**ENGINEERS AND GEOSCIENTISTS BRITISH COLUMBIA**

2019 SOFTWARE ENGINEERING SYLLABUS

For Self-Evaluation

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ User ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***For directions, refer to the*** [***Instructions for Completing Syllabus and Course Descriptions***](mailto:https://www.egbc.ca/getmedia/8fbcf379-28d9-4639-bafd-bb3df83f225d/APEGBC-Guide-to-Completing-Syllabus-and-Course-Description-1.pdf.aspx)***. Please save as a PDF document and upload via your applicant portal.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exam Number** | **Exam Name** | **Applicant’s Self-Evaluation – Course Equivalent Code** | **Page Number Reference** | **For Office Use Only** |
| **Basic Studies (4 Required)** | | | | |
| 04-BS-2 | Probability and Statistics |  |  | Full Credit No Credit  Comments |
| 04-BS-5 | Advanced Mathematics |  |  | Full Credit No Credit  Comments |
| 04-BS-8 | Digital Logic Circuits |  |  | Full Credit No Credit  Comments |
| 04-BS-16 | Discrete Mathematics |  |  | Full Credit No Credit  Comments |
| **Basic Studies (2 Required)** | | | | |
| 04-BS-1 | Mathematics |  |  | Full Credit No Credit  Comments |
| 04-BS-4 | Electric Circuits and Power |  |  | Full Credit No Credit  Comments |
| 04-BS-6 | Mechanics of Materials |  |  | Full Credit No Credit  Comments |
| 04-BS-7 | Mechanics of Fluid |  |  | Full Credit No Credit  Comments |
| 04-BS-10 | Thermodynamics |  |  | Full Credit No Credit  Comments |
| 04-BS-11 | Properties of Materials |  |  | Full Credit No Credit  Comments |
| 04-BS-12 | Organic Chemistry |  |  | Full Credit No Credit  Comments |
| 04-BS-13 | Biology |  |  | Full Credit No Credit  Comments |
| 04-BS-14 | Geology |  |  | Full Credit No Credit  Comments |
| **Group A (7 required)** | | | | |
| 19-Soft-A1 | Algorithms & Data Structures |  |  | Full Credit No Credit  Comments |
| 19-Soft-A2 | Computer Architecture and Operating Systems |  |  | Full Credit No Credit  Comments |
| 19-Soft-A3 | Software Design |  |  | Full Credit No Credit  Comments |
| 19-Soft-A4 | Real-Time Systems |  |  | Full Credit No Credit  Comments |
| 19-Soft-A5 | Requirements and Specifications |  |  | Full Credit No Credit  Comments |
| 19-Soft-A6 | Software Quality Assurance |  |  | Full Credit No Credit  Comments |
| 19-Soft-A7 | Software Development Process |  |  | Full Credit No Credit  Comments |
| **Group B (3 Required)** | | | | |
| 19-Soft-B1 | Advanced Software Design |  |  | Full Credit No Credit  Comments |
| 19-Soft-B2 | User Interface |  |  | Full Credit No Credit  Comments |
| 19-Soft-B3 | Security |  |  | Full Credit No Credit  Comments |
| 19-Soft-B4 | Dependable Systems |  |  | Full Credit No Credit  Comments |
| 19-Soft-B5 | Software Modeling & Verification (Formal Methods) |  |  | Full Credit No Credit  Comments |
| 19-Soft-B6 | Software Project Management |  |  | Full Credit No Credit  Comments |
| 19-Soft-B7 | Reverse Engineering, Maintenance & Evolution |  |  | Full Credit No Credit  Comments |
| 19-Soft-B8 | Distributed Systems |  |  | Full Credit No Credit  Comments |
| 19-Soft-B9 | Parallel Computing |  |  | Full Credit No Credit  Comments |
| 19-Soft-B10 | Networking and Communications |  |  | Full Credit No Credit  Comments |
| 19-Soft-B11 | Process Control Systems |  |  | Full Credit No Credit  Comments |
| 19-Soft-B12 | Engineering Computation: Numerics |  |  | Full Credit No Credit  Comments |
| 19-Soft-B13 | Performance Analysis & Simulation |  |  | Full Credit No Credit  Comments |
| 19-Soft-B14 | Safety Critical Systems |  |  | Full Credit No Credit  Comments |
| 19-Soft-B15 | Artificial Intelligence |  |  | Full Credit No Credit  Comments |
| 19-Soft-B17 | Programming Languages, Semantics, and Implementation |  |  | Full Credit No Credit  Comments |
| 19-Soft-B18 | Data Visualization |  |  | Full Credit No Credit  Comments |
| **Complementary Studies (All Required)** | | | | |
| 11-CS-1 | Engineering Economics |  |  | Full Credit No Credit  Comments |
| 11-CS-2 | Engineering in Society – Health and Safety |  |  | Full Credit No Credit  Comments |
| 11-CS-3 | Sustainability, Engineering and the Environment |  |  | Full Credit No Credit  Comments |
| 11-CS-4 | Engineering Management |  |  | Full Credit No Credit  Comments |

