Protocol framework, Message Transfer Procedure
  - Location Services  
Logical Reference Model, Control Procedures, Network Architecture, determination of Location Information, Location based services
- v. GPRS  
**Book1: Mobile Comm. By Jochen Schiller**  
**Book5:2G Mobile Networks: GSM and HSCSD By Nishit Narang and Sumit Kasera**

## 5. Mobile IP

Lectures 12

- i. Goals, assumptions and requirements
- ii. Entities and terminologies
- iii. Agent Discovery
- iv. Registration
- v. Tunneling and encapsulation
- vi. Reverse Tunneling
- vii. IPv6
- viii. IP micro-mobility support – Cellular IP, Hawaii, Hierarchical mobile IPv6
- ix. Mobile Routing :  
Destination sequence distance Vector, Dynamic Source Routing, Alternative Matrix, Adhoc Routing Protocols -Flat, Hierarchical, Geographic-position-assisted  
**Book1: Mobile Comm. By Jochen Schiller**

## 6. Mobile TCP

Lectures 6

- i. Traditional TCP
  - Congestion Control, Slow start, Fast retransmit / Fast recovery
  - Implications on mobility
- ii. Classical TCP improvements
  - Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit / Fast recovery, Transmission / Timeout freezing, Selective Retransmission, Transaction oriented TCP
- iii. TCP over 2.5/3G wireless networks  
**Book1: Mobile Communications By Jochen Schiller**

## 7. Wireless Application Protocol

Lectures: 7

- i. Architecture
- ii. Wireless datagram protocol
- iii. Wireless transport layer security
- iv. Wireless transaction protocol
- v. Wireless session protocol
- vi. Wireless application environment
- vii. WML
- viii. WML Scripts
- ix. Push Architecture
- x. Push – Pull Services

**Book1: Mobile Communications by Jochen Schiller**

## 8. Platform/Operating Systems

Lectures 1

- i. Palm OS
- ii. Windows CE
- iii. Embedded Linux
- iv. J2ME (Introduction)
- v. Symbian (Introduction)
- vi. File Systems (Book1)

**Book2: Pervasive Computing**

**Book1: Mobile Comm. By Jochen Schiller**

## 9. Java for Wireless Devices

Lectures 1

- i. Setting up the development environment
- ii. Basic Data types, Libraries (CLDC, MIDP)  
**Any J2ME book**

## 10. UI Controls

Lectures 3

- i. Displayable and Display
  - Image
  - Events and Event Handling

- List and choice
- Text box
- Alerts

**Any J2ME book**

### 11. Persistent Storage

**Lectures 2**

- i. Record Stores
- ii. Records
- iii. Record Enumeration

**Any J2ME book**

### 12. Network MIDlets

**Lectures 2**

- i. The Connection Framework
- ii. Connection Interface
- iii. Making a connection using HTTP
- iv. Using datagram connection

**Any J2ME book**

### 13. Wireless Messaging

- i. Architecture for Messaging application
- ii. Messaging API
- iii. Types of applications
- iv. Pro's and con's of messaging

**References:** <http://java.sun.com/products/wma>  
<http://forum.nokia.com>

### Books

1. Mobile Communications Jochen Schiller, Pearson Education, 2<sup>nd</sup> Edition
2. Pervasive Computing Technology and Architecture of Mobile Internet Applications  
Jochen Burkhardt, Dr. Horst Henn, Steffen Hepper, Klaus Rintdorff, Thomas Schack , Pearson Education
3. Wireless Java Programming with J2ME  
Yu Feng and Dr, Jun Zhu , Techmedia Publications, 1<sup>st</sup> edition
- Books
4. Complete Reference J2ME
5. Mobile Networks GSM and HSCSD  
Nishit Narang, Sumit Kasera, TataMcGrawHill
6. Mobile Computing  
Asoke K Talukdar, Roopa R. Yavagal, TataMcGrawHill

## CS23 - 303 Information Systems Security

### Objectives of the Course:

1. To enable students to get sound understanding of Info-Sys-Security, Net-Security, Cryptography.
2. To equip with knowledge and skills necessary to support for their career in Information Security.
3. To develop attitude and interest along with necessary knowledge and skills among the students to encourage them to do further academic studies / research in this area, after the completion of their M.Sc. Course.

#### 1. Conceptual foundation of Information Systems Security:

- Concepts and Terminology: Threats, Attacks, Vulnerabilities, Risks, Risk Assessment and Mitigation, Security – Confidentiality, Integrity, Availability, Identification, Authentication, Authorization, Accountability, Privacy [3]
- 2. Cryptography: Techniques, Mathematical foundation, Stream Ciphers, Block Ciphers, Cryptanalysis. [2]
- 3. Symmetric / Secret Key Encryption: Algorithm Types and Modes, DES (Data Encryption Standard), Double DES, Triple DES, AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish, RC5 [6]
- 4. Public Key Encryption: Principles of public key crypto-systems, mathematical foundation, RSA algorithm, key management, Deffie-Hellman key exchange, Elliptic curve cryptography, Digital Signatures using DSA (Digital Signature Algorithm), DSS (Digital Signature Standard) and RSA [8]
- 5. Message Integrity techniques: MD5, SHA [2]
- 6. PKI (Public Key Infrastructure) and Trust Hierarchy, Digital Certificates, transaction certificates [4]
- 7. Authentication techniques: passwords, pass-code, pass-phrase, challenge-response, biometrics-based registration and authentication, Kerbores (04 lectures)
- 8. Internet Security protocols – SSL/TLS , TSP, SET, 3 – D Secure protocol, Electronic money, email security (PGP, PEM, S/MIME), IP Security: IPSec, VPN [12]

9. Server Security : Concepts, Design and Implementation of Firewalls, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), etc. This should also include detailed study of at least one free Firewall, IDS, IPS products with demonstrations [6]
10. Virus Threats including Network Viruses, Worms, etc. [2]
11. Data Hiding and Steganography [1]

For Internal Evaluation

12. Security and Audit implementation in various Operating Systems (at least one Desktop OS and at least one Network OS should be discussed)
13. Security and audit implementation in RDBMS (at least one RDBMS product should be discussed)

Recommended Readings (Text and Reference):

1. Atul Kahate, " Cryptography And Network Security" TMH
2. William Stallings, " Cryptography And Network Security" Prentice Hall / Pearson Education