Last Minute Notes – Discrete Mathematics

Last Updated : 23 Jan, 2025

GATE CSE is a national-level engineering entrance exam in India conducted by top Indian institutions like IISc Bangalore and various IITs. Preparing for the GATE (Graduate Aptitude Test in Engineering) requires strategic planning, especially as the exam day approaches.

Discrete mathematics typically constituting 10% of the total marks. Some of the important topics of Discrete Mathematics are :

- Propositional and First Order Logic
- Sets, Relations and Functions
- Group Theory
- Graph Theory
- Combinatorics

> *Read : Last Minute Notes on all subjects – For GATE Exams*

Propositional Logic

Propositional Logic deals with propositions (simple declarative statements) that can be true or false.

Operators in Propositional logic: Words or symbols that modify or join statements to form more complex statements.

Operator	Term	Precedence
¬	Negation (NOT)	1

Operator	Term	Precedence
$\wedge$	Conjunction (AND)	2
$\vee$	Disjunction (OR)	3
$\rightarrow$	Implication	4
$\leftrightarrow$	Double Implication	5

Propositional Statements and their Meanings

Statement	Meaning
if p, then q	p implies q
if p, q	p only if q
p is sufficient for q	a sufficient condition for q is p
q if p	q whenever p
q when p	q is necessary for p
a necessary condition for p is q	q follows from p
q unless ¬ p	q provided that p

Truth tables for propositional Statements.

p	q	¬p	p ∧ q	p ∨ q	p → q	p ↔ q
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

- **Implication (→)**: The statement "if p then q" is called an implication and it is denoted by p → q. (q occurs when ever p occurs but vice versa is not true).
- **Biconditional (↔)**: The statement "p if and only if(iff) q" is called a biconditional and it is denoted by p ↔ q. (q occurs when ever p occurs and vice versa is also true).

Laws in Propositional Logic

De Morgan's Law

- **¬(p ∧ q) ≡ (¬p ∨¬q)** The negation of a conjunction is equivalent to the disjunction of the negations.
- **¬(p ∨ q) ≡ (¬p ∧¬q)** The negation of a disjunction is equivalent to the conjunction of the negations.

Some other laws :

- **Commutative Laws**
    - **p ∨ q ≡ q ∨ p**
    - **p ∧ q ≡ q ∧ p**
- **Associative Laws**
    - **p ∨ (q ∨ r) ≡ (p ∨ q) ∨ r**
    - **p ∧ (q ∧ r) ≡ (p ∧ q) ∧ r**
- **Distributive Laws**

- **p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r)**
- **p ∧ (q ∨ r) ≡ (p ∧ q) ∨ (p ∧ r)**

- **Identity and Domination Laws**
  - **p ∨ F ≡ P**
  - **p ∨ T ≡ T**
  - **p ∧ T ≡ P**
  - **p ∧ F ≡ P**

- **Inverse Laws**
  - **p ∨ ¬p ≡ T**
  - **p ∧ ¬p ≡ F**

Conditional Statements in Propositional Logic

1. Implication : If p (the antecedent) is true, then q (the consequent) must also be true. If p is false, the statement is true regardless of q's value.

Representation : *p → q*
Example: *If it rains (p), then the ground will be wet (q)*

2. Converse : It represents the reverse of a conditional statement. If q is true, then p must be true.

Representation : *For a statement* ***p → q*** *its converse is represented as* ***q → p***
Example: *If the ground is wet (q), then it must have rained (p).*

3. Contrapositive : Negates and reverses the original conditional statement. If q is false, then p is false.

Representation : *For a statement* ***p → q*** *its contrapositive is represented as* ***(¬q → ¬p)***
Example: *If the ground is not wet (¬q), then it did not rain (¬p).*

4. **Inverse :** Negates both the antecedent and the consequent of the original conditional statement. If p is false, then q is false.

Representation : *For a statement* ***p*** *→* ***q*** *its inverse is represented as* ***(¬p → ¬q)***
Example: *If the ground is not wet (¬q), then it did not rain (¬p).*

5. Biconditional: In a biconditional p is true if and only if q is true. Both p and q must either be true or false together.

