5/20/2009

THE WPE MEMO 2009-2010
(FALL 2009)
Department of Computer Science and Engineering
University of Minnesota

1. Introduction

This memo describes the Fall 2009 written preliminary examination (WPE) for the PhD program in the Department of Computer Science and Engineering at the University of Minnesota.

Who Can Take This Examination

Only students admitted to the graduate program in Computer Science (MS or PhD) or Computer Engineering MS students may take this examination. MS students who pass this examination may be allowed to apply to the PhD program.

All graduate students who intend to take the WPE must complete the application form attached to this memo. Applications must be submitted in 4-192 EE/CSci by the deadline specified on the form. Students will be permitted to take the WPE only if the WPE Committee accepts their applications.

2. WPE Format

To be accepted into Ph.D. candidacy, students must complete the Breadth requirement and pass the Major exam, which consists of two parts:

1. In-Class Exam
2. Take Home Exam

2.1 PhD Breadth Requirements

The Breadth requirement can be satisfied through coursework as detailed below and need not be completed before taking the written part of the WPE:

1. The required number of Breadth courses is 6. Two courses are required in each of the 3 breadth areas:

   - Theory
   - Systems
   - Applications

   The courses in these areas are listed below.

2. The student should get a B (3.0) or better grade in each of the courses. Students must maintain an overall GPA of 3.25 (for Master's candidates) and 3.45 (for Ph.D. candidates) for all courses on their degree program, including those used to satisfy the breadth requirement. The breadth courses must be listed on the student's Degree Program.

3. A student with an MS degree from another University can petition to transfer up to 3 breadth courses, one from each area. Courses used to obtain the MS degree in our CS or CE programs can be reused for the PhD.

4. For students entering the Ph.D. program with a substantial number of requisite Breadth courses from their previous programs, there would be a provision to substitute advanced courses (including 8xxx level) in place of the Breadth courses listed below. This would be decided on a case-by-case basis by the DGS in consultation with the appropriate faculty. The student will be required to file a petition to make such substitutions.

5. List of courses:

Note: OR means exclusive-or.

Theory:

5302 Analysis of Numerical Algorithms
5304 Computational Aspects of Matrix Theory
5403 Computational Complexity
5421 Advanced Algorithms and Data Structures
5451 Intro to Parallel Computing: Architecture, Algorithms & Programming
5471 Modern Cryptography
5525 Machine Learning

No more than one of the following Mathematics courses (PhD students only):
MATH 5165 Mathematical Logic
MATH 5707 Graph Theory and Non-enumerative Combinatorics
MATH 5711 Linear Programming and Combinatorial Optimization
EE 5531 Probability and Stochastic Processes

Systems:

5103 Operating Systems
5104 System Modeling and Performance Evaluation
5105 Foundations of Modern Operating Systems
5106 Programming Languages
5131 *Advanced* Internet Programming (Beginning Fall 2002)
5143 Real-Time and Embedded Systems
5161 Introduction to Compilers
5204 Advanced Computer Architecture
5211 Data Communications and Computer Networks
5271 Introduction to Computer Security
5708 Architecture and Implementation of Database Management Systems

No more than one of the following EE courses (PhD students only):
EE 5323 VLSI Design I
EE 5371 Computer Systems Performance Measurement and Evaluation
EE 5381 Telecommunications Networks (This cannot be counted with CSci 5211)

Applications:

5107/5108 Fundamentals of Computer Graphics I OR II
5109 Visualization
5115/5116 User Interface Design: Implementation and Evaluation OR GUI Tools
5283 Computer Aided Design I
5481 Computational Techniques for Genomics
5511/5512/5519 Artificial Intelligence I OR Artificial Intelligence II
5521 Pattern Recognition
5523 Introduction to Data Mining
5541 Natural Language Processing
5551 Intro to Intelligent Robotic Systems
5552 Sensing and Estimation in Robotics
5561 Computer Vision
5707 Principles of Database Systems
5801/5802 Software Engineering I OR Software Engineering II

No more than one of the following EE courses (PhD students only):
EE 5329 VLSI Digital Signal Processing Systems
EE 5301 VLSI Design Automation I (This cannot be counted with CSci 5283)

These "OR" operators are exclusive-OR: only one of the two classes may be taken to satisfy the breadth requirement in the area. The breadth requirement is specified in this way to ensure that students gain broad exposure within computer science, rather than focusing on a narrow sub-discipline.

2.2 Major In-Class and Take-Home Examinations

Students will be required to discuss with their research advisor or the DGS their research plans and objectives to determine the subjects in which to take the in-class and take-home exams.

The Major exam will have a uniform structure for all students, irrespective of their areas of research. The Major exam will have two parts:

In-Class

- The student will be taking an in-class exam consisting of **two** parts on **two** different subjects from the list in row 1 of Table 1. A student will choose **any two** subjects in which to answer questions. The subjects selected should be most appropriate for the student's thesis work. For example, a student can select HCI and Learning, or OS and Networks. The duration of the in-class exam will be 4 hours, with about 2 hours needed for each subject exam, which will be comprehensive.

- Table 1 consists of a row of in-class questions and a column down the side of take home questions. If there is an X where the two topics intersect, a student can only take either the in-class **or** the take-home exam. The student should choose, in consultation with his/her advisor, the subject areas that are best related to the student's research plans.

- Each subject area exam will be either open-book or closed book. If an in-class exam subject will be offered as an open book exam, only reference books listed in the WPE memo will be permitted. No typed or handwritten notes or any other printed materials will be permitted for any exam and only basic calculators will be allowed. There will be no laptops. PDAs or cell-phones allowed.

Take-Home

- The student will be required to answer a take-home exam in **one** of the subjects listed in column 1 of Table 1. A student can take the take-home exam in any one of these subjects, independent of the choice of major area. For instance, a theory student interested in computer vision algorithms may choose to take the Computer Vision take-home exam.

- The take-home exam will be distributed on a Friday by noon, or immediately after the in-class exam if the in-class exam is held on a Friday. The students will be required to submit their paper by noon on the following Tuesday.

- The purpose of the take-home exam is to assess the ability of the student to analyze research literature in the subject area or answer questions based on a set of papers.

- The students will be required to declare the subjects for the in-class and the take-home at the time they register for the WPE. Take-home exams will only be prepared for subjects students have chosen in advance.

If there is an X in a box, you cannot take both the in-class and take-home questions.

5/20/2009

In-Class Exam Subjects / **Take-Home**	Architecture (5204)	CAD (5283)	Compilers (5161)	Computational Complexity (5403)	Data Structures and Algorithms (5421)	Databases (5707)	Graphics (5107)	HCI (5115)	Learning (5512)	Logic, Knowledge Rep, Plan, Search (5511)	Networks (5211)	Numerical Analysis (5302 and 5304)	Operating Systems (5103)	Programming Languages (5106)	Robotics and Sensing (5551)	Software Engineering (5801)
Architecture	**X**															
CAD		**X**														
Compilers			**X**													
Computational Geometry																
Computer Vision																
Data Mining																
Databases						**X**										
Distributed Systems													**X**			
Graphics							**X**									
HCI/GUI								**X**								
Intelligent Agents																
Math Models for Data Mining																
Networks											**X**					
Parallel Algorithms																
Programming Languages														**X**		
Security/ Cryptography																
Software Engineering																**X**
Sparse matrix computations																
Special matrix problems and Advanced matrix computations																
Visualization																

Table 1

2.3 Administration of the Examination

The student is allowed to take this examination, at most, two times. This means that the student can attend only two examination cycles. At the student's first attempt, the student will take the major in-class and take-home examinations. A student who fails any of the exams can retake the failed exam(s) at the next examination cycle.

The two in-class subject exams and the take home exam will be graded separately and can be passed or failed independently.

The student will be required to pass the In-Class exams and the Take-Home in at most 2 attempts. In case of failing any of the exams, the student would get only one additional chance to retake the failed exams on the next offering of the WPE. In order to pass the In-class exam, the student must pass the two subject exams. These can be passed independently and if one is passed and one is failed at the first attempt, only one subject exam needs to be re-taken at the second attempt. The student does not have to re-take the same subject exam. A student retaking only one in-class subject exam in the second attempt will be given only 2 hours. If a student does not pass the Take-Home exam the first time, it must be re-taken the next time it is offered. The student may change the exam subject with approval of the research advisor or DGS.

2.4 Grading

The grading and decisions will be anonymous, only at the final general WPE meeting will the names of candidates be known. Conditional passes will be given only in the most extraordinary exceptional cases and only if the student has passed at least one of the exams. The examination will be administered centrally. Graduate faculty members within each area will be in charge of preparing and grading questions in their area.

Although the format of the examinations may vary across areas, the grading procedures and standards applied will be uniform. Each question will be graded on a scale from one to ten with 6 considered minimal passing. Two graduate faculty members will grade the answers to each question. In case of unresolved substantial disagreement in grades, an arbitrator will be used to resolve the discrepancy.

2.5 Appeals Procedure

All appeals of the results of the WPE must be initiated within one month of the announcement of the results. An examination will be reread only if an obvious error was made. No rereading to improve a marginal result will be performed. The chairperson of the appropriate area will be contacted in this regard, as well as the WPE chair and the DGS.

2.6 Time and Place of the Examination

The Fall 2009 in–class exam will take place on Thursday, September 3, 2009 and the take home exam will be distributed the next day, Friday, September 4, 2009 with the paper due on Tuesday, September 8, 2009. The place for the exam will be announced later.

5/20/2009

Fall 2009 WPE Reading Lists

3.0 In-Class Exam Reading Lists

Architecture (5204) (Open Book)

1) J. L. Hennessy & D. A. Patterson, *Computer Organization & Design: The Hardware & Software Interface*, Morgan & Kaufman, Second Edition, 1997.

2) J.L. Hennessy & D.A. Patterson, *Computer Architecture: A Quantitative Approach,* Morgan & Kaufman, Fourth Edition, 2007.

CAD (5283) (Open Book)

1) M. Abramovichi et. al. *Digital Systems Testing and Testable Design*. Ch. 1-6. IEEE Press.

2) Sudhakar Yalamanchilli, *Introductory VHDL From Simulation to Synthesis,* Prentice Hall 2001.

Compilers: (5161) (Open Book)

1) Alfred V Aho, Monica S. Lam, Ravi Sethi, and Jeffery D Ullman, "Compilers: Principles, Techniques and Tools (2nd Edition)", Addison Wesley, 2006.

2) Steven S. Muchnick, "Advanced Compiler Design Implementation" Morgan Kaufmann, 1997.

Computational Complexity (5403) (Open Book)

1) Michael Sipser, "Introduction to the Theory of Computation." (2nd Edition), Thomson, 2006.

2) Sanjeev Arora and Boaz Barak, "Complexity Theory: a Modern Approach", 2007; chapters 1-10. available from http://www.cs.princeton.edu/theory/complexity/

Data Structures and Algorithms (5421) (Closed Book)

The reading list will consist of the following material, most of which is generally covered in CSci 5421. This includes:

(i) Chapters 1-4, 6-10, 12-18, 20-25, 28.1-2, and 33 from the 2nd edition (year 2001) of the Cormen text [CLRS] or the equivalent chapters from the 1st edition (year 1990).

(ii) additional material covered in class, e.g., splay trees, d-ary heaps, skip-lists, persistent data structures, and any geometric algorithms beyond those covered from [CLRS].

[CLRS] Introduction to Algorithms, by T. Cormen, C. Leiserson, R. Rivest and C. Stein, MIT Press and McGraw Hill, 2nd edition, 2001.

Databases (5707) (Closed Book)

R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Benjamin Cummings. 3rd Edition Chapters 1-8, 13, 14 and 23.

