SE/CS 6329 Object-Oriented Software Engineering

The exam will test knowledge of:

- Software system models from the *use-case*, *logical*, *process*, *implementation* and *deployment* views;
- Iterative, evolutionary and agile processes;
- OO methodology for software analysis, design and implementation;
- Design patterns;
- UML notation including *structure diagrams* (class diagrams, composite structure diagrams, component and subsystem diagrams, package diagrams, deployment diagrams) and *dynamic diagrams* (use case diagrams, activity diagrams, statechart diagrams, interaction diagrams).

Textbook:

(L) *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Craig Larman, ISBN: 013 148 9062, Prentice-Hall, 2005.

(F) *UML Distilled: A brief Guide to the Standard Object Modeling Language*, Martin Fowler, Addison Wesley, 2004.

NOTE:

Students need to be familiar with basic software engineering concepts (lifecycles, requirements, high-level architecture, detailed design, coding) to provide solutions to the questions. Students are encouraged to review the case studies discussed in textbook (L).

List of topics:

- Iterative, evolutionary and agile process (*(L)* Chapters 2, 4, 5, 8,12, 14)
- Use cases (*(L)* Chapters 4-7);
- Domain Model (*(L)* Chapter 9)
- System Sequence Diagrams (*(L)* Chapter 10)
- Operation Contracts (*(L)* Chapter 11)
- Architecture (*(L)* Chapters *13, 38, (F)* Chapters 13-14, 8*),*
- Documenting Architecture: UML & the N+1 View Model (*(L)* Chap 39)
- Design Patterns (*(L)* Chapters 17, 25)
- Use-Case Realizations (*(L)* Chapters 18-19)
- Design Class Diagrams (*(L)* Chapters 18-19),
- Implementation (*(L)* Chapter 20)
- Statechart Diagrams (*(F)* Chapter 10, *(L)* Chapter 29)

UML Notation (*(F)* Chapters 3-15)

CS 6352 Performance of Computer Systems and Networks

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 queuing 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:

- T. G. Robertazzi, Computer Networks and Systems: Queueing Theory and Performance Evaluation. Springer, 2000.

Other References:

- D. Gross and C. M. Harris, Fundamentals of Queueing Theory. Wiley, 1997
- L. Kleinrock, Queueing Systems, Volume 1, Theory. Wiley, 1975
- J. J. Higgins and S. Keller-McNulty, Concepts in Probability and Stochastic Modeling. Duxbury Press, 1995
- K.S. Trivedi, Probability and Statistics with reliability, Queueing, and Computer Science Applications. First or Second Edition (2001, Wiley)
- C. M. Grinstead and J. L. Snell, Introduction to Probability. American Mathematical Society, 1997

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:

- Lecture Notes at http://www.utdallas.edu/~chung/SA/contents.html

Articles:

- Mary Shaw, Paul Clements, "The Golden Age of Software Architecture," IEEE Software, Vol.23, no. 2, pp. 31-39, Mar./Apr.2006
- Paul Clements, Mary Shaw, ""The Golden Age of Software Architecture Revisited, " IEEE Software, vol. 26, no. 4, pp. 70-72, July/August, 2009.
- Hofmeister, C., Kruchten, P., Nord, R., Obbink, H., Ran, A., & America, P. (2207). A General Model of Software Architecture Design derived from Five Industrial Approaches, Journal of Systems & Software, 80(1), 106-126.
- Hassan Gomma, "Advances in Software Design Methods for Concurrent, Real-Time and Distributed Application," in proceedings The Third International Conference on Software Engineering Advances, 2008, pp. 451-456.
- Kendall Scott, The Unified Process Explained. ISBN 0-201-74204-7, 2002, best practices in Architecture and Design.
- Advanced Design Patterns. Re-use
- M-A. Laverdiere; A. Mourad; A. Hanna; M. Debbabi; "Security Design Patterns: Survey and Evaluation", in Canadian Conference on Electrical and Computer Engineering, May 2006, pp. 1605 – 1608
- David Kalinsky, "Design Patterns for High Availability", March 13, 2003 URL: http://www.eetimes.com/story/OEG20020729S0030

- Ashraf Armoush, Falk Salewski, Stefan Kowaleski, "Design Pattern Representation for SafteyCritical Embedded Systems", JSEA
- Glen B. Alleman, "Exception Handling in CORBA Environments, The Late Introduction of Distributed Exception Handling in JAVA TM, CORBA-Based COTS Application Domains", 2000.