Representation : *For a statement* ***p*** *→* ***q*** *its biconditional is represented as* ***(p ↔ q)***
Example: *It rains if and only if the ground is wet.*

Types of propositions based on Truth values

1. Tautology– A **tautology** is a logical proposition or formula that is always **true**, regardless of the truth values of its components.
Example: p ∨¬p

2. Contradiction– A **contradiction** is a logical proposition or formula that is always **false**, regardless of the truth values of its components.
Example: p ∧¬p

3. Contingency – A **contingency** is a logical proposition or formula that is **neither always true nor always false**; its truth value depends on the truth values of its components.
Example: p ∧q

Rules of Inference

A **rule of inference** is a logical rule that specifies valid ways to derive conclusions from premises in a formal argument.

Rule of Inference	Tautology	Name
p p$\to q$ $\therefore q$	(p$\wedge(p \to q)) \to q$	Modus Ponens
$\neg q$ p$\to q$ $\therefore \neg p$	$(\neg q \wedge (p \to q)) \to \neg p$	Modus Tollens
p$\to q$ q$\to r$ $\therefore p \to r$	((p$\to q) \wedge (q \to r)) \to (p \to r)$	Hypothetical syllogism
$\neg p$ p$\vee q$ $\therefore q$	$(\neg p \wedge (p \vee q)) \to q$	Disjunctive Syllogism
p $\therefore (p \vee q)$	p$\to (p \vee q)$	Addition
(p$\wedge q) \to r$ $\therefore p \to (q \to r)$	((p$\wedge q) \to r) \to (p \to (q \to r))$	Exportation
p$\vee q$ $\neg p \vee r$ $\therefore q \vee r$	((p$\vee q) \wedge (\neg p \vee r)) \to q \vee r$	Resolution

Predicate Logic

Predicates are expressions in logic that represent properties, conditions, or relationships involving one or more variables.
Example : P(x) ∧ Q(x)

Quantifiers in predicate : Quantifiers like ∀(for all) and ∃(there exists) are used with predicates to create logical statements:

- ∀x P(x): “For all x, P(x) is true.”
- ∃x P(x): “There exists an x for which that P(x) is true.”

Important equivalences involving quantifiers

- **∀x (P(x) ∧ Q(x)) ≡ ∀x P(x) ∧ ∀x Q(x)**
- **∃x (P(x) ∨ Q(x)) ≡ ∃x P(x) ∨ ∃x Q(x)**

Predicate logic follows all the rules of **propositional logic**, but it extends those rules to include variables, quantifiers, and predicates.

Combinatorics

Combinatoricsis the branch of mathematics that studies the counting, arrangement, and combination of objects within a set under specific rules.

Permutation: A permutation of a set of distinct objects is an ordered arrangement of these objects.

$$\begin{aligned} P(n,r) &= n*(n-1)*...*(n-r+1) \\ &= \frac{n*(n-1)*...*(n-r+1)*(n-r)*...*2*1}{(n-r)*(n-r-1)*....*2*1} \\ &= \frac{n!}{(n-r)!} \end{aligned}$$

Combination: A combination of a set of distinct objects is just a count of the number of ways a specific number of elements can be selected from a set of a certain size. The order of elements does not matter in a combination. and it is given by

$$\begin{aligned} C(n,r) &= \frac{P(n,r)}{P(r,r)} \\ &= \frac{n!}{(n-r)!} * \frac{1}{r!} \\ &= \frac{n!}{r!(n-r)!} \end{aligned}$$

Binomial Coefficients: The combination of r items form a set of n elements is denoted by ^nC_r. This number is also called a binomial coefficient since it occurs as a coefficient in the expansion of powers of binomial expressions.
Let x and y be variables and n be a non-negative integer. Then

$$\begin{aligned} (x+y)^n &= \sum_{j=0}^{n} \binom{n}{j} x^{n-j} y^j \\ &= \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + ... + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n \end{aligned}$$

The binomial expansion using Combinatorial symbols

$$(a+b)^n = {}^nC_0 a^n b^0 + {}^nC_1 a^{n-1} b^1 + {}^nC_2 a^{n-2} b^2 .. + {}^nC_{n-k} a^k b^{n-k} .. + {}^nC_n a^0 b^n$$