Graphics (5107) (Closed Book)

FvDFH is Foley J. D., vanDam A., Feiner S. K., and Hughes J. F., (1996). Computer Graphics: Principles and Practice. Addison-Wesley.

Hill is Hill F. S. (2001). Computer Graphics: Using OpenGL. Second Edition Prentice Hall.

Shirl is Shirley P. (2002). Computer Graphics. A. K. Peters.

Watt is Watt A., 3D Computer Graphics, Addison Wesley.

Scan Conversion and Anti-Aliasing
FvDFH: Sections 3.2-3.6, 3.11-3.14, 3.17-3.18, 14.10, 19.2, 19.3, 19.5,
Hill: Sections 10.4-10.8
Shirl: Chapter 3, 14, 15
Watt: Chapter 6.4, 14.1-14.7

Geometric Transformations
FvDFH: Sections 5.1-5.7
Hill: Chapter 5
Shirl: Chapter 4, 5
Watt: Chapter 17.2.3-17.2.5

Viewing and Projection
FvDFH: Chapter 6
Hill: Chapter 7
Shirl: Chapter 6
Watt: Chapter 5.1-5.2

Hidden Surface
FvDFH: Chapter 15
Hill: Chapter 13
Shirl: Chapter 7
Watt: Chapter 6.6

Parametric Curves and Surfaces
FvDFH: Chapter 11
Hill: Chapter 11
Shirl: Chapter 13
Watt: Chapter 3.1-3.2

Color Models, Basic Illumination and Shading
FvDFH: Sections 13.2-13.5; 16.1-16.2
Hill: Sections 8.1-8.4
Shirl: Chapter 8, 18
Watt: Chapter 6.2-6.3, 15.1-15.3, 15.5.4

Texture Mapping
FvDFH: Sections 16.3, 16.10
Hill: Section 8.5
Shirl: Chapter 10
Watt: Chapter 8.1-8.6, 8.8

Ray Tracing
FvDFH: Section 15.10, 16.5, 16.12
Hill: Chapter 14
Shirl: Chapter 9
Watt: Chapter 1.4, 10.6 - 10.7, 12

Advanced Illumination and Shading
FvDFH: Sections 16.7, 16.13
Hill: Section 14.7

Shirl: Chapter 19, 21
Watt: Chapter 7, 11

HCI (5115) (Closed Book)

1) Design of Everyday Things (by Donald Norman), Reissue edition (March 1990) Currency/Doubleday

2) Task-Centered User Interface Design (by Clayton Lewis and John Rieman), 1993, 1994.
http://home.att.net/~jrieman/jrtcdbk.html

Learning (5512) (Open Book)

This question will be from the topics of decision trees, version space, neural networks, and reinforcement learning.

The material is covered in the AI textbook:

Stuart Russell and Peter Norvig
"Artificial Intelligence. A modern approach. Second Edition", Prentice-Hall, 2003.

in Chapters 18 (decision trees), 19.1 (version space)
20.5 (neural networks), 17.1-3 and 21 (MDPs and reinforcement learning).

The treatment in Russell and Norvig is adequate, but we strongly suggest to supplement it with:

Tom Mitchell, Machine Learning, McGraw Hill, 1997 Chapters 1 (introduction), 2 (version space), 3 (decision trees),
4 (neural nets), and 13 (reinforcement learning)

Additional supplemental readings are in the textbooks for CSci 5521:

R. Duda, P. Hart, D. Stork, "Pattern Classification," Wiley, 2000.
Chapter 1 (introduction), Chapter 5 (perceptron), Chapter 6 (multilayer neural networks), Chapter 8 (decision trees).

C. M. Bishop, "Neural Networks for Pattern Recognition", Oxford University Press, 1995. Chapters 3 and 4 on neural networks.

Logic, Knowledge Representation, Planning and Search (5511) (Open Book)

This question will be from the topics of search, knowledge representation, predicate calculus and resolution, and planning.

The material is covered in the AI textbook:

Stuart Russell and Peter Norvig
"Artificial Intelligence. A modern approach. Second Edition", Prentice-Hall, 2003.

in Chapters 3, 4, 5 and 6 (search), 7, 8, and 9 (logic and resolution), 10 (knowledge representation), 11 and 12 (planning).

An excellent supplemental book for the material on search is:

J. Pearl, Heuristics: Intelligent search strategies for computer problem solving, Addison Wesley, 1984, Chapters 2-3.

An additional useful reference for the material on planning is:

D. Weld, "An introduction to least commitment planning", AI Magazine, Winter 1994, pp 27-61.

Networks (5211) (Closed Book)

1) Andrew S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, 1996, Chapters 1-6.

2) or Jim F. Kurose and Keith W. Ross, *Computer Networks: A top-Down Approach Featuring the Internet,* Addison Wesley, 2001.

3) Douglas E. Comer, *Computer Networks and Internets,* 3rd Edition, Prentice Hall, Chapters 1-25, in particular chapters 17-25.

Numerical Analysis (5302 and 5304) (Open Book)

1) *Matrix Computations* 3rd edition. G. Golub and C. Van Loan. John Hopkins, Chapters 1, 2, 3, Sec 4.1 -- 4.4, Sec 5.1 -- 5.5, Sec 7.1 – 7.5, Sec 8.1 – 8.5, Sec 10.1.

2) Demmel, J.W., *Applied Numerical Linear Algebra,* SIAM, 1997, Chapters 1 to 4, Sec. 5.1 – 5.3, Sec 6.5.

3) Scientific Computing, An Introductory Survey, 2nd edition, by Michael T. Heath, McGraw Hill, 2002.

4) Kincaid, D. R. and Cheney, W. W. , *Numerical Analysis: the Mathematics of Scientific Computing,* 3rd edition, Van Nostrand, 2002.

Operating Systems (5102) (Closed Book)

The in-class exam questions will be based on the following topics. The following two textbooks provide an elaborate exposition of these topics.

1. *Modern Operating Systems* (Second Edition), by Andrew Tanenbaum, Prentice-Hall, 2001.
2. *Operating System Concepts* (Fifth Edition), by Silberschatz and Galvin, Addison-Wesley, 1998.

Topics:

Basic concepts (Chapter 1 of book [1] and Chapters 1-3 of [2])
Important functions of an operating system: resource management, protection, process management, memory management, file systems.
History of development, multiprogramming, time-sharing systems, protection, real-time operating systems, system calls
Computer system structures: hardware organization, memory, I/O buses, storage hierarchy, DMA, I/O operations, interrupts and their handling.
Operating System Structures: system calls, system programs, command shell, virtual machines.

Process Management (Chapter 2 of book [1] and Chapters 4-5 of [2])
Concept of processes: logical view of a process as a virtual computer, process creation, execution context, termination, process hierarchies, state of a process, implementation level structures, process control block.
Concept of threads: thread model, user level vs. kernel level threads. Cooperating processes, interprocess communication: shared memory vs. messages and pipes.

CPU scheduling: basic concepts, context switching, preemptive and non-preemptive scheduling, FCFS, round-robin, multi-level feedback queues, scheduling in real-time systems, earliest-deadline-first and fair-share scheduling.

Process Synchronization (Chapter 2 of book [1] and Chapter 6 of [2])
Critical section problem, process coordination using shared data (flags), hardware based synchronization mechanisms (e.g. test-and-set), concept of semaphores and mutexes; classical problems: bounded buffer, dining philosopher, sleeping barber problem, reader/writer problem; monitors: Hoare monitor, Java monitor model; atomic transactions.

System Deadlock (Chapter 3 of book [1] and Chapter 7 of book [2])
Necessary and sufficient conditions of system deadlocks; techniques for deadlock prevention, avoidance, and detection; use of resource allocation graphs to characterize deadlock conditions; Banker's algorithm for deadlock prevention; recovery from deadlocks.

Memory Management (Chapter 4 of book [1] and Chapters 8-9 of [2])
Address space of a process; address binding; logical vs. physical address space.
Contiguous allocation: multiprogramming with fixed partitions; multiprogramming with variable partitions.
Non-contiguous allocation: page based systems, page tables, segmented systems.
Memory free-space management: bit maps, free-lists.
Virtual memory management: demand paging, page tables, page-faults, address translation and TLBs, page replacement algorithms: LRU, FIFO, Clock, stack algorithms; Belady's anomaly; concept of distance strings, predicting page faults;
Working set model for memory allocation: techniques for working set estimation, reference bit/register based techniques, WSClock algorithm.
System level issues: effect of page sizes, support for variable page sizes; page locking in memory' system level data structures and mechanisms for virtual memory support.
Multilevel page tables, inverted page tables.

Input/Output (Chapter 5 of book [1] and Chapter 12-13 of [2]
I/O devices and controllers, interrupt handling, memory mapped I/O, interrupt driven I/O, DMA based I/O, I/O software organization, device drivers, user-space I/O software.
Disk hardware, disk formatting, I/O scheduling algorithms.
Clock management.
Character-oriented I/O: RS-232 terminal.
Graphical user interfaces.

File Systems (Chapter 6 of book [1] and Chapter 11 of [2])
File system functions and APIs, file system structures, file naming, directories,
File system implementation, disk space allocation, directory implementation.
File system performance and reliability. Case studies: UNIX, MS DOS, Windows 98 file systems.

Protection and Security (Chapter 9 of book [1] and Chapter 19-20 of [2])
Goals of protection, concept of protection domains, access control matrix model, access control lists and capability based protection,.
User authentication, using passwords, one-time passwords, authentication in distributed systems. Digital signatures. RSA algorithm.
Security threats, Trojan horses, login spoofing, trap doors, buffer overflow problem, design principles for secure systems.
Viruses and worms, structure of viruses.
Multilevel security models.
Classification of secure computer systems: Orange Book classification.

Programming Languages (5106) (Open Book)

5/20/2009

This reading list is broken up into two parts, the first of which is common to the in-class and take-home major exams and some further readings for the take home part only. The material in the relevant list of papers and books should be understood as *basic* for each kind of major exam in the Programming Languages area. Questions in the in-class exam will probe understanding of the area obtained from familiarity with this material. The take-home exam may involve specific additional reading to be done during the exam period and will also call for critical thinking in a more relaxed timeframe than afforded by an in-class exam.

No single course covers all the topics in the listed papers. The way to interpret this reading list and, hence, the depth exam in Programming Languages is that you are expected through independent study to familiarize yourself with material beyond what you are exposed to in CSCI 5106. The textbook by Sethi thus defines the point of departure. The additional papers indicate some of the important themes that you should have a grasp of beyond what is in that text.

Common Readings for the In-Class and Take-Home Exams

1) A.W. Appel, Garbage collection, in *Topics in Advanced Language Implementation*, P. Lee (ed.), pages 89-100, MIT Press, 1991.

2) J. Bentley, Programming pearls, *CACM* 26(12): 1040-1045, December 1983.

3) L. Damas and R. Milner, Principal type-schemes for functional programs, *Ninth ACM Symposium on Principles of Programming Languages*, 1982, pages 207-212. (Make sure to know how to complete the proofs in this paper and also to understand how all this relates to type inference in ML. A slower paced reference can also be provided if necessary.)

4) C.A.R. Hoare, An axiomatic basis for computer programming, *CACM* 12(10): 576-580, October 1969.

5) C.A.R. Hoare, Communicating Sequential Processes, *CACM* 21(8): 666-777, August 1978.

6) C.A.R. Hoare, Hints on programming language design. In *Programming Languages: A Grand Tour*, Horowitz, E. (ed.), pp 31-40, Computer Science Press, Rockville, Maryland, 1987.

7) P.J. Landin, The mechanical evaluation of expressions, *Computer Journal* 6(4): 308-320, Jan 1964.