**INTRODUCTION**

The Canadian Engineering Qualifications Board of Engineers Canada issues the Examination Syllabus that includes a continually increasing number of engineering disciplines.

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.

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

Information on examination scheduling, textbooks, materials provided or required, and whether the examinations are open or closed book, will be supplied by the constituent association.

**SOFTWARE ENGINEERING EXAMINATIONS**

**GROUP A**

**COMPULSORY EXAMINATIONS (SEVEN REQUIRED)**

**19-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.

**19-Soft-A2 Computer Architecture and Operating Systems**

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; virtual machines.

**19-Soft-A3 Software Design**

Role of software design activity. Software design quality attributes: correctness, reliability, maintainability, portability, robustness. Software design principles: separation of concerns, abstraction, information hiding.

Static and dynamic typing. Mutable and immutable types. Modularity and decomposition. Function-oriented design. Object-oriented design. Subtyping. Components. Interface design. Module level design.

Notations: UML and other notations. Basic concepts of design patterns. Introduction to testing: unit tests,

blackbox vs. grey box testing, test coverage.

**19-Soft-A4 Real-Time Systems**

Definition and characteristics of real-time systems. Hard and soft real-time systems. Dynamic responses of simple physical processes. Designing real-time systems (requirements, design methods, implementation, testing, human-computer interaction). Reliability and fault tolerance. Exceptions and exception handling. Concurrency, synchronization, communication and resource control. Scheduling (cyclic executive, rate monotonic and deadline priority, priority ceiling protocols). Real-time operating systems. Simple embedded systems.

**19-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 (e.g., use cases) and specification languages. Validation, reviews and inspections, prototyping, validating non-functional requirements. Acceptance test design.

**19-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. 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.

**19-Soft-A7 Software Development Process**

Software life cycles. Software process models. Control and life-cycle management of correct, reliable, maintainable and cost-effective software. Software documentation. Project management tools. Risk management. Communication and collaboration. Cause and effects of project failure. Cost estimation and scheduling. Factors influencing productivity and success. Productivity metrics. Configuration management. Defect management.

**GROUP B**

**ELECTIVE EXAMINATIONS (THREE REQUIRED)**

**19-Soft-B1 Advanced Software Design**

Software design paradigms: object-oriented, service-oriented, component-based, agent-based, functional programming, client-server (including protocols such as REST), virtualization. Distributed component-based frameworks and systems. Design patterns. Model-driven design of software. Software architecture.

Architecture representation.

**19-Soft-B2 User interface**

Psychological principles of human-computer interaction. Evaluation of user interfaces. Usability engineering. Task analysis, user-centered design and prototyping. Conceptual models and metaphors. Software design rationale. Design of windows, menus and commands. Voice and natural language I/O. Response time and feedback. Colour, icons and sound. Internationalization and localization. User interface architectures and APIs. Case studies and project.

**19-Soft-B3 Security**

Security risks, threats, and vulnerabilities. Confidentiality, integrity, and privacy. Cryptography, access control, assurance, accountability. Engineering of secure systems, architectural approaches (e.g., confinement, virtual machines, trusted computing). Analysis techniques (e.g., static analysis and testing, model checking). Implications on human interface design and usability.

