CS 6329 Object-Oriented Software Engineering

Student Learning Objectives/Outcomes:
- Develop models using the UML notation;
- Apply an iterative, agile process;
- Analyze requirements with use cases;
- Create domain models;
- Relate analysis and design artifacts;
- Design object solutions with patterns and architectural layers;
- Apply concepts to a semester-long software engineering project;
- Document and present project deliverables;
- Use an advanced CASE tool.

Textbook:
List of Topics:

- Properties of Poisson streams of customer arrivals.
- Analysis and performance figures of the M/M/1 queue.
- Continuous parameter Markov chains.
- Single state dependent (continuous time) Markovian queueing systems.
- Various applications of such state dependent cases in computer systems and data communication networks.
- Generalized Little's result for multiple non-FIFO queues.
- Development and analysis of Markov chains for simple priority queues.
- Developments of Pollaczek-Khinchin mean value formula for the M/G/1 queue.
- Applications.
- Development of discrete parameter Markov chains for discrete time queues.
- Analysis of discrete parameter Markov chains.
- Evaluation of performance figures.
- Applications of discrete time queues in computer systems and data networks (such as, for examples, cross-bar and simple multistage switches).
- Product form solutions for networks of continuous time open and closed
- Markovian queues (unlimited buffer, state independent service rates).
- Convolution algorithm and Mean Value Analysis techniques for such closed queueing networks.

Type of Questions:

- Questions will be combinations of theoretical development, analysis of given systems, development of appropriate models and follow up analysis starting from verbal descriptions of physical systems. In most cases, students should attempt to solve problems from fundamental principles rather than trying to remember and apply formulae for various special cases.
- A set of helpful formulae, etc. (such as the Pollaczec-Khinchin mean value formula and the MVA algorithm) will be supplied along with the question paper.
- The following list of references include the commonly used text book, other reference books on queues, and a sample of books on Probability Theory. Students are responsible for correcting errors in the reference material.

Textbook:

Other References:

- L. Kleinrock, Queueing Systems, Volume 1, Theory. Wiley, 1975
CS 6360 Database Design

Topics:

• Database System Concepts and Architecture (Data models, Schemas, Instances, Database architecture, classification)
• Entity-Relationship (ER) model, ER diagrams
• The Enhanced Entity-Relationship (EER) model, EER Diagrams
• Relational Data Model, Relational algebra, SQL
• Relational Database Design by ER/EER-to-Relational Mapping
• Database Design Theory and Normalization (Basics of Functional Dependencies and Normalization for Relational Databases; Algorithms for Relational Database Schema design)
• Query processing and optimization
• Transaction processing concepts and theory
• Concurrency Control Techniques
• Database Recovery Techniques

Textbook:

• "Fundamentals of database systems" by Elmasri and Navathe:
CS 6362 Software Architecture and Design

Topics:
- Introduction to Software Architecture Classical
- Module Interconnection
- Languages Abstract DataTypes and Objects Module
- Decomposition Issues
- Data Flow
- Repositories Events
- Process Control
- JavaBeans
- Client Server

Middleware:
- CORBA, OLE/DCOM, J2EE/J2ME, .Net Patterns

Main Reference:

Articles:
- Advanced Design Patterns. Re-use

Representation:

• P. Clements, Comparing the SEI’s Views and Beyond Approach for Documenting Software Architectures’ with ANSI-IEEE 1471-2000
• Pengcheng Zhang, Henry Muccini and Bixin Li, “A classification and comparison of model checking software architecture techniques”, 2009.

Books:

4. Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley, 1994.

CS 6363 Design and Analysis of Computer Algorithms

The exam will test knowledge of:

1. Major techniques for algorithm design (as listed below);
2. Methods to prove algorithm correctness and to analyze its running time;
3. Basic knowledge of NP-Completeness.