8) P.J. Landin, The next 700 programming languages, *CACM* 9(3): 157-166, March 1966.

9) J. McCarthy, Recursive functions of symbolic expressions and their computation by machine, Part I, *CACM* 3(4): 184-195, 1960.

10) R. Sethi, *Programming Languages: Concepts and Constructs (Second Edition),* Addison-Wesley, 1996.

11) R.D. Tennent, The denotational semantics of programming languages, *CACM* 19(8): 437-453, 1976.

12) D. Friedman and D. Wise, CONS should not evaluate its arguments, in *Automata, Languages and Programming*, Edinburgh University Press, 1976, pages 257-284.

13) N. Wirth, On the design of programming languages, IFIP Congress 74, pages 386-393, North Holland Publishing Company, Amsterdam. Also reprinted in *Programming Languages: A Grand Tour*, Horowitz, E. (ed.), pp 23-30, Computer Science Press, Rockville, Maryland, 1987.

Robotics and Sensing (5551) (Open Book)

1. J.J. Craig, "Introduction to Robotics, Mechanics and Control" 3rd Edition [Chapters: 1,2,3,4,5,7,9]

2. D.A. Forsyth and J. Ponce, "Computer Vision - A Modern Approach" [Chapters: 1 & 2]

3. R. Siegwart and I. Nurbakhsh, "Introduction to Autonomous Mobile Robots" [Chapter 4]

Software Engineering (5801) (Open Book)

The material in the papers and books listed below should be understood as *basic* for the major exam in the Software Engineering area. Questions in the in-class exam will probe the understanding of the area obtained from familiarity with this material. The take-home exam may involve specific additional reading and will also call for critical thinking in a more relaxed timeframe than afforded by an in-class exam.
Note: No single course covers all the topics in the listed papers. The way to interpret this reading list and, hence, the depth exam in Software Engineering is that you are expected through independent study to familiarize yourself with material beyond what you are exposed to in CSci 5801. The textbook by Hamlet and Maybee thus defines the point of departure. The additional papers indicate some of the important themes that you should have a grasp of beyond what is in that text.

Readings for the in class exam

Textbooks

1) The Engineering of Software. Dick Hamlet and Joe Maybee. Addison Wesley, 2001. ISBN 0-201-70103-0

Process

1) Frederick P. Brooks. No Silver Bullet: Essence and Accidents in Software Engineering. *IEEE Computer.* 20(4): 10-19, April 1987.

2) David L. Parnas and Paul C. Clements. A Rational Design Process: How and Why to Fake It. *IEEE Transaction on Software Engineering.* 12(2): p. 251-257, February 1986.

Testing

1) W. Richard Adrian, Martha A. Branstad, and John C. Cherniavsky. Validation, Verification, and Testing of Computer Software. *ACM Computing Surveys.* 14(2): p. 159-192, June 1982.

Architecture

1. David Garlan, Robert Allel, and John Ockerbloom. Architectural Mismatch: Why Reuse is so Hard. *IEEE Software* 12(6): 17-26, November 1995.
2. Davis Garlan. Research directions in software architecture. *ACM Computing Surveys*. 27(2): p. 257-261, 1995.

Requirements Reference Models

1. Carl A. Gunter, Elsa L. Gunter, Michael Jackson, and Pamela Zave. A Reference Model for Requirements and Specifications. *IEEE Software*. 17(3): p. 37-43, May/June, 2000
2. J.M. Thompson, M.P.E. Heimdahl, and S.P. Miller. Specification-Based Prototyping for Embedded Systems. *Proceedings of the Seventh ACM/SIGSOFT Symposium on the Foundations of Software Engineering*. P. 163-179. Toulouse, France, September 1999.
3. D.L. Parnas and Jan Madey. Functional documents for computer systems. *Science of Computer Programming*. 25: p. 41-61, 1995.

4.0 Take Home Exam Reading Lists

Architecture

From Readings in Computer Architecture, Edited by Mark D. Hill, Norman P. Jouppi, and Gurindar S. Sohi, September 1999, Morgan Kaufman Publishers

1. Cramming More Components into Integrated Circuits, G. E. Moore

2. Evaluating Associativity in CPU Caches, M. D. Hill and A. J. Smith

3. Compilers and Computer Architecture, W. A. Wulf

4. Instruction Sets and Beyond: Computers, Complexity, and Controversy, R. P. Colwell, C. Y. Hitchcock III, E. D. Jensen, H. M. Brinkley Sprunt, C. P. Kollar

5. A Comparison of Full and Partial Predicated Execution Support for ILP Processors, S. A. Mahlke, R. E. Hank, J. E. McCormick, D. I. August, W. W. Hwu

6. Two-Level Adaptive Training Branch Prediction, T.-Y. Yeh and Y. N. Patt

7. Executing a Program on the MIT Tagged-Token Dataflow Architecture, Arvind and R. S. Nikhil

8. Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers, N. P. Jouppi

9. A Case for Redundant Arrays of Inexpensive Disks (RAID), D. A. Patterson, G. Gibson, and R. H. Katz

Recent trends:

Reliability issue

1. Shubhendu S. Mukherjee, Christopher Weaver, Joel Emer, Steve Reinhardt, and Todd Austin. A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor, the 36th Annual International Symposium on Microarchitecture, December 2003, Annaheim, California.

2. Todd M. Austin. DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design, the 32nd International Symposium on Microarchitecture, November 1999, pp. 196-207.

Power issue

1. T. Mudge, "Power: A First-Class Architectural Design Constraint," IEEE Computer, 2001.

2. L. Shang and R. P. Dick, Thermal Crisis: Challenges and Potential Solutions, IEEE Potentials, pp. 31-35, vol. 25, no. 5, Sep. 2006.

Recent processors

1. Dean M. Tullsen, Susan J. Eggers, Joel S. Emer, Henry M. Levy, Jack L. Lo, and Rebecca L. Stamm. Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor, Proc. 23rd Annual International Symposium on Computer Architecture, May 1996, pp. 191-202.

2. Kongetira, P. Aingaran, K. Olukotun, K. Niagara: a 32-way multithreaded Sparc processor IEEE Micro March 2005 Volume: 25, Issue: 2 page: 21- 29

3. Harsh Sharangpani, Ken Arora Itanium Processor Microarchitecture, IEEE Micro September

4. Ron Kalla and Balaram Sinharoy and Joel M. Tendler, IBM Power5 Chip: A Dual-Core Multithreaded Processor, Microprocessor Forum, October 1999.

TM/TLS

1. James R. Larus and Ravi Rajwar. Transactional Memory. Synthesis Lectures on Computer Architecture, Morgan & Claypool Publishers, 2006.

2. J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. The STAMPede Approach to Thread-Level Speculation. ACM Transactions on Computer Systems (TOCS), Volume 23, Issue 3, August 2005.

CAD

1) Static Timing Analysis Including Power Supply Noise Effect on Propagation Delay in VLSI Circuits by G. Bai, S. Bobba, J. Hajj, in Proceedings of DAC 2001.

2) Reliable Verification Using Symbolic Simulation with Scalar Values by C. Wilson, D. Dill, in Proceedings of DAC 2000.

3) Symbolic RTL Simulation by A. Kobbe, J. Kukula, R. Damiano in Proceedings of DAC 2001.

4) SPIRIT" A Highly Robust Combinational Test Generation Algorithm, by E. Gizdarski, H, Fujiwara in Proceedings of 19th VLSI Test Symposium, VTS 2001.

5) Efficient Optimization by Modifying the Objective Function: Applications to Timing-Driven VLSI Layout, by R. Baldwick, A. Kang, A. Kennings and J. Markov. IEEE Transactions of Computer Aided Design of Integrated Circuits and Systems, 48(8), 2001, pp. 947-957.

6) Buffer Block Planning for Interconnect Planning and Prediction, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 9, No. 6, Dec. 2001, by J. Cong, T. Kong, Z. Pan

7) On Diagnosing Path Delay Faults in At-Speed Environments by R. Tekumalla and S. Venkataraman, J. Ghosh-Dastidar, in Proceedings of 19th VLSI Test Symposium, VTS 2001.

8) A Novel Framework for Logic Verification in a Synthesis Environment, in IEEE Transactions on CAD, Vol. 15, issue 1, Jan. 1996, by W. Kunz, D. K. Pradham, S. M. Reddy.

9) Fast and Accurate Timing Characterization Using Functional Information by H. Yalcin, M. Mortazavi, R. Palermo, C. Bamji, and K. Sakallah in IEEE Transactions on CAD, Vol. 20, issue 2, Feb. 2001.

10) Realization-Independent ATPG for Designs with Unimplemented Blocks, by H. Kim, J. P. Hayes, in IEEE Transactions on CAD, Vol. 20, issue 2, Feb. 2001.

Compilers

Static Flow and Pointer Analysis

1) On the importance of points-to analysis and other memory disambiguation methods for C programs, Rakesh Ghiya, et. al., Proceedings of the ACM SIGPLAN'01 conference on Programming language design and implementation 2001

2) Efficient context-sensitive pointer analysis for C programs Robert P. Wilson, and Monica S. Lam; Proceedings of the ACM SIGPLAN '95 conference on Programming language design and implementation, 1995, Pages 1-12

3) Improving Data-flow Analysis with Path Profiles, Glenn Ammons and James R. Larus, Proceedings of the ACM SIGPLAN '98 conference on Programming language design and implementation, 1998

Binary Instrumentation, Translation, and Optimization

1) Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation, C-K Luk, et. al., Proceedings of the ACM SIGPLAN '05 conference on Programming language design and implementation, 2005

2) Post-Pass Binary Adaptation for Software-Based Speculative Precomputation, Steve Liao, et. al., Proceedings of the ACM SIGPLAN '02 conference on Programming language design and implementation, 2002

3) IA-32 Execution Layer: a two-phase dynamic translator designed to support IA-32 applications on Itanium-based systems, Leonid Baraz, et. al., Proceedings. 36st Annual ACM/IEEE International Symposium on Microarchitecture, 2003

Dynamic Compilation and Optimization

1) A Survey of Adaptive Optimization in Virtual Machines, Matthew Arnold, et. al., Proceedings of IEEE, Vol. 93, No.2, February, 2005.

2) Dynamo: a transparent dynamic optimization system; Vasanth Bala,et. al., Proceedings of the ACM SIGPLAN '00 conference on Programming language design and implementation, 2000.

3) The Performance of Runtime Data Cache Prefetching in a Dynamic Optimization System, Jiwei Lu, et. al., MICRO'36, 2003

Profile Based Optimization

1) The Superblock: An Effective Technique for VLIW and Superscalar Compilation. W. W. Hwu, et. al., Journal of Supercomputing, 7(1,2):229--248, March 1993.

2) Profile guided code positioning; Karl Pettis and Robert C. Hansen; Proceedings of the conference on Programming language design and implementation, 1990, Pages 16 – 27

3) A Compiler Framework for Speculative Analysis and Optimizations, Jin Lin, et. al., Proceedings of the ACM SIGPLAN '03 conference on Programming language design and implementation, 2003.

4) Practical Structure Layout Optimization and Advice, Robert Hundt, et. al., Proceedings of the 4th Annual ACM/IEEE International Symposium on Code Generation and Optimization (CGO), 2006.

