

GATE-Computer science and IT updated on Jan 2025

How to Prepare for GATE : Computer Science

The GATE tests the basic knowledge that you have in engineering fields. So you should prepare thoroughly by understanding the basics and fundamentals of the subject. A Complete Guide to students for the Graduate Aptitude Test in Engineering by covering the course in depth. This Practice tests for GATE Computer Science Engineering contains chapterwise solved paper of last 12 years, Equipped with multiple sets of practice tests, these online test will help the GATE aspirants evaluate their level of preparedness for the entrance examination.

GATE Computer science & Information Technology online test 1

[GATE Computer science & Information Technology online test 2](#)

[GATE Computer science & Information Technology online test 3](#)

[GATE Computer science & Information Technology online test 4](#)

GATE Computer science & Information Technology online test 5

[GATE Questions from Databases](#)

GATE Computer science & Information Technology online test 6

[GATE Questions from Computer Organization and Architecture](#)

GATE Computer science & Information Technology online test 7

[GATE Questions from Computer Algorithms: Time complexity, Worst & average case analysis](#)

GATE Computer science & Information Technology online test 8

[GATE Questions from Computer Algorithms : Tree and graph traversals](#)

GATE Computer science & Information Technology online test 9

[GATE Questions from Computer Algorithms](#)

GATE Computer science & Information Technology online test 10

[GATE Questions from Computer Algorithms](#)

GATE Syllabus for Computer Science & Information Technology - CS

Engineering Mathematics

Mathematical Logic: Propositional Logic; First Order Logic.

Probability: Conditional Probability; Mean, Median, Mode and Standard Deviation; Random Variables; Distributions; uniform, normal, exponential, Poisson, Binomial.

Set Theory & Algebra: Sets; Relations; Functions; Groups; Partial Orders; Lattice; Boolean Algebra.

Combinatorics: Permutations; Combinations; Counting; Summation; generating functions; recurrence relations; asymptotics.

Graph Theory: Connectivity; spanning trees; Cut vertices & edges; covering; matching; independent sets; Colouring; Planarity; Isomorphism.

Linear Algebra: Algebra of matrices, determinants, systems of linear equations, Eigen values and Eigen vectors.

Numerical Methods: LU decomposition for systems of linear equations; numerical solutions of non-linear algebraic equations by Secant, Bisection and Newton-Raphson Methods; Numerical integration by trapezoidal and Simpsons rules.

Calculus: Limit, Continuity & differentiability, Mean value Theorems, Theorems of integral calculus, evaluation of definite & improper integrals, Partial derivatives, Total derivatives, maxima & minima.

Computer Science & Information Technology - CS

Digital Logic: Logic functions, Minimization, Design and synthesis of combinational and sequential circuits; Number representation and computer arithmetic (fixed and floating point).

Computer Organization and Architecture: Machine instructions and addressing modes, ALU and data-path, CPU control design, Memory interface, I/O interface (Interrupt and DMA mode), Instruction pipelining, Cache and main memory, Secondary storage.

Programming and Data Structures: Programming in C; Functions, Recursion, Parameter passing, Scope, Binding; Abstract data types, Arrays, Stacks, Queues, Linked Lists, Trees, Binary search trees, Binary heaps.

Algorithms: Analysis, Asymptotic notation, Notions of space and time complexity, Worst and average case analysis; Design: Greedy approach, Dynamic programming, Divide-and-conquer; Tree and graph traversals, Connected components, Spanning trees, Shortest paths; Hashing, Sorting, Searching. Asymptotic analysis (best, worst, average cases) of time and space, upper and lower bounds, Basic concepts of complexity classes - P, NP, NP-hard, NP-complete.

Theory of Computation: Regular languages and finite automata, Context free languages and Push-down automata, Recursively enumerable sets and Turing machines, Undecidability.

Compiler Design: Lexical analysis, Parsing, Syntax directed translation, Runtime environments, Intermediate and target code generation, Basics of code optimization.

Operating System: Processes, Threads, Inter-process communication, Concurrency, Synchronization, Deadlock, CPU scheduling, Memory management and virtual memory, File systems, I/O systems, Protection and security.

Databases: ER-model, Relational model (relational algebra, tuple calculus), Database design (integrity constraints, normal forms), Query languages (SQL), File structures (sequential files, indexing, B and B+ trees), Transactions and concurrency control.

Information Systems and Software Engineering: information gathering, requirement and feasibility analysis, data flow diagrams, process specifications, input/output design, process life cycle, planning and managing the project, design, coding, testing, implementation, maintenance.

Computer Networks: ISO/OSI stack, LAN technologies (Ethernet, Token ring), Flow and error control techniques, Routing algorithms, Congestion control, TCP/UDP and sockets, IP(v4), Application layer protocols (icmp, dns, smtp, pop, ftp, http); Basic concepts of hubs, switches, gateways, and routers. Network security basic concepts of public key and private key cryptography, digital signature, firewalls.

Web technologies: HTML, XML, basic concepts of client-server computing.