Study Guide for Written Comprehensive Exams

Spring 2022

A PHOTO ID IS REQUIRED FOR ENTRANCE TO THE EXAM.

YOU WILL ONLY BE ALLOWED TO TAKE WRITING INSTRUMENTS TO YOUR SEAT DURING THE EXAM. YOU WILL BE REQUIRED TO LEAVE ALL BACKPACKS, NOTES, AND ELECTRONIC DEVICES AT THE BACK OR FRONT OF THE EXAM ROOM.

The written exams cover the core courses.
So be sure to check that you have the right set of topics.

Software Engineering Study Guide
-------------------------------

CS 5391
- software life cycle models
- software measurement

CS 5392
- data flow analysis
- symbolic execution
- propositional logic and predicate logic
- model checking
- satisfiability based verification
- verification by proof

CS 5393
- software inspections
- code coverage measures for testing
- coverage criteria for testing
- syntax based testing
- regression testing

Computer Science Study Guide
-------------------------------

1. CS5306 Advanced Operating Systems
   a. mutual exclusion solutions in distributed os
   b. deadlock solutions in distributed os
   c. clock synchronization in distributed os (physical clock, logical clock)
   d. consistency models of distributed shared memory

2. CS5310 Network and Communication Systems
a. Significances of the Fourier series with regard to data transmissions
b. The seven layers of the OSI reference model
c. Definitions and comparisons of CSMA/CD, Token Bus, and Token Ring LAN protocols
d. Basic concepts, structures, and programming of BSD Socket programming

3. CS5318 Design of Programming Languages
   a. computational paradigms (imperative, object-oriented, functional, etc.)
   b. abstractions in programming languages
c. programming language concepts (types, parameter passing, lambda calculus, etc.)
d. regular, context-free, and attribute grammars
e. parsing, parsing trees, and abstract syntax trees
f. programming language implementation issues

4. CS5329 Algorithm Design and Analysis
   a. The divide conquer algorithm. Understand the concept and the Recurrences including recursion trees, substitution and master method, with examples.
   b. Algorithm design strategies - including characteristics, analysis, and examples.
c. Insertion Sort, Quick Sort, Heap Sort- Know the definitions, properties and their running time for the best and worst cases in terms of big O notation. Be able to analyze and to prove some of the theorems, complexity, and be able to apply to specific inputs.
d. Hash functions - definitions, collision resolution schemes, linear, quadratic probing, double hashing. Be able to provide and work with specific examples
e. Dynamic Programming and Greedy Algorithm. Understand the elements of the methodologies and the application to examples. Understand their computation time complexity.
f. Tree and elementary graph theory understand some tree structures including, but not limited to Binary Search Tree, Heap, and BFS for Graph theory, with Definitions, properties, And examples. Be able to add and delete nodes from each of those entities.

5. CS5332 Data Base Theory and Design
   a. elementary concepts of DBMS
   b. data analysis for table formation
c. normalization of tables(relations): 1NF, 2NF, 3NF, and BCNF
d. E/R modeling
e. SQL, database tools and database implementation for real world problems
f. relational algebra

6. CS5338 Formal Languages
   a. Mathematical foundations including countable and uncountable infinity and diagonalization
   b. Definition, construction and properties of finite state machines, pushdown automata and Turing machines
c. Halting behavior of deterministic and non-deterministic machines
d. Language hierarchy and closure properties of regular and context free languages

e. Complexity classes and NP-completeness proof techniques

f. Church-Turing thesis, decidability and semi-decidability

7. CS5346 Advanced Artificial Intelligence
   a. Conventional and Informed Search Methods
   b. Minimax Search Procedures including Alpha-Beta Pruning
   c. Representation of Knowledge and Inference in Propositional and First-Order Logic
   d. Use of Forward and Backward Chaining in Expert Systems
   e. Building knowledge bases and reasoning for applications like the testing of the design of electronic circuits
   f. Rule generation through machine learning methods
   g. Intelligent planning

8. CS 5351 Parallel Processing
   a. Performance metrics (speedup, efficiency, overhead, scalability, complexity)
   b. Parallelization approaches (data parallelism, task parallelism, divide and conquer, partitioning, agglomeration, scans)
   c. Shared-memory machines and programming (consistency models, cache coherence, false sharing)
   d. Message-passing machines and programming (network topologies, bandwidth, routing)
   e. Synchronization primitives (barriers, mutexes, locks, read/write locks, semaphores, condition variables)
   f. Communication primitives (send, receive, broadcast, reduce, scatter, gather, point to point vs. collective, blocking vs. non-blocking)
   g. Parallelism issues (data races, livelock, deadlock, indeterminacy, load balance, termination detection, reentrancy)
   h. Parallel programming (OpenMP, MPI)

9. CS5391 Survey of Software Engineering
   a. Software Development Lifecycle Models
   b. Capability Maturity Model
   c. Design Methods
   d. Software Testing Strategies