Number of elements in the Union of Finite sets

For a number of finite sets, $A_1, A_2, \ldots, A_n$, the number of elements in the union $A_1 \cup A_2 \cup \ldots \cup A_n$ is given by :

$$| A_1 \cup A_2 \cup \ldots \cup A_n | = \sum_i | A_i | - \sum_{i<j} | A_i \cap A_j | + \sum_{i<j<k} | A_i \cap A_j \cap A_k | - \ldots + (-1)^{n+1} | A_1 \cap A_2 \cap A_3 \cap \ldots \cap A_n |$$

Important Binomial Expansion Formulas

Some of the important Binomial Expansion formulas are:

- $\mathbf{(1 + x)^n = {}^nC_0\, x^0\, 1^n + {}^nC_1\, x^1\, 1^{n-1} + {}^nC_2\, x^2\, 1^{n-2} + {}^nC_3\, x^3\, 1^{n-3} + \ldots.. + + {}^nC_n\, x^n\, 1^0}$
- $\mathbf{(1 + ax)^n = {}^nC_0\, a^0x^0\, 1^n + {}^nC_1\, a^1 1x^1\, 1^{n-1} + {}^nC_2\, a^2 2x^2\, 1^{n-2} + {}^nC_3\, a^3x^3\, 1^{n-3} + \ldots.. + + {}^nC_n\, a^nx^n\, 1^0}$

- **$(1 + x^m)^n = {}^nC_0\, x^0\, 1^n + {}^nC_1\, x^m\, 1^{n-1} + {}^nC_2\, x^{2m}\, 1^{n-2} + {}^nC_3\, x^{3m}\, 1^{n-3} + \ldots.. + + {}^nC_n\, x^{nm}\, 1^0$**

- **$(1 - x^{n+1})/(1 + x) = 1 + x + x^2 + x^3 + \ldots.. + x^n$**

- **$1/(1 - x) = 1 + x + x^2 + x^3 + \ldots.. =$** $\sum_{i=0}^{\infty} x^i$

- **$1/(1 - ax) = 1 + ax + a^2x^2 + a^3x^3 + \ldots.. =$** $\sum_{i=0}^{\infty} a^i x^i$

- **$1/(1 + x)^n =$** $\sum_{i=0}^{\infty} {}^{-n}C_i\, x^i = \sum_{i=0}^{\infty} {}^{n+1-i}C_i\, (-x)^i$

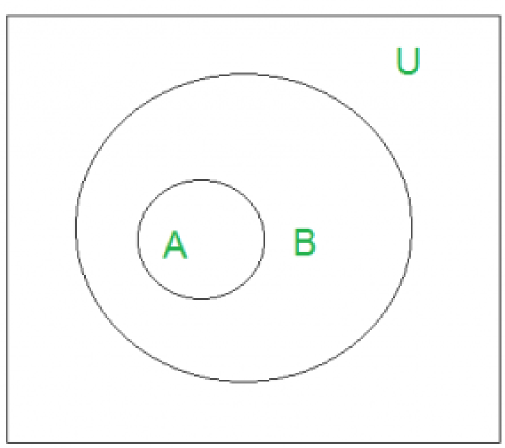
- **$1/(1 + x)^n =$** $\sum_{i=0}^{\infty} {}^{-n}C_i\, (-x)^i = \sum_{i=0}^{\infty} {}^{n+1-i}C_i\, (x)^i$

Set Theory

Set is an unordered collection of objects, known as elements or members of the set.
Example { 1,3,5,4,7, 9,2 }
An element ‘a’ belong to a set A can be written as ‘a ∈ A’, ‘a ∉ A’ denotes that a is not an element of the set A.

- **Equal sets** : Two sets are said to be equal if both have same elements. For example A = {1, 3, 9, 7} and B = {3, 1, 7, 9} are equal sets.
  NOTE: Order of elements of a set doesn’t matter.

- **Subset**: A set A is said to be **subset** of another set B if and only if every element of set A is also a part of other set B. Denoted by ‘⊆‘. ‘A ⊆ B ‘ denotes A is a subset of B.
  If the number of elements in the set B is less then the number of element in the set A then it is called a proper subset and if the number of element in set B is equal to the number of element in set A then it is called an improper subset.