**19-Soft-B4 Dependable systems**

Basic geomorphological concepts: formation and composition of landforms, geomorphologic cycles. Weathering and soils. Mass wasting. Fluvial processes and landforms. Coastal processes and landforms. Glacial geomorphology and landforms. Frozen-ground phenomena. Karst geomorphology. Physical geology of Canada. Quaternary geology of selected areas of Canada. Influence of geomorphology on human activity.

**18-Geol-B5 Environmental Geology**

Software and hardware faults. Faults, latent faults and failures. Characterization of failure functions, probability distribution of failures, failure intensity function. Software reliability definition and measures. MTTF, MTBF, MTTR, availability, maintainability. Hardware reliability and software reliability. Techniques for prediction of remaining faults, including fault injection, classification tree analysis, code coverage. General lifecycle techniques for producing reliable software, including defect prevention, early defect detection and removal; design for robustness; use of process measurements; stabilization of requirements, design, code and test artifacts. Active and Passive fault detection. N-version programming, forward and backward check-pointing, recovery blocks, and arbitration techniques. Fault handling and correction, exceptions, fault tolerance. Survivability, critical functions and degraded modes of operation. Data integrity protection.

**19-Soft-B5 Software Modeling & Verification (Formal Methods)**

Mathematical modelling of software, including topics such as programming logics, process algebras, model based specification, object constraint languages, and algebraic specification. Mathematical reasoning using such models, including proofs of program correctness. Tools for static checking of the correctness of software relative to its specification.

**19-Soft-B6 Software Project Management**

Software development lifecycles (sequential, iterative, spiral, agile). Managing software costs: size and effort estimation. Managing risks. Managing software quality. Managing software assets (configuration management, open source software and related IP issues). Software development governance, in particular in regulated environments. Software production and deployment.

**19-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. Technical debt.

**19-Soft-B8 Distributed Systems**

Characteristics of distributed systems. Networked vs. centralized systems. Fundamental concepts and mechanisms. Architectural concepts of distributing an application over several platforms. Overview of network configurations and topologies. Client-server systems. Process synchronization and inter-process communications. Principles of fault tolerance. Transaction processing techniques. Distributed file systems. Operating systems for distributed architectures. Cloud computing. Security.

**19-Soft-B9 Parallel Computing**

Models of parallel computation. Superscalar architecture. Shared memory parallel machines. Interconnection networks and their topological properties. Massively parallel computers. Hypercube architectures. Performance measurement for parallel algorithms. Parallel evaluation of expressions. Parallel searching and data structures. Parallel algebraic and geometric processing.

**19-Soft-B10 Networking and Communications**

Data communications, including signals, modulation, and reception. Channel models and channel capacity. Error detecting and correcting codes. Bit error rate. Data transmission protocols, including half/full duplex, asynchronous/synchronous, point-to-point/multidrop. Character sets, switching alternatives, including circuit and packet. Layered network architecture. Data link and network layer protocols. Transport protocols. Local and wide area networks. Elements of queuing theory. Network performance measures (queue length, delay and throughput). Standards and the standardization process.

**19-Soft-B11 Process Control Systems**

Discrete time models of continuous physical phenomena. Z-transform and transfer functions. Time domain and frequency domain response of first, second and higher order systems. Stability and feedback compensation. Steady state error and proportional, integral and derivative (PID) control. Compensator design using Nyquist criterion and frequency domain design. Sampling theorem, aliasing, anti-aliasing filtering. Design of digital controllers. Software implementation of digital controllers. Computer control interfacing.

**19-Soft-B12 Engineering Computation: Numerics**

Representation of numbers and floating-point round-off. Caveats of computations with floating point.

Linear systems: direct and iterative methods, conditioning, structured systems. Zeros of functions.

Quadrature. Data-fitting methods. Ordinary differential equations: initial value problems, predictor corrector, boundary value problems, systems of ODEs. Simple partial differential equations. Continuous optimization.

