1. Computer Science and Engineering/Applications)

Digital Logic: Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).
Computer Organization and Architecture: Machine instructions and addressing modes. ALU, data-path and control unit. Instruction pipelining, pipeline hazards. Memory hierarchy: cache, main memory and secondary storage;
Programming and Data Structures: Programming in C. Recursion. Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.
Algorithms: Searching, sorting, hashing. Asymptotic worst case time and space complexity. Algorithm design techniques: greedy, dynamic programming and divide-and-conquer. Graph traversals, minimum spanning trees, shortest paths
Theory of Computation: Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and context-free languages
Compiler Design: Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, common subexpression elimination.
Operating System: System calls, processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU and I/O scheduling. Memory management and virtual memory.
Databases: ER-model, Relational model (Relational algebra, Tuple calculus), Integrity constraints, Normal Forms- 1 NF, 2 NF, 3 NF and BCNF, Query languages (SQL), Transactions and concurrency control.
Software Engineering: Software Process, Modeling Languages, Requirement and feasibility analysis, Process Models, Planning and managing the project, Domain specific modeling, Software architecture and design patterns, Software reliability and testing techniques.