NOTE:
You should know more than just the algorithms; you are responsible for proving correctness, including all necessary supporting lemmas, and are responsible for proving the correctness of any statements about the asymptotic running times. In addition, you should know the stated subject matter well enough to enable you to provide solutions for closely related questions.

Most topics (and knowledge) required are in the CS6363 textbook:
Introduction to algorithms, Second edition, Cormen, Leiserson, Rivest and Stein.

General topics:

- Introduction, recurrences and Master Theorem (Theorem 4.1, the proof is not required)
- Divide-and-Conquer algorithms
- Linear time median selection algorithm (Section 9.3, pp. 189-192)
- Closest pair of points in the plane (Section 33.4, pp. 957-961)
- Permutation networks (Problem 27-3, page 722)
- Sorting Networks (Chapter 27)
- Note: students should be able to design divide-and-conquer algorithms for various problems beside those mentioned above
- Dynamic Programming
- Matrix Chain Order (Section 15.2, pp. 331-338)
- Longest Common Subsequence Algorithm (Section 15.4, pp. pp. 350-355.)
- All pairs shortest paths (Section 25.2, pp. 629-634)
- 0/1-knapsack problem (Problem 16.2-2, page 384)
- Greedy Method
- Huffman's code algorithm (Section 16.3, pp. 385-392)
- Minimum spanning tree (Chapter 23)
- Single Source Shortest Paths (e.g. Dijkstra's algorithm) (Chapter 24, up to page 601)
- Maximum flow (Chapter 26, up to page 668)
- Graph algorithms (Chapter 22)
- NP-Completeness (Chapter 34, specifically 3SAT, VERTEX COVER, INDEPENDENT SET,
- CLIQUE, 3COLOR, HAMILTON CIRCUIT (both directed and undirected), as well as definitions and properties of polynomial time reducibilities.)
- Linear programming: (Chapter 29, pp. 770-789 and pp. 804-807.)
CS 6364 Artificial Intelligence

Text:

Problem solving by search:

Uninformed (Blind) Search and Heuristic (Informed) Search

Problem formulation; Uninformed search strategies: Depth-First Search, Breadth-First Search, Uniform-Cost Search, Iterative-Deepening.

Informed Search strategies: Greedy Best-First Search, A*, IDA*.

Heuristic Functions: heuristic domination, inventing admissible heuristics.

Adversary Search (Game Trees)

How to design computer programs that play games intelligently. The MIN/MAX and the ALPHA/BETA-Pruning algorithms, their complexity and efficient implementations.

Knowledge Representation

Propositional logic. Syntax, semantics and inference in prepositional logic as well as reasoning patterns. First Order Logic: syntax and semantics.

Resolution in FOL.

Probabilistic Reasoning


Bayesian Networks / Belief Networks

Representation of knowledge in uncertain domains. Semantics of Bayesian Networks. Exact inference in Bayesian Networks: inference by enumeration; PolyTree Bayesian networks.
CS 6367 Software Testing, Validation, and Verification

Textbook:

- Software Testing by Paul Jorgensen, 2nd Edition, CRC.

**Part 1: Requirements-Based Testing, Inspections**

Introduction, Approaches to Reliability, Requirements-based
Testing strategies (Equivalence Partitioning, Boundary value Analysis, Cause-Effect graphing), Valid and Reliable testing strategies and the Fundamental Theorem of testing, the Partition Testing Model, Random/Statistical testing.
Software Inspections and related approaches.
Textbook: Ch 1, 3, 5-8

**Part 2: Program Proofs**

Predicate calculus, validity, theoretical limitations, deduction systems, the Resolution method.
Verification of Programs (Flowchart Programs, Inductive Assertions, Termination, Programs with Arrays, extensions).
Chapter 2: Predicate Calculus
Chapter 3: Verification of Programs

**Part 3: Structural, Fault-Based Testing Strategies**

Structural Testing, Statement, Branch, Predicate, Base-Path,
Textbook: Ch. 9-11, 13, 16-20
Part 4: Reliability Estimation
Failure rate estimation from test outcomes, error-seeding, reliability growth models.