Representation:

- Nenad Medvidovic. Modeling software architectures in unified modeling language. ACM Transactions on Software Engineering and Methodology, 11(1):2-57, January 2002.
- Nenad Medvidovic and Richard N. Taylor. A classification and comparison framework for software architecture description language. IEEE Transactions on Software Engineering, 26(1):70-93, January 2000.
- P. Clements, Comparing the SEI's Views and Beyond Approach for Documenting Software Architectures' with ANSI-IEEE 1471-2000
- Grady Booch, James Rumbaugh, and Ivar Jacobson, Unified Modeling Language User Guide, (2nd Edition), Advanced UML topics, Sections 3, 5, and 6.
- Mugurel T. Ionita, Henk Obbink and Dieter Hammer, Scenario-Based Architecture Evaluation Methods: An overview, International Conference on Software Engineering 2002 (ICSE'02), Orlando, Florida.
- Hans-Peter Hoffmann, UML 2.0-Based Systems Engineering Using a Model-Driven Development Approach, 2005, available at http://www.stsc.hill.af.mil/crosstalk/2005/11/0511Hoffman.pdf
- Michel, M.M. Gala-Edeen, G.H., "Detecting inconsistencies between software architecture views", in proceedings International Conference on Computer Engineering & Systems, 2009. ICCES 2009. pp. 429-434
- Pengcheng Zhang, Henry Muccini and Bixin Li, "A classification and comparison of model checking software architecture techniques", 2009.
- Jeannette M. Wing and Mandanna Vaziri-Farahani, "A case study in model checking software systems", Science of Computer Programming 28 (1997) pp. 273-299.

Books:

1. Mary Shaw and David Garlan, Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, 1996.
2. L. Bass, P. Clements & R. Kazman, Software Architecture in Practice, Addison Wesley, 1998.
3. A. W. Brown (Editor), Component-Based Software Engineering, IEEE Computer Society, 1996.
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.
5. Wolfgang Pree, Design Patterns for Object-Oriented Software Development, Addison-Wesley Longman, 1995.
6. Singh, B. Stearns, M. Johnson, The Enterprise Team, Designing Enterprise Applications with the J2EE Platform, 2/E, Addison Wesley & Benjamin Cummings, 2002.

7. Randy Otte, Paul Patrick and Mark Roy, Understanding CORBA: The Common Object Request Broker Architecture, Prentice Hall, 1996.
8. Robert Orfali, Dan Harkey and Jeri Edwards, The Essential Client/Server Architecture: Survivor's Guide, John Wiley & Sons, 1995.

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)
- Multiplication of large integers (Section 7.1, page 219-223, and
- Problems 7.2, 7.3, page 250, of "Fundamentals of Algorithms, by
- Brassard and Bratley, Prentice Hall Publ.)
- 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:

- S. Russell & P. Norvig. Artificial Intelligence, A Modern Approach, Second Edition 2002.

Problem solving by search:

Uninformed (Blind) Search and Heuristic (Informed) Search

Problem formulation; Uninformed search strategies: Depth-First Search,

Breadth-First Search, Ununiform-Cost Search, Iterative-Deepening.

Informed Search strategies: Greeedy 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

Modeling uncertainties with probabilities. Inference using Full Joint Distributions. Bayes' Rule. Naïve-Bayesian 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

Myers: The Art of Software Testing, Wiley.

Goodenough + Gerhart, "Toward a Theory of Test Data Selection", IEEE Trans. on Software Engineering, June 1975.

Hamlet+Taylor, "Partition Testing Does Not Inspire Confidence", IEEE TSE, Dec. 1990.

Wheeler, Brykczynski, Meeson, "Software Inspection: An Industry Best Practice", IEEE Computer Society Press.

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).

