**Self-Assessment Form**

**Instructions for Applicants**

**General:**

1. You must use your WES course-by-course (CxC) assessment to complete this form.
	1. When completing the self-assessment form, use the Bachelor’s degree courses.
	2. Only use your Master’s or Ph.D. in engineering **if they are necessary**. If you use too many graduate courses, the degree will **not be** eligible to use for waiving confirmatory exams.
2. Only complete column C2. Do not enter any information in column C3 or C4. If you do, it will be deleted.
	1. Enter the year, course name, credits and grade from the WES assessment Course-by-Course Analysis.
	2. Both the Basic Studies and Discipline Specific Syllabus Tables contain compulsory subjects and elective subjects. Include courses that cover any part of the syllabus even if you have more than the minimum number in the elective sections.
	3. Colour code the content in column C1 by highlighting it the same colour as the corresponding course you entered in column C2.

|  |  |  |  |
| --- | --- | --- | --- |
| **C1** **APEGS Syllabus** | **C2** **Self-Assessment (by applicant)** | **C3** **for Staff only** | **C4** **for ARC only** |
| **COMPULSORY SUBJECTS** **(all required)** | **WES assessment: year, course name, credits and grade.**  | **Program Syllabus: page number, course name** | **Preliminary Review** | **Final Review** |
| **20-BS-A1 Mathematics:** Vector and Linear Algebra: Applications involving matrix algebra, determinants, eigenvalues and eigenvectors, vector functions and operations, orthogonal curvilinear coordinates. Calculus: first and second order linear ordinary differential equations, series solutions of ordinary differential equations, applications of partial derivatives, Lagrange multipliers, multiple integrals, line and surface integrals, integral theorems (Gauss, Green, Stokes). Power series. | 2004-2005: Applied Mathematics I, 2 credits. Grade: B2004-2005: Applied Mathematics II, 2 credits. Grade: B2005-2006: Applied Mathematics III,2 credits. Grade: B |  |  |  |

1. Once you have completed column C2, submit the **Word document** to documents-academicreview@apegs.ca.

**Program Syllabus (only required if requested by APEGS):**

1. Provide the program syllabus in a PDF document through the Contact Us page on the APEGS website.
2. If the course names in the program syllabus are different than those in your WES assessment you must provide an explanation of how they correlate in the program syllabus column of the form.
3. Use the page number of the PDF document of the program syllabus (not the original page number).

***By submitting this self-assessment, I declare that I have read and followed the instructions and that this self-assessment is accurate and complete, to the best of my knowledge and ability, and that I have provided all the relevant information that I have available to me. I understand that if information is incorrect or missing, that it may delay my application and may result in the assignment of academic deficiencies.***

**Self-Assessment Form – Software Engineering**

Use the information provided on the WES assessment to complete this information

|  |  |
| --- | --- |
| **Applicant Information:**  | **Last Name, First Name**  |
|   |
| **APEGS File #** |   |
| **Institution Information** |
| **Credential** | **Awarded By** | **Major/Specialization** | **Year** | **Country** |
|   |  |   |       |       |
|       |       |       |       |       |
|       |   |       |       |       |
|       |       |       |       |       |

