Syllabus for Part I of the Software Engineering Ph.D. Comprehensive Examination
Department of Computing and Software
McMaster University

CAS-2012-10

Area 1 - Computing Fundamentals

1. Fundamental Data Structures
   Arrays, lists, queues, stacks, trees, priority queues, balanced trees, sets.

2. Discrete mathematics
   Sets, functions, relations; partial and linear orders, lattices, boolean algebras; basic algebraic structures like monoids, rings; graphs and trees; sequences and series.

3. Combinatorics
   Basic counting principles; permutations and combinations; basic probability

4. Logic
   Syntax vs. semantics; languages, theories, models; propositional logic; first-order logic; higher-order logic; formal proof systems; inductive proofs; pre-post conditions; weakest pre-condition; program verification

5. Algorithms
   Sorting and searching algorithms; algorithm design schemes such as greedy algorithms, dynamic programming; graph and network algorithms; linear programming; recursion; algorithm complexity

6. Theory of computation
   Regular, context-free, context-sensitive, and recursively enumerable languages; finite automata; Church-Turing thesis and common models of computation such as recursive functions, Turing machines; complexity classes

7. Information security
   Confidentiality, authentication and integrity; defence mechanisms; cryptography; network security; secure communication protocols; security management

Area 2 - Software Engineering

1. Formal specification techniques and model checking
   Abstract data types; algebraic specifications; model checking

2. Requirements
   Elicitation; verification and validation techniques; requirements documentation

3. Design principles
   Separation of concerns; low coupling and hight cohesion; open-closed principle; economy of mechanisms;

4. Architecture design
   Architectural styles; models for software architecture; Object Oriented Architecture; data flow architecture; data centered architecture; hierarchical architecture; implicit asynchronous communication architecture; interaction oriented architecture; distributed architecture;component based architecture; heterogeneous architecture

5. Detailed design and issues in embedded and realtime system design
   Module interface design and module internal design; resource sharing; multi-threaded processes; creational design patterns (e.g. factory, singleton, abstract factory, prototype); structural design patterns (e.g. Facade, decorator, adapter, proxy); behavioural design patterns (e.g. Interpreter, iterator, mediator, observer); design issues related to the design of real-time and embedded systems (jitter, latency, hard vs. soft real-time, pre and runtime scheduling); PID controller Design and fundamental control concepts

6. Testing
   Unit testing techniques; integration testing; statistical testing; regression testing; risk assessment; test strategies; test plans.

7. Software qualities
   Software metrics and models for assessing software quality; Software quality attributes