

Post Name: Technical Officer (Software Design & Development/Programmer)

### **SYLLABUS (FOR PAPER-III)**

#### **Section-1: Programming Aptitude Test:**

To test the aptitude required for a programmer/software developer e.g. analytical skill, problem-solving skill, numerical ability, logical reasoning, data interpretation, aptitude to devise programming logic etc.

#### **Section-2: Programming Language/Coding Proficiency Test (C, C++, Java):**

To test programming logic/language constructs/concepts using code snippets of C, C++ & Java programming languages e.g. conceptual understanding of Data Types/ Scope of variables/parameter passing/recursion, OOPs Concepts etc. Questions in the form of output/errors of code snippets etc.

At least 50% questions of this section would be on C, which would be compulsory. Questions on C++ and Java would be in **separate sections** and candidates would be given **option** to attempt questions from any one section only i.e. either C++ or Java.

#### **Section-3: Data Structures & Algorithms:**

Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs. 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

#### **Section-4: DBMS (incl. SQL):**

ER-model. Relational model: relational algebra, tuple calculus, SQL, Integrity constraints, normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control.

#### **Section-5: Digital Logic, Computer Organisation and Architecture, Microprocessor, Theory of Computation, Compiler Design, Operating System, Computer Network, Web programming:**

- i. **Digital Logic:** Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).
- ii. **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; I/O interface (interrupt and DMA mode).
- iii. **Theory of Computation:** Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and context-free languages, pumping lemma. Turing machines and undecidability.
- iv. **Compiler Design:** Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, Data flow

analyses: constant propagation, liveness analysis, common subexpression elimination.

- v. **Operating System:** System calls, processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU and I/O scheduling. Memory management and virtual memory. File systems, distributed systems
- vi. **Computer Networks:** Concept of layering: OSI and TCP/IP Protocol Stacks; Basics of packet, circuit and virtual circuit switching; Data link layer: framing, error detection, Medium Access Control, Ethernet bridging; Routing protocols: shortest path, flooding, distance vector and link state routing; Fragmentation and IP addressing, IPv4, CIDR notation, Basics of IP support protocols (ARP, DHCP, ICMP), Network Address Translation (NAT); Transport layer: flow control and congestion control, UDP, TCP, sockets; Application layer protocols: DNS, SMTP, HTTP, FTP, Email.
- vii. **Web Programming:** HTML, DHTML, XML, Scripting, JavaScript, Servlets, Applets