**SELF-ASSESSMENT – FOR APPLICANT TO COMPLETE**

**BASIC STUDIES SYLLABUS TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **C1** **APEGS Syllabus** | **C2** **Self-Assessment (by applicant)** | **C3** **for Staff only** | **C4** **for ARC only** |
| **COMPULSORY SUBJECTS** **(all required)** | **WES assessment: year, course name, credits and grade.**  | **Program Syllabus: page number, course name** | **Preliminary Review** | **Final Review** |
| **20-BS-A1 Mathematics:** Vector and Linear Algebra: Applications involving matrix algebra, determinants, eigenvalues and eigenvectors, vector functions and operations, orthogonal curvilinear coordinates. Calculus: first and second order linear ordinary differential equations, series solutions of ordinary differential equations, applications of partial derivatives, Lagrange multipliers, multiple integrals, line and surface integrals, integral theorems (Gauss, Green, Stokes). Power series. |  |  |  |  |
| **20-BS-A2 Probability and Statistics:** Concepts of probability, events and populations, probability theorems, concept of a random variable, continuous and discrete random variables, probability distributions, distributions of functions of a random variable, sampling and statistical estimation theory, hypothesis testing, simple regression analysis. |  |  |  |  |
| **20-BS-A3 Computation Methods:** Use of computers for numerical solution of engineering problems, including techniques involving high-level languages and other computational tools (e.g., spreadsheets). Data representation, approximations and errors. |  |  |  |  |
| **20-BS-A4 Engineering Design Process:** Design process and methods. Project management & teamwork. Requirements and function analysis in design. Conceptual design and testing. Concept evaluation design factors such as: cost, quality, manufacturability, safety, etc. Systems modelling & design detail. |  |  |  |  |
| **20-BS-B2 Electric Circuits and Power:** Current, voltage, Ohm’s law, Kirchoff’s voltage and current laws, power; DC circuits, network theorems, network analysis; simple transients, AC circuits. Impedance concept, resonance; application of phasors and complex algebra in steady-state response; application of Laplace transforms; simple magnetic circuits; basic concepts and performance characteristics of transformers; an introduction to diodes and transistors; rectification and filtering; simple logic circuits. |  |  |  |  |
| **20-BS-B5 Digital Logic Circuits:** Boolean algebra, truth tables and minimization techniques. Logic devices, combinational logic, encoders, decoders and shift registers. Design of asynchronous circuits and synchronous circuits, arithmetic circuits and finite state machines together with clock and timing considerations. Introduction to programmable logic and computer-aided design and simulation tools for digital system design. |  |  |  |  |
| **20-BS-B13 Advanced Mathematics:** Solutions of differential equations, boundary value problems and orthogonal functions, Fourier series, complex variable analysis. |  |  |  |  |
| **C1** **APEGS Syllabus** | **C2** **Self-Assessment (by applicant)** | **C3** **for Staff only** | **C4** **for ARC only** |
| **ELECTIVE SUBJECTS** **(minimum of three required)** | **WES assessment: year, course name, credits and grade.**  | **Program Syllabus: page number, course name** | **Preliminary Review** | **Final Review** |
| **20-BS-B1 Statics and Dynamics:** Force vectors in two- and three-dimensions, equilibrium of a particle in two- and three-dimensions; moments and couples; equilibrium of rigid bodies in two- and three-dimensions; centroids, centres of gravity; second moment of area, moment of inertia; truss, frame and cable static analysis; friction. Planar kinematics of particles and rigid bodies; planar kinetics of particles and rigid bodies; work and energy, impulse, and momentum of particles and rigid bodies. |  |  |  |  |
| **20-BS-B3 Mechanics of Materials:** Definitions of normal stress, shearing stress, normal strain, shearing strain; shear force and bending moment diagrams; members subjected to axial loading; members subjected to torsional loading; compound stresses, Mohr's circle; deformation of flexural and torsional members; failure theories; elastic and inelastic strength criteria; columns. |  |  |  |  |
| **20-BS-B4 Mechanics of Fluids:** Fluid characteristics, dimensions and units, flow properties, and fluid properties; the fundamentals of fluid statics, engineering applications of fluid statics; the one-dimensional equations of continuity, momentum, and energy; laminar and turbulent flow, flow separation, drag and lift on immersed objects; wall friction and minor losses in closed conduit flow; flow of incompressible and compressible fluids in pipes; dimensional analysis and similitude; flow measurement methods. |  |  |  |  |
| **20-BS-B6 Basic Electromagnetics:** Introduction to the fundamental electromagnetic fields and forces used in engineering, including fundamental laws, principles, and equations developed by Gauss, Faraday, Ampere, Kirchoff, Maxwell, leading to electromagnetic design and applications in engineering, such as for capacitors, dielectrics, and magnetic devices. |  |  |  |  |
| **20-BS-B7 Thermodynamics:** Basic concepts and definitions, energy concepts and the first law of thermodynamics, properties of pure substances, closed systems, open systems, the second law of thermodynamics, enthalpy, entropy, exergy, gas power cycles, vapor and combined power cycles, refrigeration cycles. |  |  |  |  |
| **20-BS-B8 Properties of Materials:** Properties of materials for mechanical, thermal and electrical applications. Atomic bonding, solid solutions, crystallisation. Equilibrium phase diagrams, applications to steel and aluminium alloys, heat treatments. Structure and special properties of polymers and ceramic materials. General characteristics of metallic composites, polymeric composites and concrete. Introduction to materials in hostile environments: corrosion, creep at high temperature, refractory materials, subnormal temperature brittle fracture. |  |  |  |  |
| **20-BS-B9 Organic Chemistry:** Principles of organic chemistry developed around the concepts of structure and functional groups. The main classes of organic compounds. Properties of pure substances. Introduction to molecular structure, bond types, properties, synthesis and reactions, reaction mechanisms, as a means of systematizing organic reactions. |  |  |  |  |
| **20-BS-B10 Biology:** Cellular reproduction, growth, and differentiation; metabolism and bioenergetics of living cells; cell structure and function related to the material properties of plant and animal tissues; introductory microbiology — characteristics and classification of microorganisms; interactions of microorganisms with humans in the natural world; kinetics and mathematical models of microbial growth; engineered biological systems such as bio-reactors, bio-instrumentation, bioprinted devices and waste treatment systems for sustainability. |  |  |  |  |
| **20-BS-B11 Geology:** The structure of the earth, plate tectonics, earthquakes and igneous activity. Minerals and rocks including their formation, identification, basic properties, and classification. Processes of weathering, erosion, transport, and deposition of geological materials and their results of significance to engineering. Occurrence, flow, and quality of groundwater. Introductory aspects of structural geology including faulting, folding, and the overall formation of discontinuities and their effect on the engineering properties of rock masses. Aerial photography and geological maps. |  |  |  |  |
| **20-BS-B12 Engineering Graphics:** Engineering drawing: Orthographic sketching. Standard orthographic projection. Principal views, selection and positioning of views. Visualization. Conventions and practices. First and second auxiliary views. Basic descriptive geometry. Section views, types, hatching conventions. Basic dimensioning requirements. Tolerance for fits and geometry control. Detail drawings and assembly drawings, other drawings and documents used in an engineering organization. Bill of materials. Fasteners and welds. |  |  |  |  |

