Syllabus for Part II of the
Ph.D. Qualifying Exam in Computer Science
Revised (9/17/13)

Part II of the Ph.D. qualifying exam is also for four hours. Students will be tested on the two specialty areas they chose when they registered for the exam. Each area will have four questions; a student must answer any three of them. The exam is closed-book and closed-notes. The use of electronic devices is not allowed.

This document describes the topics from which the questions for each specialty area will be chosen. Each area is accompanied by (i) suggested texts and (ii) Computer Science courses that cover the area in sufficient depth.
Artificial Intelligence

Topics: state space search, heuristic (informed) search, adversarial search, knowledge representation, propositional logic, first-order logic, deductive inference, resolution, planning, natural language, phrase structure grammars, network grammars, chart parsing, and programming in Lisp.

Suggested Texts:

Courses: CompSci 422: Introduction to Artificial Intelligence or CompSci 710: Artificial Intelligence, and CS 743: Intelligent User Interfaces
Computer Architecture and Performance Evaluation

**Computer Architecture**
Topics: central processing unit design, instruction set design, control unit design, arithmetic and logical (ALU) design, pipelining and vector processing, input/output organization, memory organization, parallel computer architecture, multi-core computer architecture.

Suggested Texts:

Course: CompSci 458: Computer Architecture

**Computer Systems Performance Evaluation**
Topics: discrete and continuous random variables, expectation, conditional expectation, stochastic processes, discrete and continuous time Markov chains, networks of queues.


Course: CompSci 760: Computer Systems Performance Evaluation
Computer Graphics and Visualization

Topics: graphics primitives and attributes, line-drawing algorithms, 2D and 3D geometric transformations, 2D and 3D viewing, 3D object representations (Hermite splines, Bezier splines, B-splines), visibility detection, illumination models and polygon rendering, texture mapping, color models, ray-tracing, and basic computer animation.


Networks

The Computer Networks portion of the qualifying exam covers the fundamental concepts in wired and wireless networking, hardware technology, and software protocols.

Topics: physical layer, data communications, wired and wireless transmission media, data link layer, wired and wireless local area networks, wide area networks, network layer, routing protocols, mobile IP, transport layer, TCP/IP protocol suite, bluetooth, ad hoc and wireless sensor networks, multicast routing, congestion control and traffic management, and quality of service (QOS).


Courses: CompSci 520: Computer Networks and CompSci 730: Advanced Computer Networks
Computer Security

Topics: components of computer security and classification of security threats, symmetric key and public key cryptosystems, cryptographic hash functions, digital signatures, key managements, network security protocols, entity authentication, system security and computer viruses.

Suggested Texts:

Courses: CompSci 469: Intro to Computer Security and CompSci 759: Data Security
Human Computer Interaction and User Interfaces

Natural Language Processing
Topics: words, syntax: grammars and parsing (syntactic and statistical), semantics, semantic analysis, and word sense discrimination, discourse, applications (dialog, information extraction, factoid question answering).

Suggested Texts:

Course: CompSci 423: Introduction to Natural Language Processing or CompSci 723: Natural Language Processing

Human-Computer Interaction
Topics: requirements gathering and specification, design concepts and components, designing for GUIs and the web, evaluation planning and analysis, advocating usability.


Course: CompSci 747: Human-Computer Interaction
Programming Languages and Compilers

The programming languages and compilers portion of the qualifying examination covers the topics of programming languages concepts, compilers and type theory.

Programming Language Concepts

Topics: Syntax, BNF grammar, language systems, types, scopes, memory location for variables, polymorphism, memory management, parameter passing, formal semantics, functional programming (ML language, pattern matching, higher order functions, data types), object-oriented programming (classes, abstract classes and interfaces, iterators, subtype polymorphism), logic programming (Prolog language, unification, resolution, optimization).


Course: CompSci 431: Programming Languages Concepts

Compilers

Topics: compiler phases (scanning, parsing, semantic analysis, machine-independent optimization, machine dependent optimization, code generation), regular expressions, parsing tables, symbol tables, type rules, attribute grammars, pipeline issues, flow analysis, separate compilation, linking.


Course: CompSci 654: Introduction to Compilers, or CompSci 754: Compiler Construction

Type Systems

Topics: operational semantics, type rules, lambda calculus, proofs (progress, preservation, inversion, canonical forms), type concepts (products, sums, records, mutable state, subtyping, recursive types, universals, existentials, bounded quantification, higher-order polymorphism).


Course: CompSci 732: Type Systems for Programming Languages
Theory

The theory portion of the exams covers the topics of automata and formal languages, and algorithm design and analysis.

Automata and Formal Languages

Topics: finite automata, pushdown automata, Turing machines and variants, regular languages, context-free languages, recursive languages, recursively enumerable languages, regular expressions, various classes of grammars and normal forms, reducibility, decidability.

Suggested Texts:

Course: CompSci 417: Introduction to the Theory of Computation

Algorithm Design and Analysis

Topics: asymptotic notation, solving recurrence relations, stacks, queues, vectors, lists, trees, priority queues and heaps, hashing, binary search trees (including AVL trees and red black trees), sorting and selection, structures/algorithms for disjoint sets, algorithm design techniques (the greedy method, divide and conquer, recursion, dynamic programming), graph algorithms (including minimum spanning tree algorithms and shortest path algorithms).

Suggested Texts:

Course: CompSci 535: Algorithm Design and Analysis

Analysis of Algorithms

Topics: selection, amortized analysis, algorithm design techniques (the greedy method, divide and conquer, recursion, dynamic programming), graph algorithms (including minimum spanning tree algorithms, shortest path algorithms and network flows), randomized algorithms (including quicksort, the closest pair of points problem, the hiring problem, the Rabin-Karp string matching algorithm, primality testing), NP-completeness, approximation algorithms (including the vertex cover problem, the metric TSP, set covering).

Suggested Texts:

Course: CompSci 704: Analysis of Algorithms