- **Size of a Set**: Size of the set S is known as **Cardinality** ,and it is denoted as |S|.
  Note: Cardinality of a null set is 0.

- **Power Sets**: The power set is the set all possible subset of the set S. Denoted by P(S). **Example:** For a set S = {0, 1, 2} the power set P(S) = {{∅}, {0}, {1}, {2}, {0, 1}, {0, 2}, {1, 2}, {0, 1, 2}}.
  Note: Empty set and set itself is also the member of this set of subsets.

- **Cardinality of power set** : The Cardinality of power set of a set S is 2^n, where n is the number of elements in a set S.

- **Cartesian Products**: Cartesian product of A and B is denoted by A × B, is the set of all ordered pairs (a, b), where a belong to A and b belong to B.
  $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$.

- **Cardinality of Cartesian Product :** The cardinality of A × B is N*M, where N is the Cardinality of A and M is the cardinality of B.
  Note: A × B is not the same as B × A.

Set Operations

- **Union** : Union of the sets A and B, denoted by A ∪ B, is the set of distinct element belongs to set A or set B, or both.
  $A \cup B = n(A) + n(B) - n(A \cap B)$

- **Intersection**: The intersection of the sets A and B, denoted by A ∩ B, is the set of elements belongs to both A and B i.e. set of the common element in A and B

- **Set Difference**: Difference between sets is denoted by 'A – B', is the set containing elements of set A but not in B. i.e all elements of A except the element of B.
  $A - B = n(A) - n(A \cap B)$

- **Complement**: The complement of a set A, denoted by A^c ,It is the set of all the elements in the universal set except A. Complement of the set A is U – A.

Laws Related to Set Operations

- **Idempotent Law**
    - $A \cup A = A$

- A ⋂ A = A

- **Domination Law**
    - A ∪ U = U
    - A ⋂ ϕ = ϕ

- **Identity Law**
    - A ⋂ U = A
    - A **U** ϕ = A

- **Associative Law**
    - **A** ∪ **(B** ∪ **C) = (A** ∪ **B)** ∪ **C**
    - **A** ⋂ **(B** ⋂ **C) = (A** ⋂ **B)** ⋂ **C**

- **Distributive Law**
    - **A** ∪ **(B** ⋂ **C)** = **(A** ∪ **B)** ⋂ **(A** ∪ **C)**
    - **A** ⋂ **(B** ∪ **C)** = **(A** ⋂ **B)** ∪ **(A** ⋂ **C)**

- **De Morgan's Law**
    - $A \cup B = A \cup B$
    - $A \cap B = A \cap B$

Multiset : A **multiset** is a collection of elements where duplicates are allowed, and the number of times an element appears (its "multiplicity") matters. Unlike a regular set, a multiset can have repeated elements. **Example :** {a, a, b, c, c, c} is a multiset where multiplicity of c is 3.
Note : Size of multiset is equal to the sum of all the multiplicities.

Relations

A **relation** R from a set A to a set B is a subset of the Cartesian product A×B. That is: R ⊆ A×B . Each element of R is an ordered pair (a, b), where a ∈ A and b ∈ B, and a is related to b under R.

Number of possible relations on a set A = 2^{n^2} where n is the number of element of set A.
For a relation (x, y) ∈ R

Domain of R = all possible values of x
Range of R = all possible values of y

Inverse of Relation

For a relation R the inverse of the relation is given by R^{-1} . R^{-1} = { (y, x) (x, y) ∈ R }

Types of relations

- **Reflexive :** aRa, ∀ a ∈ A

- **Symmetric :** ∀a, b ∈ A (aRb ⇒ bRa)

- **Antisymmetric :** ∀a**,** ∀b (aRb ∧ bRa ⇒ a = b)

- **Asymmetric:** ∀a**,** ∀b (aRb ∈ R ⇒ bRa ∉ R)

- **Transitive :** ∀a**,** ∀b, ∀c (aRb ∧ bRc ⇒ aRc)

Closure

The **closure of a relation** refers to the smallest relation that contains the original relation and satisfies a specific property (such as reflexivity, symmetry, or transitivity).