Compiler Optimization Techniques

1) Global code motion/global value numbering Cliff Click; Proceedings of the ACM SIGPLAN '95 conference on Programming language design and implementation, 1995, Pages 1-12

2) Register Allocation by Priority-based Coloring, Frederick Chow and John Hennessy, Proceedings of the ACM SIGPLAN '84 conference on Programming language design and implementation, 1984

3) Efficiently Computing Static Single Assignment Form and the Control Dependence Graph, Ron Cytron, et. al., ACM Transactions on Programming Languages and Systems, 1991

4) Cooperative prefetching: compiler and hardware support for effective instruction prefetching in modern processors; Chi-Keung Luk and Mowry, T.C.; Proceedings. 31st Annual ACM/IEEE International Symposium on Microarchitecture, 1998

5) Software Pipelining: An Effective Scheduling Technique for VLIW Machines, Monica Lam, Proceedings of the ACM SIGPLAN '88 conference on Programming language design and implementation, 1988

5/20/2009

Computational Geometry

1. Leonidas Guibas, John Hershberger, Daniel Leven, Micha Sharir and Robert E. Tarjan Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons; Algorithmica, Volume 2, Issue 1, pages 209 -- 233, 1987
   URL: http://www.springerlink.com/content/p25p124856385585

2. Hristo N. Djidjev, Andrzej Lingas, and Jörg-Rüdiger; Sack An O(n log n) algorithm for computing the link center of a simple polygon; Discrete and Computational Geometry, Volume 8, Issue 1, pages 131--152, 1992
   URL: http://www.springerlink.com/content/r2381w046021nt15

3. Hershberger, J. and Snoeyink, J. 1994. Computing minimum length paths of a given homotopy class. Comput. Geom. Theory Appl. 4, 2 (Jun. 1994), 63-97.
   DOI= http://dx.doi.org/10.1016/0925-7721(94)90010-8
   http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.7502&rep=rep1&type=pdf

4. Hershberger, J. and Suri, S. 1999. An Optimal Algorithm for Euclidean Shortest Paths in the Plane. SIAM J. Comput. 28, 6 (Aug. 1999), 2215-2256.
   DOI= http://dx.doi.org/10.1137/S0097539795289604
   http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.2037&rep=rep1&type=pdf

5. Suri, S., "On some link distance problems in a simple polygon," Robotics and Automation, IEEE Transactions on , vol.6, no.1, pp.108-113, Feb 1990
   URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=88124&isnumber=2879

6. Boaz Ben-Moshe, Matthew J. Katz, and Joseph S. B. Mitchell, A Constant-Factor Approximation Algorithm for Optimal Terrain Guarding In Proc. 16th ACM-SIAM Symposium on Discrete Algorithms, 2005.
   URL: http://www.ams.sunysb.edu/~jsbm/papers/soda05.pdf

7. Hershberger, J. and Suri, S. 1995. A pedestrian approach to ray shooting: shoot a ray, take a walk. J. Algorithms 18, 3 (May. 1995), 403-431. DOI=
   http://dx.doi.org/10.1006/jagm.1995.1017

8. Marshall Bern, David Dobkin, David Eppstein and Robert Grossman, Visibility with a moving point of view; Algorithmica, Volume 11, Issue 4, pages 360 -- 378, 1994
   URL: http://www.springerlink.com/content/t86661822674m652

9. Ghosh, S. K. and Mount, D. M. 1991. An output-sensitive algorithm for computing visibility. SIAM J. Comput. 20, 5 (Oct. 1991), 888-910. DOI=
   http://dx.doi.org/10.1137/0220055

Computer Vision

1) J. Aloimonos, I. Weiss, and A. Bandyopadhyay, ``Active Vision'', Int. Journal of Computer Vision, Vol. 1, No. 4, pp. 333-356, 1987.

2) J.K. Aggarwal and N. Nandhakumar, ``On the Computation of Motion from Sequences of Images'', Proc. IEEE, Vol. 76, No. 8, pp. 917-935, 1988.

3) R. Bajcsy, ``Active Perception'', Proc. IEEE, Vol. 76, No. 8, pp. 996-1005, 1988.

4) P. Belhumeur, J. Hespanha, and D. Kriegman, ``Eigenfaces vs. Fisherfaces: recognition using class specific linear projection'', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 711-720, 1997.

5) P.J. Burt and E.H. Adelson, ``The Laplacian Pyramid as a Compact Image Code'', IEEE Trans. Communications, Vol. COM-31, No. 4, pp. 532-540, 1983.

6) J. Canny, ``A Computational Approach to Edge Detection'', IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, 1986.

7) E.D. Dickmanns, B. Mysliwetz, and T. Christians, ``An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles'', IEEE Trans. on Systems, Man, and Cybernetics, Vol. 20, No. 6, pp. 1273-1284, 1990.

8) J. Gluckman and S. Nayar, ``Rectified catadioptric stereo sensors'', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, No. 2 , pp. 224-236, 2002.

9) B.K.P. Horn, `` Extended Gaussian Images'', Proc. IEEE, Vol. 72, No. 12, pp. 1671-168 6, 1984.

10) B.K.P Horn and B.G. Schunck, ''Determining Optical Flow'', Artificial Intelligence, Vol. 17, Nos. 1-3, pp. 185-204, 1981.

11) B. Kass, A. Witkin, and D. Terzopoulos, ``Snakes: Active Contour Models'', Int. Journal of Computer Vision, Vol. 1, No. 4, pp. 321-331, 1987.

12) E. Krotkov, ``Focusing'', Int. Journal of Computer Vision, Vol. 1, No. 3, pp. 223-237 , 1987.

13) R.K. Lenz and R.Y. Tsai, ``Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology'', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 5, pp. 713-720, 1988.

14) L. Matthies, T. Kanade, and R. Szeliski, ``Kalman Filter-Based Algorithms for Estimat ing Depth from Image Sequences'', Int. Journal of Computer Vision, Vol. 3, pp. 209-236, 19 89.

15) H.P. Moravec, ``The Stanford Cart and the CMU Rover'', Proc. IEEE, Vol. 71, No. 7, pp . 872-884, 1983.

16) S. Nayar, S. Nene, and H. Murase, ``Real-time 100 object recognition system '', Proc. of IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 2321-2325, 1996.

17) S. Mori, C.Y. Suen, and K. Yamamoto, ``Historical Review of OCR Research and Development'', Proc. IEEE, Vol. 80, No. 7, pp. 1029-1058, 1992.

18) Y. Ohta and T. Kanade, ``Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming'', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 7, No. 2, pp. 139-154, 1985.

19) N. Otsu, ``A Threshold Selection Method from Gray-Level Histograms'', IEEE Trans. on Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62-65, 1979.

20) C. Tomasi and T. Kanade, `` Shape and Motion from Image Streams under Orthography: A Factorization Method'', Int. Journal of Computer Vision, Vol. 9, No. 2, pp. 137-154, 1992.

21) J.A. Walter and K.J. Schulten, ``Implementation of Self-Organizing Neural Networks for Visuo-Motor Control of an Industrial Robot'', IEEE Trans. Neural Networks, Vol. 4, No. 1, pp. 86-96, 1993.

Data Mining

1. A.K. Jain et al., Data Clustering: A Review , ACM Computing Surveys (CSUR), Volume 31 , Issue 3 (September 1999),
2. B.W.Wang, Yang, R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining. VLDB 1997.
3. A.G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases, 24th International Conference on Very Large Data Bases, New York City. 1998.
4. F. S. K. Murthy, Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery 2(4): 345-389, 1998
5. G. Karypis, E. Han and V. Kumar, CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer 1999, (http://www.cs.umn.edu/~karypis/publications/Papers/PDF/chameleon.pdf)
6. J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, `` PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth '', Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001.
7. J. Srivastava, R. Cooley, M. Deshpande, P.-N. Tan, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, in SIGKDD Explorations, Vol. 1, Issue 2, 2000. (http://www.acm.org/sigkdd/explorations/issue1-2/srivastava.pdf)
8. K. H. Tung, J. Hou, and J. Han, `` Spatial Clustering in the Presence of Obstacles '', Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001.
9. K. P. Bennett, C. Campbell, Support Vector Machines: Hype or Hallelujah? in SIGKDD Explorations, Vol. 2, Issue 2, 2000. (http://www.acm.org/sigs/sigkdd/explorations/issue2-2/bennett.pdf)
10. Kjersti Aas, Microarray Data Mining: A Survey, January 2001 (http://www.nr.no/documents/samba/research_areas/SIP/microarraysurvey.pdf)
11. M. Deshpande and G. Karypis, Selective Markov Models for Predicting Web-Page Accesses, 1st SIAM Data Mining Conference, 2001. (http://www.cs.umn.edu/~karypis/publications/Papers/PDF/select.pdf)
12. M. Seno and G. Karypis, LPMiner: An Algorithm for Finding Frequent Itemsets Using Length Decreasing Support Constraint. (1st IEEE Conference on Data Mining) (http://www.cs.umn.edu/~karypis/publications/Papers/PDF/lpm.pdf)
13. Mahesh Joshi, Ramesh Agarwal, Vipin Kumar, Mining Needle in a Haystack: Classifying Rare Classes via Two-phase Rule Induction, SIGMOD-2001 May 2001.
14. Pang Tan and Vipin Kumar, Modeling of Web Robot Navigational Patterns, Data Mining and Knowledge Discovery, Volume 6, Number 1, pp 9-35, 2002.
15. Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava, Indirect Association: Mining Higher Order Dependencies in Data, Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lyon, France, pp 632-637, Sept 13-16, 2000.
16. Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava, Selecting the Right Interestingness Measures for Association Patterns, Computer Science Dept Tech Report, February 2002.
17. Pang-Ning Tan, Vipin Kumar, Mining Association Patterns in Web Usage Data, International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet, L'Aquila, Italy, January (2002).
18. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, (http://www.almaden.ibm.com/cs/quest/publications.html)
19. R. Grossmand, C. Kamath, W. P. Kegelmeyer, V. Kumar, and R. Namburu, Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers, 2001.

20. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", Proc. of the ACM SIGMOD Int'l Conference on Management of Data, Seattle, Washington, June 1998.
21. S. Arnborg, A Survey of Bayesian Data Mining - Discrete and semi-discrete Data Matrices, Swedish Institute of Computer Science November 15, 2001. (http://www.google.com/search?q=%22Data+Mining%22+Survey&hl=en&start=40&sa=N)
22. S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation, in SIGKDD Explorations, Vol. 2, Issue 2, 2000. (http://www.acm.org/sigs/sigkdd/explorations/issue2-2/contents.htm#Bay)
23. S. Shekhar and Y. Huang, Discovering Spatial Co-location Patterns: A Summary of Results, Proc. of 7th International Symposium on Spatial and Temporal Databases (SSTD01), L.A., CA, July 2001
24. S. Shekhar, C.T. Lu, P. Zhang, A Unified Approach to Spatial Outliers Detection Technical Report TR01-045, Computer Science Dept., U. of Minnesota, submitted to the IEEE Transactions on Knowledge and Data Engineering (A Summary of results appeared in SIGKDD 2000).
25. S. Shekhar, P. Schrater, W. R. Raju, W. Wu, Spatial Contextual Classification and Prediction Models for Mining Geospatial Data, to appear in IEEE Transactions on Multimedia (special issue on Multimedia Databases), 2002
26. V. Guralnik, J. Srivastava, Event Detection from Time Series Data, in the Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), San Diego, 1999.
27. Ying Zhao and G. Karypis, Criterion Functions for Document Clustering: Experiments and Analysis. University of Minnesota, TR 01-40. (http://www.cs.umn.edu/~karypis/publications/Papers/PDF/vscluster.pdf)

Databases

"Readings in Database Systems", 3rd edition; edited by Michael Stonebraker and Joseph M. Hellerstein; published by Morgan Kaufmann, Inc.

Distributed Systems

Distributed System Architectures:

1) J. H. Saltzer, D. P. Reed, and D. D. Clark. “End-to-end arguments in system design,” ACM Transactions on Computer Systems, 2(4): 277--288, November 1986.