**19-Soft-B13 Performance Analysis & Simulation**

Basic techniques of system performance evaluation. Specific topics include: measurement methods and tools, experimental design and analysis, modeling (including queuing and network of queuing systems), discrete event simulation, verification and validation of simulation models, analysis of simulation output, statistical methods (comparing systems using sample data, hypothesis testing and confidence measures).

**19-Soft-B14 Safety Critical Systems**

Safety and hazard analysis. Use of software in safety related systems. Legal and ethical considerations.

Risk analysis techniques: FMEA, HAZOP, FTA, ETA. Safety integrity levels and safety cases, use of GSN

(Goal Structuring Notation). Software reliability. Distinction between safety and reliability of systems.

Achievement of software reliability by fault prevention and fault tolerance. Software design aspects for safety and fault tolerance. Human factors in design for safety. Choice of programming language, safe subsets. Formal methods, algebraic, model and process based specification, formal specification languages, refinement proofs, verification proofs, STAMP/STPA techniques. Fault tolerance, redundancy and common mode failures, N-version programming and recovery blocks. Safety related standards. Certification.

**19-Soft-B15 Artificial Intelligence**

Artificial intelligence; definition and applications. Problem solving: search, adversarial search and constraint solving. Knowledge and Reasoning: agents, logic, planning, knowledge representation. Uncertainty: probabilistic computation and reasoning, decision problems. Learning: from examples, learning models, reinforcement learning, neural networks, deep learning. Communication: natural language processing, perception, robotics.

**19-Soft-B17 Programming Languages, semantics and implementation**

Programming paradigms (procedural, object-oriented, logic and functional). Structuring features (modules, objects, inheritance, polymorphism). Explicit examples from a variety of languages. Abstract syntax. Type systems. Interpretation, compilation, code generation, code transformation, code analysis. Structure and components of compilers. Run-time support. Code optimization.

**19-Soft-B18 Data Visualization**

Data abstractions. Task abstractions. Data analysis and data mining. Pattern discovery. Human visual system, perception. Visual presentations, visual design. Chart types. Maps and networks. Data visualization tools.

***NOTE: Please feel free to use the most recent edition of textbooks referenced in this list NOTA : Utilisez l’édition la plus récente des manuels cités dans cette liste.***

19‐Soft‐A1 Algorithms & Data StructuresAdam Drozdek, Data Structures and Algorithms in C++, Fourth Edition, Cengage Learning, 2013.

19‐Soft‐A2 Computer Architecture and Operating SystemsDavid A. Patterson, John L. Hennessy, Morgan Kaufman, Computer Organization & Design, 1998.  
David A. Patterson, John L. Hennessy, Computer Organization and Design MIPS Edition: TheHardware/Software Interface, 2013 edition.  
Saltzer and Kaashoek, Principles of Computer System Design: An Introduction.

19‐Soft‐A3 Software DesignCarlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, Second  
Edition, Pearson 2003, ISBN 0‐13‐305699‐6.  
Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering, Third Edition, Pearson 2006, ISBN 0‐13‐146913‐4.  
Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices, Prentice Hall,  
2002.

19‐Soft‐A4 Real Time SystemsA. Burns, Andy Wellings, Real‐Time Systems and Programming Languages: Ada, Real‐Time Java andC/Real‐Time POSIX Addison –Wesely, 2009 ISBN‐13: 978‐0321417459.  
Giorgio Buttazzo, Hard Real‐Time Computing Systems, Springer ISBN 978‐1‐4614‐0676‐1

19‐Soft‐A5 Requirements and SpecificationsShari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering, Third Edition, Pearson 2006, ISBN0‐13‐146913‐4.  
Karl E. Wiegers, Software Requirements, Microsoft Press, 1999.

19‐Soft‐A6 Software Quality AssuranceDaniel Galin, Software Quality Assurance: From Theory to Implementation, Pearson, 2003, ISBN‐13: 978‐0201709452.

19‐Soft‐A7 Software Development ProcessShari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering, Third Edition, Pearson 2006, ISBN 0‐13‐146913‐4.

Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, *Fundamentals of Software Engineering*, SecondEdition, Pearson 2003, ISBN 0‐13‐305699‐6.

**19‐Soft‐B1 Advanced Software Design**Bass, L., Clements, P., & Kazman, R., *Software Architecture in Practice*, Third edition, 2012, Reading,MA: Addison‐Wesley.Gorton, I.,*Essential Software Architecture*,Second Edition, 2006, Berlin: Springer.Bernd Bruegge, Allen H. Dutoit, *Object‐Oriented Software Engineering*, Second Edition, Pearson2004.Joshua Bloch, *Effective java*, Addison‐Wesley, ISBN‐13: 978‐0134685991.

**19‐Soft‐B2 User interface**J. Tidwell, O’Reilly, *Designing Interfaces: Patterns for Effective Interaction Design*, ISBN‐13: 978‐1449379704.J. Johnson, Morgan Kauffman, *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, ISBN‐13: 978‐0124079144.

**19‐Soft‐B3 Security**R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, SecondEdition, Wiley, 2008.

**19‐Soft‐B4 Dependable systems**Nancy G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*,2012, MIT Press,ISBN‐13: 978‐0262016629.

**19‐Soft‐B5 Software Modeling & Verification (Formal Methods)**V. S. Alagar and K. Periyasamy, Springer‐Verlag, *Specification of Software Systems*, 1998, ISBN 0‐387‐98430‐5.

**19‐Soft‐B6 Software Project Management**Not available at this time.

**19‐Soft‐B7 Reverse Engineering, Maintenance & Evolution**Not available at this time.

**19‐Soft‐B8 Distributed Systems**Not available at this time.

19‐Soft‐B9 Parallel ComputingNot available at this time.

19‐Soft‐B10 Networking and CommunicationsB. P. Lathi and Z. Ding, Modern Digital and Analog Communication Systems, Fifth edition, Oxford  
University Press, 2018.  
B. A. Forouzan, Data Communications and Networking, Fifth edition, McGraw‐Hill, 2013.  
A. Leon‐Garcia and I. Widjaja, Communication Networks: Fundamental Concepts and KeyArchitectures, McGraw‐Hill, 2004.  
D. Bertsekas and R. Gallager, Data Networks, Second edition, Upper Saddle River, NJ: Prentice Hall, 1991.

19‐Soft‐B11 Process Control SystemsNot available at this time.

19‐Soft‐B12 Engineering Computation: NumericsLaurene V. Fausett, Numerical Methods, Pearson, 2003.  
Singiresu S. Rao, Applied Numerical Methods for Engineers and Scientists, Prentice Hall, 2002.

19‐Soft‐B13 Performance Analysis & SimulationMor Harchol‐Balter, Performance Modeling and Design of Computer Systems: Queueing Theory inAction, Cambridge University Press, 2013, ISBN: 9781107027503.

19‐Soft‐B14 Safety Critical SystemsNancy Leveson, Engineering a safer world, MIT Press, 2012.  
Clifton Ericsson, Software safety primer, CreateSpace, 2013.

19‐Soft‐B15 Artificial IntelligencePeter Norvig and Stuart J. Russell, Artificial Intelligence: A Modern Approach, Pearson, ISBN: 0‐13‐  
604259‐7.

19‐Soft‐B17 Programming Languages, semantics and implementationBenjamin C. Pierce, Types and Programming Languages, MIT Press, ISBN 0‐262‐16209‐1.  
Daniel P. Friedman and Mitch Wand, Essentials of Programming Languages, MIT Press, Third Edition, ISBN‐13: 978‐0262062794.  
Robert Harper, Practical Foundations for Programming Languages, Cambridge University Press,  
Second Edition, ISBN: 9781107150300.  
Keith Cooper and Linda Torczon, Morgan Kaufmann, Engineering: A Compiler, ISBN‐13: 978‐  
0120884780.

19‐Soft‐B18 Data VisualizationTamara Munzner, Visualization Analysis and Design, CRC Press, 2014, ISBN‐13: 978‐1466508910.