- **Reflexive Closure :** The reflexive closure of a relation R on a set A is the smallest relation that contains R and is reflexive. It ensures every element is related to itself.
  Example : Given R={(1,2),(2,3)} on A={1,2,3}, then the reflexive closure $R^{'}$ = {(1,1),(2,2),(3,3),(1,2),(2,3)}.

- **Symmetric closure** : The symmetric closure of a relation R is the smallest relation that contains R and is symmetric. It ensures if (a,b)∈R , then (b,a)∈R.
  Example : Given R={(1,2),(2,3)} on A={1,2,3}, then the symmetric closure $R^{'}$ = {(1,2),(2,1),(2,3),(3,2)}.

- **Transitive Closure**: The transitive closure of a relation R is the smallest relation that contains R and is transitive. It ensures if (a,b)∈R and (b,c)∈R, then (a,c)∈R.
  Example : Given R={(1,2),(2,3)} on A={1,2,3}, then the transitive closure $R^{'}$ = {(1,2),(2,3),(1,3)}.

Equivalence relation

An Equivalence relation is a relation which is reflexive **,** symmetric, and transitive. **Example:** The relation “is equal to” (=) is an equivalence relation on the set of real numbers (R).

Equivalence Class : For an equivalence relation R on a set A. For any element a ∈ A, the **equivalence class** of a, denoted as [a], is defined as: [a]={x∈A∣(a,x)∈R}
This means [a] contains all elements of A that are related to a under the relation R.

For any two equivalence class either **[a] U [b] = ϕ or [a] = [b]**

Partial Order Relation: A **partial order relation** is a binary relation defined on a set that is reflexive antisymmetric and transitive.

POSET: A POSET (partially ordered set) is a set of itemsthat follows a partial order relations it is represented as [A: R] where A is the set in which the partial order relation R is defined. **Example : (R, ≤)** , relation defined by less than equal to on the real numbers.

Hasse Diagram : A **Hasse diagram** is a graphical representation of a **poset** (partially ordered set) that shows the ordering of elements in a simplified way**.**

Minimal and Maximal Elements: In a poset (P,≤), **minimal** and **maximal elements** are defined as follows:

1. **Minimal Element**:
   - An element a ∈ P is minimal if there is no x ∈ P such that x<a (strictly smaller).
   - a has no smaller element in the poset.

2. **Maximal Element**:
   - An element a∈P is maximal if there is no x∈P such that a<x (strictly larger).
   - a has no larger element in the poset.

Note: There can be multiple minimal or maximal elements in a poset.

Least Upper Bound (LUB) : The **least upper bound** of a subset S of a poset P is the smallest element in P that is **greater than or equal to** every element in S. It is also known as supremum and is denoted as sup(S). If u is an upper bound: For all x ∈ S, x ≤ u.

Greatest Lower Bound (GLB) : The **greatest lower bound** of a subset S of a poset P is the largest element in P that is **less than or equal to** every element in S. It is also known as Infimum and Denoted as inf(S). If g is a lower bound: For all x ∈ S, g ≤ x.

Lattice

A**lattice**is a special type of partially ordered set (**poset**) in which **every pair of elements** has both **Least Upper Bound (LUB) and Greatest Lower Bound (GLB).**

Types of lattice :

- **Bounded Lattice**: A lattice with a **least element** (denoted 0) and a **greatest element** (denoted 1).
- **Complete Lattice**: A lattice in which every subset (not just pairs) has both a join and a meet.
- **Distributive Lattice :** A lattice where the operations ∨ and ∧ distribute over each other.

Groups

A **group** is a set G with a binary operation + that satisfies:

1. **Closure**: For all a, b ∈ G a + b ∈ G.
2. **Associativity**: (a + b) + c = a + (b + c) for all a, b, c ∈ G.
3. **Identity Element**: There exists an element e ∈ G such that a + e = e + a = a for all a ∈ G.
4. **Inverse Element**: For every a ∈ G, there exists b ∈ G such that a + b = b + a = e.

Semigroup: A **semigroup** is a set S with a binary operation + that satisfies the conditions for closure and Associativity .

Monoid : A **Monoid** is a set S with a binary operation + that satisfies the conditions for closure and Associativity and has an identity element.