2) A. Birrell and B. Nelson. “Implementing Remote Procedure Calls,” ACM Transactions on Computer Systems, 2(1): 39--59, February 1984.

3) A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier. “Cluster-Based Scalable Network Services,” In Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles, pages 78--91, Saint-Malo, France, October 1997.

4) I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. “Chord: A scalable peer-to-peer lookup service for Internet applications,” In Proceedings of ACM SIGCOMM 2001.

Mobile and Wireless Computing:

1) G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, Vol. 27, No. 4, (April 1994), pp. 38-47 .

2) Imielinski, T., and Badrinath, B.R., “Mobile Wireless Computing: Challenges in Data Management,” CACM Vol. 37, No. 10, pp. 18-28.

3) M. Satyanarayanan, “Pervasive Computing: Vision and Challenges,” IEEE Personal Communications, August 2001, pp. 10-17.

4) R. Caceres, L. Iftode, "The Effects of Mobility on Reliable Transport Protocols," in Proc. Of the 14th International Conference on Distributed Computing Systems, 1994.

5) B. Noble, M. Satyanarayanan, D. Narayanan, J. Tilton, J. Flinn, K. Walker "Agile Application-Aware Adaptation for Mobility," Proceedings of the Sixteenth ACM Symposium on Operating Systems.

Distributed File/Storage Systems and Data Management

1) Anderson, T. et. al., “Serverless Network File Systems,” ACM Transactions on Computer Systems, February 1996.

2) M. Spasojevic and M. Satyanarayanan, "An Empirical Study of a Wide-Area Distributed File System," ACM Transactions on Computer Systems, Vol. 14, No.2, May 1996.

3) Eliezer Levy, Abraham Silberschatz: “Distributed File Systems: Concepts and Examples,” ACM Computing Surveys 22(4): 321-374 (1990)

4) John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao. “OceanStore: An Architecture for Global-Scale Persistent Storage,” Proceedings of the Ninth international Conference Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November 2000.

System Performance, Scheduling/Load Balancing, Process Management in Distributed Systems:

1) Carl A. Waldspurger, William E. Weihl, “Lottery Scheduling: Flexible Proportional-Share Resource Management,” OSDI 1994.

2) Eager, Lazowska, Zahorjan, “Adaptive Load Sharing in Homogeneous Distributed Systems,” IEEE Trans SE, 12(5),” 1986.
3) Krishna Kant, Youjip Won, “ Server Capacity Planning for Web Traffic Workload,” IEEE TKDE 11(5): 731-747 (1999).

4) M. J. Litzkow, M. Livny, and M. W. Mutka. “Condor---A Hunter of Idle Workstations,” in Proc. of the 8th Int'l Conf. on Distributed Computing Systems, pages 104--111, 1988.

5) F. Douglis and J. Ousterhout, "Transparent process migration - design alternatives and the Sprite implementation", Software - Practice & Experience, vol. 21, no. 8, pp. 757-785, (1991).

Foundations of Distributed Computing Systems:

1) Leslie Lamport, "Time, Clocks, and Ordering of Events in Distributed Systems", Communications of the ACM, July 1978.

2) J. Fidge, "Timestamps in the Message Passing System that Preserve the Partial Ordering," Proc. of the 11'th Australian Computer Science Conference, pp. 56-66 Vol. 10, No. 1, February 1988.

3) K. Birman, Andre Schiper, Pat Stephenson, "Lightweight causal and group multicast," ACM Transactions on Computer Systems, Vol. 9, No.3, 1991, pp. 272-314.

4) Fred Mattern, "Algorithms for Termination Detection", Distributed Computing, 1987.

5) F. Cristian, "Understanding Fault-Tolerant Distributed Systems," Communications of the ACM, February 1991, pp. 56-78.

Security and Protection

1) R.M. Needham and M.D. Schroeder, “Using Encryption for Authentication in Large Network of Computers,” Communications of the ACM, December 1978.

2) Radia Perlman, “An Overview of PKI Trust Models,” IEEE Network, November/December 1999, pages 38-43.

3) David K. Gifford , “Cryptographic sealing for information secrecy and authentication,” Communications of the ACM, April, 1982 , 25(4).

4) Steiner, J. G., Neumann, C., and Schiller, J. I., Kerberos: An Authentication System for Open Network Systems. In Procedings of the Winter 1988 USENIX Conference, pages 191-202.

Graphics

Local Reflection Models and Shading

1) Cook R. L. and Torrance K. E. (1982). A Reflection Model for Computer Graphics. Computer Graphics, 15 (3), 51-72.

2) Cook R. L. (1984). Shade Trees. Computer Graphics, 18 (3), 223-231.

3) Ward G. J. (1992). Measuring and Modeling Anisotropic Reflection. Computer Graphics, 26 (4), 265-272.

Global Illumination Models

1) Whitted T. (1980). An Improved Illumination Model for Shaded Display. Communications of the ACM, 23 (6), 343-349.

2) Goral C. M., Torrance K. E., Greenberg D. P., and Battaile B. (1984). Modeling the Interaction of Light Between Diffuse Surfaces. Computer Graphics, 18 (3), 213-222.

3) Cook R. L. (1986). Stochastic Sampling in Computer Graphics. ACM Transactions on Graphics, 5 (1), 51-72. Kajiya J. T. (1986). The Rendering Equation. Computer Graphics, 20 (4), 143-150.

4) Cohen M., Chen S. E., Wallace J. R., and Greenberg D. P. (1988). A Progressive Refinement Approach to Fast Radiosity Image Generation. Computer Graphics, 22 (4), 75-84.

Perceptually Based Rendering

1) Tumblin J. and Rushmeier H. (1993). Tone Reproduction for Realistic Images. IEEE Computer Graphics and Applications, 13 (6), 42-48.

2) Ferwerda J. A., Pattanaik S. N., Shirley P., and Greenberg D. P. (1996). A Model of Visual Adaptation for Realistic Image Synthesis. Proceedings of SIGGRAPH'96, 249-258.

Texture Mapping

1) Blinn J. F. and Newell M. E. (1976). Texture and Reflection in Computer Generated Images. Comm. ACM, 19 (10), 542-547.

2) Williams L. (1983). Pyramidal Parametrics. Computer Graphics, 17 (3), 1-11.

Texture Synthesis

1) Perlin K. (1985). An Image Synthesizer. Computer Graphics, 19 (3), 287-296.

2) Heeger D. J. and Bergen J. R. (1995). Pyramid-Based Texture Analysis/Synthesis. Proceedings of SIGGRAPH'95, 229-238.

Non-Photorealistic Rendering

1) Haeberli, P. (1990). Paint by Numbers: Abstract Image Representations. Proceedings of SIGGRAPH'90, 207-214.

2) Winkenbach, G. and Salesin, D. H. (1994). Computer-Generated Pen-and-Ink Illustration. Proceedings of SIGGRAPH'94, 91-100.

Image Techniques

1) Porter T. and Duff T. (1984). Compositing Digital Images. Computer Graphics, 18 (3), 253-259.

2) Chen S. E. and Williams L. (1993). View Interpolation for Image Synthesis, Proceedings of SIGGRAPH'93, 279-288.

Multi-Resolution Representation

1) Clark J. H. (1976). Hierarchical Geometric Models for Visible Surface Algorithms, Communications of the ACM, 19 (10), 547-554.

2) Schroeder W. J., Zarge J. A., and Lorensen W. E. (1992) Decimation of Triangle Meshes, Proceedings of SIGGRAPH'92, 65-70.

HCI/GUI

1) Vannevar Bush 1945. As we may think. In Atlantic Monthly, volume 176, pages 101-108.

2) Jonathan Grudin 1988. Why cscw applications fail: Problems in the design of evaluation of organizational interfaces. In Proceedings of CSCW '88, pages 85-93.

3) Henry Kautz, Bart Selman, and Mehul Shah 1997. Referral Web: Combining social networks and collaborative filtering. Communications of the ACM, 40(3):63-65.

4) Goldberg, D., Nichols, D., Oki, B. M. and Terry, D 1992. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 35(12):61-70.

5) Sketchpad: A Man-Machine Graphical Communication System. Ivan Sutherland (SJCC 1963)

6) Pick -and-Drop: a direct manipulation technique for multiple computer environments. Jun Redimotot (UIST 1997)

7) Chapters 4,5, and 6 of Usability Engineering. Jacob Nielsen

8) Augmenting the organizational memory: a field study of answer garden. Mark Ackerman (ACM Trans. on Information Systems, July 1998)

9) Perceptual user interfaces: multimodal interfaces that process what comes naturally. Sharon Oviatt and Philip Cohen (CACM, March 2000)

10) Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces. Brad A. Meyers, Dario Giuse, Roger B Dannenberg, Brad Vander Zanden, David Kosbie, Ed Pervin, Andrew Mickish, and Philippe Marchal (IEEE Computer, November 1990)

11) Pad: an alternative approach to the computer interface. Ken Perlin and David Fox (SIGGRAPH) 1993

12) Visual information seeking: tight coupling of dynamic query filters with starfield displays. Christopher Ahlberg and Ben Shneiderman (CHI 1994)

13) The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. Ramana Rao and Stuart K. Card (CHI 1994)

14) Linking and messaging from real paper in the Paper PDA. Jeremy M. Heiner, Scott E. Hudson and Kenichiro Tanaka (UIST 1999)

15) Interaction and Outeraction: Instant Messaging in Action. Bonnie Nardi, Steve Whittaker, and Erin Bradner (CSCW 2000)

16) The Keystroke-Level Model for User Performance Time with Interactive Systems. Stuart K. Card, Thomas P. Moran, and Allen Newell (CACM, July 1980)

17) Agents that reduce work and information overload. Pattie Maes (CACM, July 1994)

18) The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site. Ed H. Chi, Peter Pirolli and James Pitkow (CHI 2000)

19) Adtranz: A Mobile Computing System for Maintenance and Collaboration. Dan Siewiorek, Asim Smailagic, Len Bass, Jane Siegel, Richard Martin, Ben Bennington (ISWC 1998)

Intelligent Agents

Read from the book:

Multiagent systems. A modern approach to Distributed Artificial Intelligence." edited by Gerhard Weiss.
The MIT Press, 1999. ISBN: 0262731312

the following Chapters:

Chapter 1 - Intelligent Agents.
Chapter 2 - Multiagent Systems and Societies of Agents.
Chapter 3 - Distributed Problem Solving and Planning.
Chapter 5 - Distributed Rational Decision Making.
Chapter 6 - Learning in Multiagent Systems.

and these papers:

Philip R Cohen and Hector J Levesque, Intention is choice with commitment, Artificial Intelligence, Vol. 42, No 2-3 (March 1990)

Jelle R. Kok and Nikos Vlassis, Sparse cooperative Q-learning.
Proc. of the 21st Int. Conf. on Machine Learning, pp. 481-488, ACM, Banff, Canada, July 2004.

Martha Pollack and John Horty, There's More to Life than Making
Plans: Plan Management in Dynamic, Multi-agent Environments, AI Magazine, Vol 20, No 4, pp 71-83, Winter 1999.

Jeffrey Rosenschein and Gilad Zlotkin, Designing Conventions for Automated Negotiation, AI Magazine, Fall 1994, pp 29-46.

Tuomas Sandholm, Algorithm for Optimal Winner Determination in Combinatorial Auctions. Artificial Intelligence, Vol 135, pp 1-54, 2002.

