SOFTWARE ENGINEERING EXAMINATIONS

GROUP A
COMPULSORY EXAMINATIONS (EIGHT 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.

21-Soft-A8 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.

GROUP B
OPTIONAL 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
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-B16 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-B17 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.