Manna: Mathematical Theory of Computation, McGraw-Hill.
Chapter 2: Predicate Calculus

Chapter 3: Verification of Programs

Part 3: Structural, Fault-Based Testing Strategies

Structural Testing, Statement, Branch, Predicate, Base-Path,

Path Testing, Variations of Path Testing, Data-Flow Testing, Domain Testing, Mutation Analysis, other methods. Evaluations of testing strategies, inclusion, test set size. Integration testing; Object-oriented Testing

Textbook: Ch. 9-11, 13, 16-20

DeMillo, Lipton, Sayward, "Hints on Test Data Selection: Help for the Practicing Programmer", IEEE Computer, April 1978.

Musa, "Operational Profiles in Software Reliability Engineering", IEEE Software, March 1993.
Ntafos, "A Comparison of Some Structural Testing Strategies",

IEEE TSE, June 1988.

White, Cohen, “A domain strategy for Computer Program Testing”, IEEE-TSE, May 1980.

Part 4: Reliability Estimation

Failure rate estimation from test outcomes, error-seeding, reliability growth models.
Notes on Reserve in Library

References:

Lyu: Handbook of Software Reliability Engineering, IEEE Computer Society Press, Mc Graw Hill.
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.
- Elements of ML Programming, Jeffrey D. Ullman, ML97 Edition The Art of Prolog, L. Sterling and E. Shapiro. MIT Press, 1997.

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:

- Artificial Intelligence (second edition) by Stuart Russell and Peter
- Norvig, Prentice Hall, 2003.
- Machine Learning by Tom Mitchell, McGraw Hill, 1997.

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)
- Fidge - Logical Time in Distributed Computing Systems (1991)

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

- Ricart & Agrawala - An Optimal Algorithm for Mutual Exclusion in Computer Networks (1981)
- Maekawa - A sqrtN Algorithm for Mutual Exclusion in Decentralized Systems (1985)
- 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)
- Mills - Improved Algorithms for Synchronizing Computer Network Clocks (1995)

Agreement Protocols

- Fischer - The Consensus Problem in Unreliable Distributed Systems (1983)

Fault Tolerance and Data Consistency

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

File System

- Ghemawat, Gobioff & Leung - The Google File System (2003)
- DeCandia et al. - Dynamo: Amazon's highly available key-value store

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:

1. Reference book (Computer Networks by Peterson and Davie – 5th Edition)
    a. Chapter 6 could be referenced for Congestion Control Schemes topic
    b. Chapter 8 could be referenced for Basics of Network Security topic
2. Design Philosophy of the DARPA Internet Protocols, D. Clark, Proc. of ACM SIGCOMM '88.
3. Creating a Mathematical Theory of Computer Networks, L. Kleinrock, Operations Research, 2002.
4. The Internet is for Everyone, V. Cerf, RFC 3271, April 2002.
5. An Architecture for Wide-Area Multicast Routing, S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, Proc. of ACM SIGCOMM'94.
6. Multicast Routing in Datagram Internetworks and Extended LANS, S. Deering and D. Cheriton, ACM Transactions on Computer Systems, Vol 8 No 2, May 1990, pp. 85-110.
7. IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications, H. Holbrook and D. Cheriton, SIGCOM 1999.
8. The Stable Paths Problem and Interdomain Routing, T. Griffin, B. Shepherd, and G. Wilfong, IEEE/ACM Transactions on Networking, Vol 10 No 2, April 2002.
9. Mobile IP, C. Perkins, IEEE Communications Magazine, Vol 35, No. 5, May 1997.
10. Mobility Support in IPV6, C. Perkins, D. Johnson, ACM Mobicom 1996.
11. Congestion Avoidance and Control, V. Jacobson and M. Karels, Proc. ACM SIGCOMM '88.
12. Random Early Detection Gateways for Congestion Avoidance, S. Floyd and V. Jacobson, IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp. 397- 413, August 1993.
13. Equation-Based Congestion Control for Unicast Applications, S. Floyd, M. Handley, J. Padhye, and J. Widmer, Proc. of ACM SIGCOMM '00, Aug. 2000.
14. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, I Stoica, R Morris, D Liben-Nowell, D R. Karger, M. F Kaashoek, F Dabek, H Balakrishnan, ACM SIGCOMM 2001.

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.