Y. Shoham and M. Tennenholtz, On Social Laws for Artificial Agent Societies:
Off-Line Design, Artificial Intelligence, Vol. 73, 1995

M. Tambe et al. Conflicts in teamwork: hybrids to the rescue AAMAS 2005, pp 3-13.

Math Models for Data Mining

1) C. Couvreur and Y. Bresler. On the optimality of backward greedy algorithm for the subset selection method. SIAM J. Matrix Analysis and Applications, 21:797-808, 2000. http://epubs.siam.org/sam-bin/dbq/article/33292

2) B. Natarajan. Sparse approximate solutions to linear systems. SIAM J. Computing, 24:227-234, 1995.????

3) M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3:71-86, 1991.

4) M.W. Berry, S.T. Dumais and G.W. O'Brien. Using linear algebra for intelligent information retrieval. SIAM Review, 37:573-595, 1995.

5) T. Kolda and D. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing in information retrieval. ACM Trans. Information Systems, 16:322-346, 1998.

6) "Two Algorithms for Nearest-Neighbor Search in High Dimensions", Jon M. Kleinberg, ACM Symposium on Theory of Computing, pp. 599-608, 1997. citeseer.nj.nec.com/kleinberg97two.html

7) Authoritative sources in a hyperlinked environment. J. Kleinberg. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997. PDF format also available. http://www.cs.cornell.edu/home/kleinber/auth.ps http://www.cs.cornell.edu/home/kleinber/auth.pdf

8) Scatter/Gather: a cluster-based approach to browsing large document collections Douglass R. Cutting, David R. Karger, Jan O. Pedersen and John W. Tukey Proceedings of the Fifteenth Annual International ACM SIGIR conference on Research and development in information retrieval June 21 - 24, 1992, Copenhagen Denmark pp. 318-329. http://www.acm.org/pubs/articles/proceedings/ir/133160/p318-cutting/p318-cutting.pdf

9) Concept Decompositions for Large Sparse Text Data using Clustering. I.S. Dhillon and D.S. Modha. Machine Learning, 42:1, pages 143-175, January, 2001 (an earlier version appears as IBM Research Report RJ 10147, July 8, 1999). early version: http://www.cs.utexas.edu/users/inderjit/public_papers/concept_mlj.ps.gz

10) Principal Direction Divisive Partitioning, D. L. Boley, Data Mining and Knowledge Discovery 2(4):325-344, 1998. ftp://ftp.cs.umn.edu/dept/users/boley/reports/PDDP.ps.gz

11) Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition, John Conroy. Dianne P. O'Leary, CS-TR-4221, February 2001. http://www.cs.umd.edu/Library/TRs/CS-TR-4221/CS-TR-4221.ps.Z

12) Nonparametric discriminant analysis, K. Fukunaga and J.M. Mantock. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5:671-678, 1983.

13) Dimension reduction for text data representation based on cluster structure preserving projection
H. Park, M. Jeon, and P.J. Howland. Technical report, Department of Computer Science and Engineering, University of Minnesota, March 2001 https://wwws.cs.umn.edu/tech_reports/listing/?year=2001
Report number 01-013 (a revised version will appear in SIAM Journal on Matrix Analysis and Applications. But the above technical report will be used until the paper appears in the journal)

14) Lower dimensional representation of text data in vector space based information retrieval,
H. Park, M. Jeon, and J.B. Rosen, Proceedings of the first computational information retrieval workshop, SIAM, pp. 7-24, 2000.

15) Class Visualization of High-Dimensional Data with Applications. I.S. Dhillon, D.S. Modha, W.S. Spangler, submitted for publication, 1999.
http://www.cs.utexas.edu/users/inderjit/public_papers/cviz_journal.ps

16) The PageRank Citation Ranking: Bringing Order to the Web.Page, Lawrence; Brin, Sergey; Motwani, Rajeev; Winograd, Terry. Stanford Univ. Computer Science Dept technical report. Oct. 2001
http://dbpubs.stanford.edu/pub/1999-66

17) Support Vector Machines: Hype or Hallelujah?, K. P. Bennett, C. Campbell SIGKDD Explorations, Vol. 2, Issue 2, 2000.
http://www.acm.org/sigs/sigkdd/explorations/issue2-2/bennett.pdf

18) An Introduction to Locally Linear Embedding. Lawrence Saul & Sam Roweis. [draft version (Jan.01)
http://www.cs.toronto.edu/%7Eroweis/lle/papers/lleintro.pdf

19) The Anchors Hierarchy: Using the Triangle Inequality to Survive High Dimensional Data,Andrew W. Moore, Proceedings of UAI-2000: The Sixteenth Conference on Uncertainty in Artificial Intelligence
http://www.cs.cmu.edu/afs/cs.cmu.edu/project/reinforcement/papers/anchors.ps

20) Link Analysis, Eigenvectors and Stability, Andrew Y. Ng and Alice X. Zheng and Michael I. Jordan. IJCAI 2001, p 903-910.
http://citeseer.nj.nec.com/ng01link.html

21) Kernel principal component analysis, B. Scholkopf, A. Smola, and K.-R. Muller. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors: Advances in Kernel Methods - SV Learning, pages 327-352. MIT Press, Cambridge, MA, 1999b. http://citeseer.nj.nec.com/25296.html

Networks

Internet Routing and Interconnection

1) "Interconnection, Peering and Settlements" Part I-II, Geoff Houston, The Internet Protocol Journal, CISCO. no.1-2, vol 2, 1999.

2) "Internet Routing Instability," Craig Labovitz, G. Robert Malan, and Farnam Jahanian (University of Michigan) , SIGCOMM'97

Congestion Control

1) "Congestion Advoidance and Control" V. Jacobson, Proc. SIGCOMM 88

2) "Congestion Control Principles" Floyd, S. RFC 2914, Best Current Practice, September 2000.

Quality-of-Service

1) "An Architecture for Differentiated Services" RFC 2475 , S. Blake et al, Dec 1998.

2) G. Huston, "Next Steps for the IP QoS Architecture", RFC 2990, November 2000.

The Future of Internet and Networking Research

1) B. Braden et al., "Developing a Next-Generation Internet Architecture", July 2000

2) Marjory S. Blumenthal and David D. Clark, "Rethinking the design of the Internet: The end to end arguments vs. the brave new world". Transactions on Internet Technology.

3) "Looking Over the Fence at Networks: A Neighbor's View of Networking Research" (2001). CSTB Annual Convening on Research Horizons, January 2001.

Optical Networks

1) “All Optical Networks,” NSF Document

2) “IP over All-Optical Networks – Issues,” A. Durresi, Raj Jain, N. Chandhok, R. Jagannathan, S. Seetharaman and K. Vinodkrishnan, Global Telecommunications Conference, 2001, Vol. 4, pp. 2144-2149

Steaming Video over Internet

1) “Reducing Bandwidth Requirement for Delivering Video over Wide Area Networks with Proxy Server,” W. Ma and D. Du, to appear in IEEE Trans. on Multimedia

2) “Save: An Algorithm for Smoothed Adaptive Video over Explicit Rate Networks,” IEEE /ACM Trans. on Networking, pp. 717-728, Dec. 1998

Wireless Networks

1) “A Survey on Position-Based Routing in Mobile Ad Hoc Networks,” M. Mauve, J. Widmer and H. Hartenstein, IEEE Network, Nov/Dec 2001, pp. 30-39

2) “A Comparison of Mechanisms for Improving TCP Performance over Wireless Links,” H. Balakrishnan, V. Pandmanabhan, S. Seshan, and R. Katz, IEEE/ACM Trans. on Networking, December 1997

Parallel Algorithms

1) Parallel Algorithms for Sequence Mining Valerie Guralnik, Nivea Garg, and George Karypis Proceedings of the Europar 2001. (http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/europartp.pdf)

2) ScalParC: A New Scalable and Efficient Parallel Classification Algorithm for Mining Large Datasets. Mahesh Joshi, George Karypis, and Vipin Kumar.
(http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/scalparc.pdf)

3) Scalable Parallel Data Mining for Association Rules. Eui-Hong (Sam) Han, George Karypis and Vipin Kumar. (http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/assoc-parallel-journal.pdf)

4) A Coarse-Grain Parallel Formulation of a Multilevel k-way Graph Partitioning Algorithm. George Karypis and Vipin Kumar. (http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/mlevel-mpi.pdf)

5) Highly Scalable Parallel Algorithms for Sparse Matrix Factorization Anshul Gupta, George Karypis, and Vipin Kumar (http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/sparse-cholesky.pdf)

6) Introduction to Parallel Computing: Design and Analysis of Algorithms by Vipin Kumar, Ananth Grama, Anshul Gupta and George Karypis, Benjamin-Cummings Publishing Company, November 1993.

7) Kirk Schloegel, George Karypis and Vipin Kumar, Parallel Multilevel Algorithms for Multi-constraint Graph Partitioning Concurrency Practice & Experience, 2001.

8) Kirk Schloegel, George Karypis and Vipin Kumar, Wavefront Diffusion and LMSR: Algorithms for Dynamic Repartitioning of Adaptive Meshes, IEEE Transactions on Parallel and Distributed Systems Volume 12, Number 5, pages 451-466, May 2001.

9) Scalable Parallel Data Mining for Association Rules, E.H. Han and G. Karypis and V. Kumar, IEEE Transactions on Knowledge and Data Engineering, Volume 12, Number 3, May/June 2000, pp 337-352.

10) George Karypis and Vipin Kumar, Parallel Multilevel K-way Partitioning Scheme for Irregular Graphs, SIAM Review, Volume 41, Number 2, 1999.

11) Ananth Grama and Vipin Kumar, Parallel Search Methods - State-of-the-Art, IEEE Transactions on Knowledge and Data Engineering Volume 11, Number 1, 1999, pp 28-35.

12) A. Srivastava, E.H. Han, V. Kumar, and V. Singh, Parallel Formulations of Decision-Tree Classification Algorithms, Data Mining and Knowledge Discovery: Special Issue on Scaling Data Mining Algorithms, Applications and Systems Volume 3, Issue 3, September 1999; pp. 237-261.

13) Ananth Y. Grama, Vipin Kumar and Ahmed Sameh, Scalable Parallel Formulations of the Barnes-Hut Method for n-Body Simulations, Parallel Computing, Volume 24, Number 5-6, June 1998, pp 797-822.

14) Anshul Gupta, George Karypis and Vipin Kumar, A Highly Scalable Parallel Algorithm for Sparse Matrix Factorization, IEEE Transactions on Parallel and Distributed Systems Volume 8, Number 5, May, 1997, pp 502-520. A short version of this paper won the Outstanding Student Paper Award from the Supercomputing 94 conference.

15) S. Shekhar, S. Ravada, V. Kumar, G. Turner, and D. Chubb, High Performance Geographic Information Systems: Experiences with a Shared-Memory Multiprocessor, IEEE Computer, Volume 29, Number 12, pp 42-49, December 1996.

16) V. Kumar and A. Gupta, "Analyzing the Scalability of Parallel Algorithms and Architectures: A Survey", Journal of Parallel and Distributed Computing (special issue on scalability) Volume 22, Number 3, September 1994, pp. 379-391. Also available as Tech Report TR 91-18, Department of Computer Science, University of Minnesota, 1991.

17) Ananth Grama, Anshul Gupta, and Vipin Kumar, "Isoefficiency Function: A Scalability Metric for Parallel Algorithms and Architectures", IEEE Parallel and Distributed Technology, Special Issue on Parallel and Distributed Systems: From Theory to Practice, August 1993, Volume 1, Number 3, pp 12-21.

18) A. Gupta and V. Kumar, "Scalability of FFT on Parallel Computers", IEEE Transactions on Parallel and Distributed Systems August 1993, Volume 4, Number 8, pp 922-932.

