APEGGA The Association of Professional Engineers, Geologists and Geophysicists of Alberta

SOFTWARE ENGINEERING
SYLLABUS OF EXAMINATIONS
1998 EDITION

GROUP A
COMPULSORY EXAMINATIONS (SEVEN REQUIRED)

98-Soft-A1 Data Structures

Introduction to abstract data types. Objects and methods. Trees, binary search trees, balanced trees. Insertion and deletion. Searching. Sorting. Simple examples of complexity analysis. Graphs, simple graph algorithms: depth-first and breadth-first search, minimum spanning tree, shortest path.

98-Soft-A2 Digital Computer Architecture

Boolean algebra. Gates. Design of combinational and sequential logic. Characteristics of digital integrated circuit families. PLAs, FPGAs, and ULAs. Analysis and design for controllers, processors, and memories. Micro-computers, including components, data flow, signals, and timing. Small system design; interconnection of associated devices. Micro-computer interfacing, including parallel and serial I/O, interrupts, and DMA. Bus structures. Assembly language programming, subroutines, macros. Mainframe, mini-computer, and micro-computer comparisons. Instruction sets and addressing modes; memory management, caches, heaps and stacks. I/O and interrupt servicing. I/O processors and peripheral control Microprogramming.

98-Soft-A3 Operating Systems

Operating system principles, components, and usage. Loaders, assemblers, compilers, and libraries. Design and implementation of operating systems. Synchronization of concurrent processes, resource allocation, scheduling, protection, and privacy. Deadlock and deadlock avoidance. Fragmentation and garbage collection. Data, task and job management; loading; linking; I/O control. Multitasking and multiprocessing. Real-time aspects.

98-Soft-A4 Design and Analysis of Algorithms

Asymptotic time and space complexity of algorithms. Introduction to average case analysis and lower bound theory. Analysis of selected algorithms. Algorithm design techniques, such as divide and conquer, the greedy method, balancing, dynamic programming. Algorithms on graphs. Introduction to the theory of NP-completeness.

98-Soft-A5 Software Design

Analysis, specification, design and coding of software systems using procedural-based and selected object-oriented programming languages, reusable libraries and toolsets. API's. Design using user interface construction tools. Analysis, design and programming of simple servers and clients. State-transition based software design. Experience in the use of industrial-strength design toolsets. BNF and basic theory of grammars and parsing. Use of parser generators. Design of simple languages for applications and programming languages. Overview of design of software using functional or logic-based programming paradigms. Introduction to basic concurrent and parallel architectures.

98-Soft-A6 File and Database Systems

Characteristics of computer files and mass storage media. Algorithms for inverted lists, multilist, indexed sequential and hierarchical structures. File I/O, including control, utility, space allocation, and cataloguing. Index organization. Operations on files. File organization and access methods: sequential, indexed-sequential, direct, extendible hashing, B-trees. Design of file management systems. Concepts and structures for design and implementation. Views of data at different levels. Data models, data normalization, data description languages, query facilities, data integrity and reliability, concurrency. Entity-relationship modeling. The relational data model. Relational algebra and calculi. Relational database languages. Integrity constraints and normal forms. Transactions and concurrency. An introduction to logic-based and object-oriented data models.

98-Soft-A7 Object-Oriented Analysis, Design, and Programming

Object orientation paradigm. Object-oriented analysis: basic concepts, use cases, analysis, stereotypes and objects, analysis patterns. Object modelling languages. Object-oriented design: basic concepts, design stereotypes and objects, design patterns. Object-oriented programming: basic concepts, idioms, most common object-oriented programming languages (C++, Java). Application frameworks. Object-oriented CASE tools. State transition and interaction diagrams. Testing of object-oriented programs.

GROUP B
ELECTIVE EXAMINATIONS (THREE REQUIRED)

98-Soft-B1 Real Time Systems

Definition of real-time systems; where they fit and their general characteristics. Simple Embedded Systems: sequential event response systems and cyclic executives. Designing real-time systems. Exception and exception handling. Resource control. Concurrency and synchronization. Reliability and fault tolerance. Discrete event and real-time systems. Formalization of real-time systems. Real-time operating systems. Real-time system design tools.

98-Soft-B2 Advanced Object-Oriented Analysis and Design

Advanced approaches to object-oriented analysis and design. Frameworks and design patterns. Design for reusability. Advanced object-oriented programming techniques. Design using object-oriented databases and distributed object architectures. Design of software agents. Project involving object-oriented analysis, design, and implementation.

98-Soft-B3 Analysis and Design of User Interfaces

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.

