INTRODUCTION

Nineteen engineering disciplines are included in the Examination Syllabus issued by the Canadian Engineering Qualifications Board of Engineers Canada.

Each discipline examination syllabus is divided into two examination categories: compulsory and elective. A full set of Software Engineering examinations consists of ten, three-hour examination papers. Candidates will be assigned examinations based on an assessment of their academic background. Examinations from discipline syllabi other than those specific to the candidates’ discipline may be assigned at the discretion of the constituent Association/Ordre.

Before writing the discipline examinations, candidates must have passed, or have been exempted from, the Basic Studies Examinations.

The constituent Association/Ordre will supply information on examination scheduling, textbooks, materials provided or required, and whether the examinations are open or closed book.

SOFTWARE ENGINEERING EXAMINATIONS

GROUP A

COMPULSORY EXAMINATIONS (SEVEN REQUIRED)

04-Soft-A1 Algorithms & Data Structures

Fundamental data structures and their associated algorithms. Stacks and queues, trees, tables, lists, arrays, strings, sets; files and access methods. B-trees, multi-key organizations. Searching. Sorting. Algorithm design techniques, such as divide and conquer, the greedy method, balancing, dynamic programming. Algorithms related to set operations, Graphs, graph algorithms: depth-first and breadth-first search, minimum spanning tree, shortest path. Empirical and theoretical measures of the efficiency of algorithms. Complexity analysis. Hard problems, NP-completeness, and intractable problems.

04-Soft-A2 Computing Structures

Computer Architecture basics, including Boolean algebra, gates, combinational and sequential logic, machine-level representation of data; machine organization, assembly/machine language programming; memory organization, caches, heaps, stacks; serial and parallel I/O, interrupts, bus protocols, and direct-memory access (DMA). Operating System basics, including concurrency, process scheduling, memory management; protection, access, and authentication; linking and loading. Database basics, including database architecture, query languages, transactions. File system organization and access methods (sequential, indexed-sequential, extendible hashing, B-trees), index organization.
04-Soft-A3  Software Design


04-Soft-A4  Real Time Systems


04-Soft-A5  Requirements and Specifications

Elicitation sources and techniques. Modelling paradigms, including information modelling, behavioural modelling, domain modelling, functional modelling, constraint modelling. Quality requirements (e.g., performance, usability, reliability, maintainability); expressing quality requirements so that they are testable. Prioritization, trade-off analysis, negotiation, risk analysis, and impact analysis. Requirements management, consistency management, interaction analysis, traceability. Requirements documentation and specification languages. Validation, reviews and inspections, prototyping, validating non-functional requirements. Acceptance test design.

04-Soft-A6  Software Quality Assurance

Validation and Verification concepts, Software Lifecycle and application of validation and verification, Software Quality Assurance processes, Definitions of software product quality, Quality Characteristics, Engineering quality definitions, specifications, Definition and classifications of software Defects, Fitness for use and customer quality definitions, Software Costs, quality costs and economics, Reviews, Walkthroughs and Inspections: General Concepts, Unit (Module / Package) level testing, Subsystem / Integration testing, Regression testing, State based testing, Traditional Functional Testing, Logical Testing/Analysis, OO Testing considerations (polymorphism and inheritance), Safety / Failure Analysis and testing.

04-Soft-A7  Software Process

GROUP B

ELECTIVE EXAMINATIONS (THREE REQUIRED)

04-Soft-B1  Advanced Object Oriented Design


04-Soft-B2  User interface


04-Soft-B3  Security/Safety


04-Soft-B4  Reliability and Fault Tolerance

04-Soft-B5  Software Modeling & Verification (Formal Methods)

Mathematical modelling of software, including logic, extended finite state machines, process algebra, functions, and algebraic specifications. Mathematical reasoning of such models, including proofs of consistency, completeness, and correctness. Tools for type checking, well-formedness checking, simulation, invariant and property checking (e.g., deadlock checking, model checking), test-case generation, and code generation.

04-Soft-B6  Advanced Software Project Management, Life Cycle Methodologies

Software Project Management Processes
Planning, integration and change control; scope management; quality management; cost management; risk management; schedule management; communications management; human resource management; and procurement management.

Lifecycle Methodologies
Development models (Waterfall, V-Model, Incremental, Spiral, etc.) and techniques (rapid prototype, clean room, Object Oriented, etc.) Military/aerospace and commercial development standards. Phase specific activities: software specification and requirements analysis; software architecture comparison and selection; software design; coding and unit testing; integration testing; system testing; and operational turn-over. Lifecycle activities: baseline management, software quality management, software configuration management.

Project involving forensic project management of a real industrial project.

04-Soft-B7  Reverse Engineering, Maintenance & Evolution

Software maintenance: corrective, perfective, and adaptive. Techniques for reverse engineering software architecture and design, for the purpose of program comprehension. System and process re-engineering (technical and business). Refactoring. Migration (technical and business). Impact analysis. Release and configuration management. Models of software evolution (theories, laws). Relationship among evolving entities (e.g., assumptions, requirements, architecture, design, code, test suites). Legacy systems.

04-Soft-B8  Distributed Systems

04-Soft-B9  Parallel Computing


04-Soft-B10  Networking and Communications


04-Soft-B11  Process Control Systems


04-Soft-B12  Scientific Computation


04-Soft-B13  Performance Analysis & Simulation

Basic techniques of system performance evaluation. Specific topics include: performance modeling, discrete event simulation, verification and validation of simulation models, analysis of simulation output, analysis of single server queue and queuing networks, modeling of computer systems, networks, and other queuing or non-queuing systems.
04-Soft-B14  Safety Critical Systems


04-Soft-B15  Artificial Intelligence/Intelligent Systems


04-Soft-B16  Compilers


04-Soft-B17  Programming Language Paradigm

Principles of programming methodologies and the evolution of programming-language features (e.g., modules, objects, inheritance, polymorphism, exceptions, templates) to support those methodologies. Examination of major programming-language paradigms, including procedural, logic, functional, and object-oriented. Programming knowledge in a variety of languages.

04-Soft-B18  Computer Graphics/Imaging/Visualization