19) R. Metzger, B. VanVoorst, L. Pires, R. Jha, W. Au, M. Amin, D. Castanon, V. Kumar, The C3I Parallel Benchmark Suite - Introduction and Preliminary Results, Proceedings of Supercomputing '96, Pittsburgh, November 1996.

20) Ananth Grama, Anshul Gupta, Eui-Hong Han, and Vipin Kumar, "Parallel Algorithm Scalability Issues in Petaflops Architectures", Tom Sterling, et.al. (editors), "Topics in Ultra-Scale Computing" MIT Press, 2001.

21) K. Schloegel and G. Karypis and V. Kumar, "Graph Partitioning for High Performance Scientific Simulations" J. Dongarra and I. Foster and G. Fox and K. Kennedy and A. White (editors), "CRPC Parallel Computing Handbook", Morgan Kaufmann, 2001.

22) Mahesh V. Joshi, Eui-Hong (Sam) Han, George Karypis, Vipin Kumar, "Efficient Parallel Algorithms for Mining Associations", M. Zaki, C.-T. Ho (editors), "Large-scale Parallel and Distributed Data Mining", Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI), vol. 1759, Springer-Verlag, 2000.

Programming Languages

This reading list is broken up into two parts, the first of which is common to the in-class and take-home major exams and some further readings for the take home part only. The material in the relevant list of papers and books should be understood as *basic* for each kind of major exam in the Programming Languages area. Questions in the in-class exam will probe understanding of the area obtained from familiarity with this material. The take-home exam may involve specific additional reading to be done during the exam period and will also call for critical thinking in a more relaxed timeframe than afforded by an in-class exam.

No single course covers all the topics in the listed papers. The way to interpret this reading list and, hence, the depth exam in Programming Languages is that you are expected through independent study to familiarize yourself with material beyond what you are exposed to in CSCI 5106. The textbook by Sethi thus defines the point of departure. The additional papers indicate some of the important themes that you should have a grasp of beyond what is in that text.

Common Readings for the In-Class and Take-Home Exams

1) A.W. Appel, Garbage collection, in *Topics in Advanced Language Implementation*, P. Lee (ed.), pages 89-100, MIT Press, 1991.

2) J. Bentley, Programming pearls, *CACM* 26(12): 1040-1045, December 1983.

3) L. Damas and R. Milner, Principal type-schemes for functional programs, *Ninth ACM Symposium on Principles of Programming Languages*, 1982, pages 207-212. (Make sure to know how to complete the proofs in this paper and also to understand how all this relates to type inference in ML. A slower paced reference can also be provided if necessary.)

4) C.A.R. Hoare, An axiomatic basis for computer programming, *CACM* 12(10): 576-580, October 1969.

5) C.A.R. Hoare, Communicating Sequential Processes, *CACM* 21(8): 666-777, August 1978.

6) C.A.R. Hoare, Hints on programming language design. In *Programming Languages: A Grand Tour*, Horowitz, E. (ed.), pp 31-40, Computer Science Press, Rockville, Maryland, 1987.

7) P.J. Landin, The mechanical evaluation of expressions, *Computer Journal* 6(4): 308-320, Jan 1964.

8) P.J. Landin, The next 700 programming languages, *CACM* 9(3): 157-166, March 1966.

9) J. McCarthy, Recursive functions of symbolic expressions and their computation by machine, Part I, *CACM* 3(4): 184-195, 1960.

10) R. Sethi, *Programming Languages: Concepts and Constructs (Second Edition),* Addison-Wesley, 1996.

11) R.D. Tennent, The denotational semantics of programming languages, *CACM* 19(8): 437-453, 1976.

12) D. Friedman and D. Wise, CONS should not evaluate its arguments, in *Automata, Languages and Programming*, Edinburgh University Press, 1976, pages 257-284.

13) N. Wirth, On the design of programming languages, IFIP Congress 74, pages 386-393, North Holland Publishing Company, Amsterdam. Also reprinted in *Programming Languages: A Grand Tour*, Horowitz, E. (ed.), pp 23-30, Computer Science Press, Rockville, Maryland, 1987.

Further Readings for the Take-Home Exam

1) K.R. Apt and M.H. van Emden, Contributions to the theory of logic programming, *JACM* 29(3): 841-862, July 1982.

2) T. Johnsson, Lambda lifting: transforming programs to recursive equations, *Proceedings of the IFIP Conference on Functional Programming and Computer Architecture*, J.-P. Jouannaud (ed.), Lecture Notes in Computer Science, Volume 201, pages 190-205, Springer-Verlag, 1985.

3) G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C.V. Lopes, J.-M. Loingtier, J. Irwin, Aspect-Oriented Programming, *Proceedings of the European Conference on Object-Oriented Programming*, Lecture Notes in Computer Science Volume 1241, pages 220-242, Springer-Verlag, June 1997.

4) D.E. Knuth, Semantics of Context-free Languages. *Mathematical Systems Theory* 2(2): 127--145, 1968. Corrections in *Mathematical Systems Theory* 5(2):95--96, 1971.

5) H. Xi and F. Pfenning, Eliminating array bound checking through dependent types, *Proceedings of the ACM SIGPLAN '98 Conference on Programming Language Design and Implementation*, pages 249-257, ACM Press, 1998.

Security/Cryptography

1. A note on the confinement problem, BW Lampson, Communications of the ACM, 1973

2. New directions in cryptography, W Diffie, ME Hellman, IEEE Transactions on Information Theory, 1976

3. Using encryption for authentication in large networks of computers, RM Needham, MD Schroeder, Communications of the ACM, 1978

4. A method for obtaining digital signatures and public-key cryptosystems, RL Rivest, A Shamir, L Adleman, Communications of the ACM, 1978

5. Formal Models for Computer Security, CE Landwehr, ACM Computing Surveys, 1981

6. A Note on the Denial-of-Service Problem, VD Gligor, Proceedings of the IEEE Symposium on Security and Privacy, 1983

7. End-to-end arguments in system design, JH Saltzer, DP Reed, DD Clark, ACM Transactions on Computer Systems, 1984,

8. How to prove yourself: Practical solutions to identification and signature problems, A Fiat, A Shamir, Crypto, 1986

9. An intrusion-detection model, DE Denning, IEEE Transactions on Software Engineering, 1987

10. A digital signature scheme secure against adaptive chosen-message attacks, S Goldwasser, S Micali, R Rivest, SIAM Journal of Computing, 1988

11. Encrypted key exchange: Password-based protocols secure against dictionary attacks, SM Bellovin, M Merritt, Proceedings of the IEEE Conference on Security and Privacy, 1992

12. Random oracles are practical: A paradigm for designing efficient protocols, M Bellare, P Rogaway, First ACM Conference on Computer and Communications Security, 1993

13. A Cryptographic File System for Unix, M Blaze, Proceedings of the First ACM Conference on Computer and Communication Security, 1993

14. Efficient software-based fault isolation, R Wahbe, S Lucco, TE Anderson, SL Graham, Proceedings of the 14th ACM Symposium on Operating Systems, 1993

15. Kerberos: An Authentication Service for Computer Networks, BC Neuman, T Ts'o, IEEE Communications, 1994

16. Access control: Principles and practice, R Sandhu, P Samarati, IEEE Communications, 1994

17. Programming Satan's computer, R Anderson, R Needham, Computer Science Today, 1995

18. The SSL protocol version 3.0. Internet draft (draft-freier-ssl-version3-02.txt), Alan O. Freier, Philip Karlton, and Paul C. Kocher. Network Working Group, November 1996. Work in progress.

19. Tamper Resistance - A Cautionary Note, R Anderson, M Kuhn, Usenix Electronic Commerce Workshop, Oakland, 1996

20. The decision Diffie-Hellman problem, D Boneh, Proceedings of the Third Algorithmic Number Theory Symposium, 1998

21. Smashing the stack for fun and profit, Aleph One, Phrack Magazine, 1998

22. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0, A Whitten, JD Tygar, Proceedings of the 8th USENIX Security Symposium, 1999

23. Privacy-preserving data mining, R Agrawal, R Srikant, ACM SIGMOD International conference on Management of data, 2000

24. ID-based encryption from the Weil-pairing, D Boneh, M Franklin, Crypto'2001

25. How to 0wn the Internet in Your Spare Time, S Staniford, V Paxson, N Weaver, S Staniford, Proceedings of the 11th USENIX Security Symposium, 2002

26. Cryptanalysis of Diffie-Hellman, RSA, DSS, and Other Systems Using Timing Attacks, PC Kocher,

27. The Sybil Attack, JR Douceur, IPTPS, 2002

Software Engineering

Introduction
The take-home exam will involve specific additional reading and will also call for critical thinking in a more relaxed timeframe than afforded by an in-class exam.
Note: No single course covers all the topics in the listed papers. The way to interpret this reading list and, hence, the depth exam in Software Engineering is that you are expected through independent study to familiarize yourself with material beyond what you are exposed to in CSci 5801. The textbook by Hamlet and Maybee thus defines the point of departure. The additional papers indicate some of the important themes that you should have a grasp of beyond what is in that text.

Readings
Textbooks
1) The Engineering of Software. Dick Hamlet and Joe Maybee. Addison Wesley, 2001.
ISBN 0-201-70103-0

Process

1) Frederick P. Brooks. No Silver Bullet: Essence and Accidents in Software Engineering. *IEEE Computer.* 20(4): 10-19, April 1987.
2) David L. Parnas and Paul C. Clements. A Rational Design Process: How and Why to Fake It. *IEEE Transaction on Software Engineering.* 12(2): p. 251-257, February 1986.

Safety

1) Nancy G. Leveson. Software Safety: What, Why, and How? *ACM Computing Surveys.* 18(2): p. 125-163, July 1986.

Testing

1) W. Richard Adrian, Martha A. Branstad, and John C. Cherniavsky. Validation, Verification, and Testing of Computer Software. *ACM Computing Surveys.* 14(2): p. 159-192, June 1982.

Formal Specification

1) Kathryn L. Heninger. Specifying Software Requirements for Complex Systems: New techniques and Their Application. *IEEE Transaction on Software Engineering.* 6(1): p. 2-13, January 1980.
2) Jeanette M. Wing. A Specifier's Introduction to formal Methods. *IEEE Computer.* 23(9): p. 8-24, September 1990.
3) Anthony Hall. Seven Myths of Formal Methods. *IEEE Software*, September 1990.
4) N.G. Leveson, M.P.E. Heimdahl, H. Hildreth, and J.D. Reese. Requirements Specification for Process-Control Systems*. IEEE Transactions on Software Engineering*. 20(9) p. 684-706, September 1994.

Architecture

1) David Garlan, Robert Allel, and John Ockerbloom. Architectural Mismatch: Why Reuse is so Hard. *IEEE Software* 12(6): 17-26, November 1995.
2) Davis Garlan. Research directions in software architecture. *ACM Computing Surveys*. 27(2): p. 257-261, 1995.

Model Checking

1) Edmund M. Clarke, E.A. Emerson, A.P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*. 8(2): p. 244-263, April 1986.
2) Gerhard J. Holzmann. The model Checker SPIN. *IEEE Transaction on Software Engineering.* 23(5): p. 279-295, May 1997.
3) M.R. Huth and M.D. Ryan*. Logic in Computer Science: Modelling and reasoning about systems*, Chapter 3. Cambridge University Press, 1999.

Open Source and Agile Development

1) Kent Beck. Embracing Change with Extreme Programming. *IEEE Computer.* 32(10), p. 70-77, 1999.
2) E.S. Raymond. The Cathedral and the Bazaar. *First Monday.* 3(3), 1998.