**DISCIPINE SPECIFIC SYLLABUS TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **C1** **APEGS Syllabus** | **C2** **Self-Assessment (by applicant)** | **C3** **for Staff only** | **C4** **for ARC only** |
| **COMPULSORY SUBJECTS** **(all required)** | **WES assessment: year, course name, credits and grade.**  | **Program Syllabus: page number, course name** | **Preliminary Review** | **Final Review** |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-Soft-A4 Real-Time Systems: D**efinition 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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **C1** **APEGS Syllabus** | **C2** **Self-Assessment (by applicant)** | **C3** **for Staff only** | **C4** **for ARC only** |
| **ELECTIVE SUBJECTS** **(minimum of three required)** | **WES assessment: year, course name, credits and grade.**  | **Program Syllabus: page number, course name** | **Preliminary Review** | **Final Review** |
| **20-Soft-B1 Advanced Software Design: S**oftware 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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-Soft-B4 Dependable systems:** 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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-Soft-B6 Software Project Management: S**oftware 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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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). |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-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. |  |  |  |  |
| **20-Soft-B19 Discrete Mathematics:** Logic: propositional equivalences, predicates and quantifiers, sets, set operations, functions, sequences and summations, the growth of functions. Algorithms: complexity of algorithms, the integers and division, matrices. Methods of proof: mathematical induction, recursive definition. Basics of counting: pigeonhole principle, permutations and combinations, discrete probability. Recurrence relations: inclusion-exclusion. Relations and their properties: representing relations, equivalence relations. Introduction to graphs: graph terminology, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths. Introduction to sorting**.** |  |  |  |  |