

## Ph.D. Comprehensive Exam

### INTRODUCTION

The following document describes the ph.d. comprehensive written examination format and syllabus. This format is in effect for all students registered in the SOCS ph.d. program from fall 2005 and onwards. The exam is intended to test Computer Science material which should be covered in any reasonable CS undergraduate program.

### SCHEDULING

The examination will be offered twice a year, once at the start of the fall term and once at the start of the winter term. Candidates must notify the graduate secretary of their intent to take the examination at least 6 weeks prior to the examination date.

### OVERALL STRUCTURE

There will be two parts to the examination: Part 1 and Part 2. In each part will consist of four papers. A student will have to pass two of the four papers in each part in order to pass the examination. The length of the examination will be three hours for each part, i.e., 1.5 hours for each paper.

Part 1 (50%) (a student must attempt 2 papers)

1. Algorithm Design and Data Structures
2. Theory of Computation
3. Programming Languages
4. Numerical Computing

Part 2 (50%) (student must attempt 2 papers)

5. Operating Systems
6. Databases
7. Software Engineering
8. AI

### PASSING THE EXAM

A student will have at most two opportunities to pass the exam. Should a student fail the examination once, they should attempt the examination as soon as it is offered again. Each paper will be marked out of 25 points, taking into account the difficulty of the questions, using the following scale as a general guideline:

- 23-25 - excellent, full grasp of the material
- 20-22 - very good,
- 17-19 - acceptable

15-16 - marginal  
13-14 - very weak  
< 13 - fail

---

DETAILED SYLLABI

---

Part 1 SYLLABUS

-----

Paper 1. Algorithm Design and Data Structures

Books:

"Introduction to Algorithms Second Edition",  
by T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein:  
MIT Press, 2001.

"Algorithm Design", by Jon Kleinberg and Eva Tardos,  
Addison-Wesley, 2006.

Topics:

Foundations: chapters 1 through 4 (Cormen):  
growth of functions, recurrences.

Sorting: chapters 6 through 9 (Cormen):  
heapsort, quicksort, lower bounds, radix sort, buck sort,  
medians and order statistics.

Data structures: chapters 10 through 14 (Cormen):  
elementary data structures, heaps, binary search trees,  
red-black trees, augmentation.

Advanced data structures: chapter 17 (Cormen)  
on amortized analysis, and chapter 21 (Cormen) on disjoint  
set structures.

Analysis techniques: chapters 15 and 16 (Cormen):  
dynamic programming, greedy algorithms.

Graph algorithms, chapters 22 through 26 (Cormen):  
elementary algorithms, minimum spanning tree, shortest  
path.

Maximum flow---chapter 7 of (Kleinberg)

---

Paper 2. Theory of Computation.

Book:

"Introduction to the Theory of Computation"  
by Michael Sipser,  
Thompson, 2005.

Chapters 0-1-2-3-4-5-7-8

Topics:

automata, nondeterminism, languages, regular expressions,  
context-free languages, grammars, the Church-Turing thesis,  
decidability, the halting problem, reducibility, the recursion  
theorem, P, NP, NP completeness, polynomial time  
reducibility, the Cook-Levin theorem, PSPACE, Savitch's theorem,  
and the classes L and NL.

---

Paper 3. Programming Languages

Book:

- Types and Programming Languages, Benjamin Pierce, 2001, MIT Press  
(Chapter 1,2, 3, 5, 8, 9, 11, 13, 18.1-18.8, 22)

Topics:

- Defining a language  
(Context-free grammars, inductive definitions,  
abstract syntax, bindings, principles of inductive proofs)  
(CH 2 + 3 Pierce)
- Formal syntax and Static and operational semantics  
(Ch 3, CH 5 Ch 8, Ch 9, Ch 11 Pierce)

- Imperative features (references)  
(CH 13 Pierce)
- Foundations of object-oriented languages  
(Ch 18.1-18.8 Pierce)
- Polymorphism  
(CH 22 Pierce)
- General theory of subtyping  
(CH 15 Pierce)
- Type systems for OO languages  
including inheritance and subtyping in Java.  
(CH 18 Pierce)

---

Paper 4. Numerical Computing

Book:

Cheney and Kincaid, Numerical Mathematics and Computing (5th ed):  
 Chap 2, Chap 3, Chap 4, Chap 5, Chap 6, Chap 7, Chap 8 (8.3 to 8.4),  
 Chap 9 (9.0 to 9.2), Chap 10 (10.0 to 10.2), Chap 12 (12.1).