Requirements Reference Models

1) Carl A. Gunter, Elsa L. Gunter, Michael Jackson, and Pamela Zave. A Reference Model for Requirements and Specifications. *IEEE Software*. 17(3): p. 37-43, May/June, 2000
2) J.M. Thompson, M.P.E. Heimdahl, and S.P. Miller. Specification-Based Prototyping for Embedded Systems. *Proceedings of the Seventh ACM/SIGSOFT Symposium on the Foundations of Software Engineering*. P. 163-179. Toulouse, France, September 1999.
3) D.L. Parnas and Jan Madey. Functional documents for computer systems*. Science of Computer Programming*. 25: p. 41-61, 1995.

Sparse Matrix Computations

1) M. J. D. Powell A. R. Curtis and J. K. Reid. On the estimation of sparse jacobian matrices. J. Inst. Maths. Applics., 13:117-119, 1974.

2) M. J. Balas. Trends in large space structure control theory: fondest dreams, wildest hopes. IEEE Trans. Aut. Contr., AC-2:522-535, 1982.

3) M. Borri and P. Mantegazza. Efficient solution of quadratic eigenproblems arising in dynamic analysis of structures. Comp. Meth. Appl. Mech. and Engng, 12:19-31, 1977.

4) P. N. Brown. A theoretical comparison of the Arnoldi and GMRES algorithms. SIAM Journal on Scientific and Statistical Computing, 12:58-78, 1991.

5) P. N. Brown and A. C. Hindmarsh. Matrix-free methods for stiff systems of ODEs. SIAM Journal on Numerical Analysis, 23:610-638, 1986.

6) C. Van Loan C. Moler. Nineteen dubious ways to compute the exponential of a matrix. SIAM Review, 20:8018369, 1978.

7) T. F. Chan and H. B. Keller. Arclength continuation and multi-grid techniques for nonlinear eigenvalue problems. SIAM Journal on Scientific and Statistical Computing, 3:173-194, 1982.

8) T.F. Chan and T. Mathew. The interface probing technique in domain decomposition. SIAM J. Matrix Anal. Applic., 13(1):212-238, 1992.

9) S. N. Chow, J. Mallet-Paret, and J. A. Yorke. Finding zeros of maps: homotopy methods that are constructive with probability one. Mathematics of Computations, 32:887-899, 1978.

10) P. Concus and G. H. Golub. A generalized conjugate gradient method for nonsymmetric systems of linear equations. In R. Glowinski and J. L. Lions, editors, Computing Methods in Applied Sciences and Engineering, pages 56-65, New York, 1976. Springer Verlag.

11) P. Concus, G. H. Golub, and G. Meurant. Block preconditioning for the conjugate gradient method. SIAM Journal on Scientific and Statistical Computing, 6:220-252, 1985.

12) J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. Math. Comput, 30:772-795, 1976.

13) R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. SIAM J. Numer. Anal., 18(2):400-408, 1982.

14) S. C. Eisenstat, H. C. Elman, and M. H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. SIAM Journal on Numerical Analysis, 20:345-357, 1983.

15) V. Faber and T. Manteu_el. Necessary and sufficient conditions for the existence of a conjugate gradient method. SIAM Journal on Numerical Analysis,21:352-361, 1984.

16) M. Hochbruck and C. Lubich. On krylov subspace approximations to the matrix exponential operator. SIAM Journal on Numerical Analysis, 34:1911-1925, 1997.

17) M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. SIAM Journal on Scientific Computing, 19:1552-1574, 1998.

18) B. N. Parlett J. R. Bunch. Direct methods for solving symmetric indefinite systems of linear equations. SIAM Journal on Numerical Analysis, 8:639-655, 1971.

19) L. Yu. Kolotilina and A. Yu. Yeremin. Factorized sparse approximate inverse preconditionings I.Theory. SIAM Journal on Matrix Analysis and Applications, 14:45-58, 1993.

20) J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coeficient matrix is a symmetric M-matrix. Mathematics of Computations, 31(137):148-162, 1977.

21) C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. SIAM Journal on Numerical Analysis, 12:617-624, 1975.

22) I. S. Du_ J. Reid. The multifrontal solution of unsymmetric sets of linear equations. SIAM Journal on Scientific and Statistical Computing, 5:633-641, 1984.

Special Matrix Problems and Advanced Matrix Computations

1) A. Bjorck, H. Park, and L. Elden. Accurate downdating of least squares solutions, SIAM Journal on Matrix Analysis and Applications, 15-2, pp. 549-568, 1994.

2) A. Bjorck and C. C. Paige. Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm, SIAM Journal on Matrix Analysis and Applications, 13, pp. 176-190, 1992.
3) M.T. Chu, R.E. Funderlic, and G. Golub. Rank modifications of semi-definite matrices with applications to secant updates. Technical report 96-10, SCCM program Stanford University.
http://www-sccm.stanford.edu/wrap/pub-tech.html

4) M.T. Chu, R.E. Funderlic, and G.H. Golub. A rank-reduction formula and its applications to matrix factorizations. SIAM Rev., 37: 512-530, 1995.

5) B. De Moor. Total least squares for affinely structured matrices and the noisy realization problem. IEEE Transactions on Signal Processing, SP--42:3104--3113, 1994.

6) G.H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM J. Numer. Anal.,10:413--432, 1973.

7) G. H. Golub and C. F. Van Loan. An analysis of the total least squares problem. SIAM Journal on Numerical Analysis, 17:883--893, 1980.

8) L. Hubert, J. Meulman, and W. Heiser. Two Purposes for Matrix Factorization: A Historical Appraisal. SIAM REVIEW, Vol. 42, No. 1, pp.68-82, 2000.

9) J.G. Nagy. Fast inverse QR factorization for Toeplitz matrices. SIAM J. Sci. Comput.,, 14, pp. 1174-1193, 1993.

10) H. Park and L. Eld'en. Schur-type methods for solving least squares problems with Toeplitz structure, SIAM Journal on Scientific Computing, 22-2, pp. 406-430, 2000.

11) C.C. Paige and M.A. Saunders. Towards a generalized singular value decomposition. SIAM J. Numer. Anal., 18:398-405, 1981.

12) J.B. Rosen, H. Park, and J. Glick. Structured total least norm for nonlinear problems, SIAM Journal on Matrix Analysis and Applications, 20-1, pp. 14-30, 1999.

13) G.W. Stewart. On the asymptotic behavior of scaled singular value and QR decompositions. Mathematics of Computations, 43:483-489, 1983.

14) G.W. Stewart. On the stability of sequential updates and downdates. IEEE Transactions on Signal Processing, 43, pp. 2642-2648, 1995.

15) S. Van Huffel, H. Park, and J.B. Rosen. Formulation and solution of structured Total Least Norm problems for parameter estimation, IEEE Transactions on Signal Processing, 44-10, pp. 2464-2474, 1996.

16) C.F. Van Loan. Generalizing the singular value decomposition. SIAM J. Numer. Anal., 13:76-83, 1976.

17) C.F. Van Loan. Computing the CS and the generalized singular value decomposition, Numer. Math., 46, pp. 479-492, 1985.

Visualization

Color

1) Healey C.G. (1996). Choosing Effective Colors for Data Visualization, IEEE Visualization 96, 263-270.

2) Rogowitz and Treinish (1996) "How NOT to Lie with Visualization", Computers in Physics10(3):268-273.

Texture

1) Rao A.R., Lohse G.L. Identifying High Level Features of Texture Perception, CVGIP: Graphical Models and Image Processing, Vol. 55(3), pp. 218-233, 1993.

2) Colin Ware and William Knight (1995) "Using Visual Texture for Information Display", ACM Transactions on Graphics, 14(1): 3-20.

Perceptual issues in VR

1) Kevin Arthur, Kellog Booth and Colin Ware (1993) Evaluating 3D Task Performance for Fish Tank Virtual Worlds, ACM Transactions on Information Systems, 11(3):239-265.

2) Usoh, M. et. al (1999). Walking > walking-in-place > flying, in virtual environments, SIGGRAPH 99, pp. 359-364.

Flow Visualization

1) Cabral and Leedom (1993) Imaging Vector Fields with Line Integral Convolution, SIGGRAPH

2) Stalling, D. and H.-C. Hege (1995) Fast and Resolution Independent Line Integral Convolution, SIGGRAPH 95.

Volumetric Modeling and Rendering

1) Levoy, M. (1988) Display of Surfaces in Volume Data, IEEE Computer Graphics and Applications

2) Lorensen, W. and Cline, H. (1987) Marching Cubes, SIGGRAPH 87.

3) Lee Westover. Footprint Evaluation for Volume Rendering. SIGGRAPH 1990.

4) P. Lacroute and M. Levoy, "Fast Volume Rendering Using a Shear-Warp factorization of the Viewing Transform", SIGGRAPH '94, pp. 451--457.

5) L. Sobierajski and A. Kaufman, "Volumetric Ray Tracing," ACM/IEEE Volume Visualization Symposium '94, pp. 11-18.

6) S. Wang and A. Kaufman, "Volume Sculpting," Symposium on Interactive 3D Graphics '95, pp. 151-156.

Hardware and Software Accelerated Volume Rendering

1) Rudiger Westermann and Thomas Ertl, "Efficiently Using Graphics Hardware in Volume Rendering Applications", SIGGRAPH '98, 169--178.

2) M. L. Brady, K. Jung, T. Nguyen and T. Nguyen: Interactive Volume Navigation. IEEE Transactions on Visualization and Computer Graphics 4(3): 243-256 (Jul-Sep 1998).

Visualization Systems

1) Craig Upson, Thomas Faulhaber, David Kamins, David Laidlaw, David Schlegel, Jeffrey Vroom, Robert Gurwitz, and Andries van Dam. The Application Visualization System: A Computational Environment for Scientific Visualization. IEEE Computer Graphics and Applications, 9(4), pp. 30-42, July 1989.

2) W.J. Schroeder, L.S. Avila, W. Hoffman. Visualizing with VTK: A Tutorial, IEEE Computer Graphics & Applications, 20(5), (Sept/Oct 2000).

5/20/2009

WPE Fall 2009 Application Form
Graduate Program in Computer Science

1. **Name:**_____ **Student #**_____ **Advisor:**_____

2. **Mailing Address:** _____
(Not dept.)
_____
_____

3. **Phone #:**_____ **E-Mail Address:** _____

4. Have you taken the WPE previously? YES_____ NO_____

If answer to 4 is "yes" then state when it was taken, the area(s) taken and the results.

5. From the chart on page 4, please indicate which two in-class exams you intend to take.

_____ _____

6. From the chart on page 4, please indicate which take-home exam you intend to take.

_____

7**. Before you can be considered as having passed the WPE, you must also complete the 6 breadth courses. These courses need not be completed before taking the Major exams, however, you will not be certified as passing the WPE until all components have been completed successfully.**

List courses (with grades) that you have taken at the University of Minnesota (or their equivalent) that fulfill the Breadth General Exam option. Also include courses you expect to take if known:

Course Number	Semester/Year	Course Name	Instructor	Grade	Area
					Theory
					Theory
					Systems
					Systems
					Applications
					Applications

8. Attach the following to this application:

   i) Most recent Graduate School Transcript (unofficial transcript is acceptable)

   ii) Photograph – digital is preferred

   Applications without these attachments will not be processed.

9. This form should be completed and submitted to Georganne in 4-192 EE/CSci as soon as possible but no later than:

   June 5, 2009

   Approval/rejection notices will be sent out shortly thereafter.

Be sure that you have answered all of the questions. Incomplete applications will not be processed.

Signature: _____

Date: _____

I have discussed the WPE with my student and have agreed that the exams listed are the most appropriate for his/her research interests.

Advisor Signature: _____

Date: _____