98-Soft-B4 Software Project Management

Advanced project planning, cost estimation and scheduling. Project management tools. Factors influencing productivity and success. Productivity metrics. Analysis of options and risk. Planning for change. Managing expectations. Software contracts and intellectual property. Case studies. Project involving forensic project management of a real industrial project.

98-Soft-B5 Software Development for Large-Scale Systems

The software development life cycle. Introduction to software project management: Cost estimation and project planning. Software specification and requirements analysis. Comparison of software architectures. Design principles such as information hiding, coupling, and cohesion. Introduction to software metrics. Software quality assurance strategies: the personal-software process, reviews, testing and inspections. Assessing software designs for safety, security, performance and economic viability. Use of static and dynamic analysis tools.

98-Soft-B6 Computer Communications Networks

Data communications, including signals, modulation, and reception. Data transmission protocols, including half/full duplex, asynchronous/synchronous, point-to-point/multidrop, and character/bit oriented. Error detecting and correcting codes. Character sets, switching alternatives, including circuit, message, and packet. The OSI model, including physical, data link, and network layer protocols (RS 232C, RS 449-422/23, HDLC, X.25). Transport protocols. Session layer and ISDN. The higher layers. Introduction to network structures and designs. Switching techniques. Resource access/sharing methods (random access, polling, concentration and multiplexing). Public data networks. Local area networks, including bus, ring, and tree topologies, protocols, and hardware. Metropolitan and Wide area networks. Elements of queuing systems. Network performance measures (queue length, delay and throughput). Standards and the standardization process.

98-Soft-B7 Security in Computing

Characteristics of security problems. Basic encryption and decryption; secure encryption, encryption protocols and practices. Public key cryptosystems and digital signatures. Software design for security. Security in operating systems, databases, personal computers, computer networks and electronic communications. Risk analysis and security planning. Ethical issues in computer security.

98-Soft-B8 Software Evolution and Reengineering

Maintainability and reusability analysis. Approaches to maintenance and long-term software development. Change management and impact analysis. Release and configuration management. Software metrics. Software process standards. Reengineering and reverse engineering. Regression testing. Case studies and project in analysis and iterative development of existing large systems.

98-Soft-B9 Computer Graphics

Graphics hardware, architecture, and devices. Vector and raster graphics algorithms and systems. Techniques for describing and generating images: point, vector, and raster approaches. Image transformation, including scaling, translation, rotation, clipping, and windowing. Graphic software and data structures. Geometrical transformations. Viewing in three dimensions. Curves and splines. Picture generation using solid polyhedra. Illumination and color. Ray tracing. Display file compilers. Graphics data structures. Interactive graphics. Graphic standards such as GKS, PHIGS, TIGA, and X-WINDOWS. Virtual reality technology.

98-Soft-B10 Artificial Intelligence and Expert Systems

Artificial intelligence; definition and applications. Concepts of artificial intelligence. Predicate calculus and its use in artificial intelligence Overview of knowledge-based and expert systems. Production systems. Knowledge representations. Search methods; planning. Natural language processing. Programming languages (LISP and Prolog) for AI and expert system implementation. Knowledge representation. Rule-based and object-based systems. Vision.

98-Soft-B11 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 interprocess communications. Principles of fault tolerance. Transaction processing techniques. Distributed file systems. Operating systems for distributed architectures. Security.

98-Soft-B12 Introduction to Compilers

Fundamental features of programming languages, functions of compilers. Processors, preprocessors, linkers and translators. Organization of compilers including compile-time and run-time tables. Lexical and syntactical analysis. Intermediate code generation. Object code generation. Error diagnostic. Code optimization.

98-Soft-B13 Foundations of Modeling and Simulation

Systems definition and properties. System modeling and the simulation process. Numerical solutions for differential equations. ACSL. Elements of discrete event system simulation. Simscript II.5. Simulation validation and quality assurance.

98-Soft-B14 Image Processing

Image acquisition. Image thresholding, edge detection techniques. Contour followers. Image filtering. Mathematical morphology. Image segmentation. Pattern recognition techniques. Matching techniques. 3D object reconstruction. Image compression and related standards.

98-Soft-B15 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.

98-Soft-B16 Microprocessor-Based System
Microprocessors, characteristics and application areas. Single-chip microprocessors; instruction repertoire, programming aspects and architecture. Microprocessor interrupt system. Design alternatives for main-memory and input/output subsystems. Characteristics of microprocessor system-executive. Hardware support for multiprocessors.

98-Soft-B17 Software Quality Engineering
Requirements gathering techniques. Formal approaches to specification of requirements. Use of specification languages and CASE tools to model requirements. Handling uncertain and changing requirements. Certification, verification and validation techniques. Advanced testing techniques. Statistical software reliability engineering. Project.

Revised April 2005