Topics:

Floating-point arithmetic and rounding errors;  
 Solving systems of linear equations;  
 Eigenvalue problems;  
 Solving a nonlinear equation;  
 Polynomial and spline interpolation;  
 Least squares approximation;  
 Numerical integration;  
 Numerical solution of ordinary differential equations.

---

Part 2 SYLLABUS

---

Paper 5. Operating Systems

Book:

Operating Systems Concepts, A. Silberschatz, G. Gagne,  
and P. Galvin, Seventh Edition, John Wiley, 2005.

Topics:

(covering Chapters 5, 6, 7, 8, 9, 14, 15, 17, 18)

CPU scheduling, scheduling algorithms, multi-processor  
scheduling, algorithm evaluation

Process synchronization, critical-section, synchronization,  
semaphores, monitors, atomic transactions

Deadlocks, characterization, prevention, avoidance, detection,  
recovery from deadlock

Memory management, allocation, paging, segmentation, demand  
paging, page replacement, memory-mapped files

Protection and security, protection goals, principles of  
protection, access matrix, access, control, program threats

Distributed File systems, naming and transparency, stateful  
versus stateless service, file replication

Distributed coordination, event ordering, mutual exclusion,  
atomicity, concurrency control, deadlock handling, election  
algorithms, reaching agreement

---

Paper 6. Databases

Book:

Database Management Systems, R. Ramakrishnan and J. Gehrke,  
Third Edition, McGraw-Hill, 2003.

Topics:

(covering Chapters 2.1-2.5, 3, 4, 5.1-5.7, 12, 16, 17, 18, 19)

Database design, relational model, relational algebra and

calculus, scheme refinement and normal forms

Basic queries, nested queries, aggregate queries, integrity constraints in query languages

Query evaluation, operator evaluation, query optimization

Transactions, ACID properties, concurrent transactions, locking, serializability, recoverability, concurrency control without locking, crash recovery and logs

---

## Paper 7. Software Engineering

### Resources:

- Horstmann, Object-Oriented Design and Patterns, 2nd Edition
- Fowler, UML Distilled, 3rd Edition
- Gamma et al, Design Patterns: Elements of Reusable Object-oriented Software

### Topics:

- Encapsulation and contracts (Horstmann chapter 3)
- Design with interfaces and polymorphism (Horstmann chapter 4)
- Modeling static structure (Fowler chapters 3 and 5)
- Implementation reuse and composition (Horstmann chapter 6)
- Design patterns (Gamma et al. chapter 1 and the following patterns: Factory Method, Observer, Strategy, Composite, Decorator, Command, Visitor)
- Modeling software behavior (Fowler chapter 4)
- Modeling software state (Fowler chapter 10)
- Basic mechanisms: reflection, cloning, serialization (Horstmann chapter 7)
- Basic concepts of libraries and components (Horstmann chapter 7)

---

Paper 8. Artificial Intelligence

Book:

"Artificial Intelligence: A Modern Approach"  
by Stuart Russell and Peter Norvig (Second Edition)  
2002, Prentice-Hall

- Search algorithms: uninformed search, A\*, heuristics, local search  
Chapter 3, Sections 4.1-4.3
  - Constraint satisfaction problems  
Chapter 5
  - Adversarial search; minimax, alpha-beta pruning  
Chapter 6
  - Propositional logic, first-order logic and inference  
Chapter 7, Sections 8.1-8.3, Chapter 9
  - Classical planning  
Chapter 1, Sections 12.1-12.5
  - Probabilities and graphical models  
Chapter 13, Sections 14.1-14.5, 15.1-15.3
  - Utility theory and sequential decision making  
Sections 16.1-16.6, 17.1-17.5
  - Inductive learning, statistical methods, reinforcement learning  
Sections 18.1-18.3, 20.1-20.5, Chapter 21
- 
-