Notes on Reserve in Library

References:
Musa: Software Reliability Engineering, McGraw-Hill.
CS 6371 Advanced Programming Languages

Topics:

- Programming with Functions; Lambda Calculus and ML programming;
- Logic programming; Unification and backtracking; Search tree; Programming in Prolog;
- Abstract Syntax; Definite Clause Grammars; Grammar Classifications;
- Sets, functions, domains; Domain Theory: Primitive and Compound Domains;
- Denotational Definition of Programming Languages; Semantics of Imperative Languages; Recursive Functions; Monotonicity, Continuity, and Fix-points;
- Introduction to semantics of Logic Programming Languages, Verification of Programs, Partial Evaluation; Interpretation and Automatic Compilation;
- Axiomatic Semantics: Hoare's Axiomatization of partial correctness

Reading List:

1. "The Formal Semantics of Programming Languages: An Introduction" by Glynn Winskel
2. "Types and Programming Languages" by Benjamin C. Pierce

References:

- Denotational Semantics by D.A. Schmidt.

Also see the following web page for more details:

http://www.utdallas.edu/~gupta/courses/apl/
CS 6375 Machine Learning (Syllabus updated Oct 2006)

Topics:
- Decision Tree Learning, Artificial Neural Networks, Evaluating Hypotheses, Bayesian Learning, Computational Learning Theory,
- Instance-Based Learning, Markov Decision Processes, Reinforcement Learning, Support Vector Machines, Bagging, Boosting, Hidden Markov Models, and Clustering.

References:
CS 6378 Advanced Operating Systems

(Material in red with strikethrough is no longer in the syllabus)

Clocks and Event Ordering

• Lamport - Time, Clocks and the Ordering of Events in a Distributed System (1978)

Causal Message Ordering

• Raynal, Schiper & Toueg - The causal ordering abstraction and a simple way to implement it (1991)

Consistent Global Snapshots

• Chandy & Lamport - Distributed Snapshots: Determining Global States of Distributed Systems (1985)

Termination Detection

• Huang - Detecting Termination of Distributed Computations by External Agents (1989)

Distributed Mutual Exclusion

• Raymond - A Tree-Based Algorithm for Distributed Mutual Exclusion (1989)

Clock Synchronization

• Cristian - Probabilistic Clock Synchronization (1989)
• Gusella & Zatti - The Accuracy of the Clock Synchronization Achieved by TEMPO in Berkeley UNIX 4.3BSD (1989)

Agreement Protocols

• Fischer - The Consensus Problem in Unreliable Distributed Systems (1983)
Fault Tolerance and Data Consistency

- Bernstein, Hadzilacos & Goodman - Distributed Recovery (1987)
- Jajodia & Mutchler - A Hybrid Replica Control Algorithm Combining Static and Dynamic Voting (1989)

File System

- DeCandia et al. - Dynamo: Amazon's highly available key-value store
CS 6385 Algorithmic Aspects of Telecommunication Networks

CS 6390 Advanced Computer Networks

General topics:
1. Philosophy of the Internet
2. Internet Protocols basics (IPv4/IPv6, ICMP, etc)
3. Routing protocols (including multicast routing)
4. Transport layer protocols
5. Congestion Control Schemes
6. Quality of service and Weighted Fair Queuing
7. Mobile IP/Wireless Data
8. MPLS
9. Peer-to-peer applications
10. Voice over IP
11. Basics of Network Security

Reading List:
   a. Chapter 6 could be referenced for Congestion Control Schemes topic
   b. Chapter 8 could be referenced for Basics of Network Security topic
4. The Internet is for Everyone, V. Cerf, RFC 3271, April 2002.

NOTE: The above papers may be found in the IEEE/IEE Xplore database and in the ACM Digital library. These resources are freely accessible from within UTD campus network.