**Computer Science Comprehensive Exam syllabus**

**I. SOFTWARE SYSTEMS AND METHODOLOGY — 40%**

A. Data organization

* Data types
* Data structures and implementation techniques

B. Program control and structure

* Iteration and recursion
* Procedures, functions, methods and exception handlers
* Concurrency, communication and synchronization

C. Programming languages and notation

* Constructs for data organization and program control
* Scope, binding and parameter passing
* Expression evaluation

D. Software engineering

* Formal specifications and assertions
* Verification techniques
* Software development models, patterns and tools

E. Systems

* Compilers, interpreters and run-time systems
* Operating systems, including resource management and protection/security
* Networking, Internet and distributed systems
* Databases
* System analysis and development tools

**II. COMPUTER ORGANIZATION AND ARCHITECTURE — 15%**

A. Digital logic design

* Implementation of combinational and sequential circuits
* Optimization and analysis

B. Processors and control units

* Instruction sets
* Computer arithmetic and number representation
* Register and ALU organization
* Data paths and control sequencing

C. Memories and their hierarchies

* Performance, implementation and management
* Cache, main and secondary storage
* Virtual memory, paging and segmentation

D. Networking and communications

* Interconnect structures (e.g., buses, switches, routers)
* I/O systems and protocols
* Synchronization

E. High-performance architectures

* Pipelining superscalar and out-of-order execution processors
* Parallel and distributed architectures

**III. THEORY AND MATHEMATICAL BACKGROUND — 40%**

A. Algorithms and complexity

* Exact and asymptotic analysis of specific algorithms
* Algorithmic design techniques (e.g., greedy, dynamic programming, divide and conquer)
* Upper and lower bounds on the complexity of specific problems
* Computational complexity, including NP-completeness

B. Automata and language theory

* Models of computation (finite automata, Turing machines)
* Formal languages and grammars (regular and context free)
* Decidability

C. Discrete structures

* Mathematical logic
* Elementary combinatorics and graph theory
* Discrete probability, recurrence relations and number theory

**IV. OTHER TOPICS — 5%**

Example areas include numerical analysis, artificial intelligence, computer graphics, cryptography, security and social issues.