Subgroup : A **subgroup** is a subset H of a group G that itself forms a group under the same binary operation. Intersection of two subgroup is also a subgroup but the union of two subgroups does not necessarily a sub group

Cyclic Group : A **cyclic group** is a group G that can be generated by a single element g, Every element of G can be written as g^n for some integer n.
All sub group of a cyclic group are cyclic.

> *Read more : **Group Theory***

Graph Theory

- No. of edges in a complete graph = n(n-1)/2
- Bipartite Graph : There is no edges between any two vertices of same partition . In complete bipartite graph no. of edges =m*n
- Sum of degree of all vertices is equal to twice the number of edges.
- Maximum no. of connected components in graph with n vertices = n
- Minimum number of connected components is 0 (null graph) and 1 (not null graph)
- Minimum no. of edges to have connected graph with n vertices = n-1
- To guarantee that a graph with n vertices is connected, minimum no. of edges required = {(n-1)*(n-2)/2 } + 1
- A graph is euler graph if it there exists atmost 2 vertices of odd – degree
- Tree
    - Has exactly one path btw any two vertices
    - not contain cycle
    - connected
    - no. of edges = n -1
- For complete graph the no . of spanning tree possible = n^{n-2}
- For simple connected planar graph
    - A graph is planar if and only if it does not contain a subdivision of K_5 and $K_{3, 3}$ as a subgraph.
    - Let G be a connected planar graph, and let n, m and f denote, respectively, the numbers of vertices, edges, and faces in a plane drawing of G. Then n – m + f = 2.
    - Let G be a connected planar simple graph with n vertices and m edges, and no triangles. Then m ≤ 2n – 4.
    - Let G be a connected planar simple graph with n vertices, where n ? 3 and m edges. Then m ≤ 3n – 6.

- Every bipartite graph is 2 colourable and vice versa
- The no. of perfect matchings for a complete graph $(2n)/(2^n n!)$
- The no. of complete matchings for $K_{n.n}= n!$

Next Article

Discrete Mathematics - GATE CSE Previous Year Questions

Similar Reads

PDNF and PCNF in Discrete Mathematics

PDNF (Principal Disjunctive Normal Form)It stands for Principal Disjunctive Normal Form. It refers to the Sum of Products, i.e., SOP. For eg. : If P, Q, and R are the variable...

4 min read

Discrete Mathematics - GATE CSE Previous Year Questions

Discrete Mathematics Previous Year GATE Questions help analyze the question pattern, marking scheme, and improve time management, boosting GATE scores....

2 min read

Discrete Mathematics | Types of Recurrence Relations - Set 2

Prerequisite - Solving Recurrences, Different types of recurrence relations and their solutions, Practice Set for Recurrence Relations The sequence which is defined by...

4 min read

Discrete Mathematics | Representing Relations

Prerequisite - Introduction and types of Relations Relations are represented using ordered pairs, matrix and digraphs: Ordered Pairs - In this set of ordered pairs of x an...

2 min read

Types of Proofs - Predicate Logic | Discrete Mathematics

The most basic form of logic is propositional logic. Propositions, which have no variables, are the only assertions that are considered. Because there are no variables i...

13 min read

Four Color Theorem and Kuratowski's Theorem in Discrete Mathematics

The Four Color Theorem and Kuratowski's Theorem are two fundamental results in discrete mathematics, specifically in the field of graph theory. Both theorems address...

14 min read

Prime Numbers in Discrete Mathematics

Prime numbers are the building blocks of integers, playing a crucial role in number theory and discrete mathematics. A prime number is defined as any integer greater...

9 min read

Types of Sets in Discrete Mathematics

A set in discrete mathematics is a collection of distinct objects, considered as an object in its own right. Sets are fundamental objects in mathematics, used to define various...

12 min read

Hasse Diagrams | Discrete Mathematics

A Hasse diagram is a graphical representation of the relation of elements of a partially ordered set (poset) with an implied upward orientation. A point is drawn for each...

10 min read

Discrete Mathematics for GATE CSE Exam

Discrete Mathematics plays an important role in the GATE (Graduate Aptitude Test in Engineering) examination, particularly for GATE CSE aspirants aiming for disciplines...

4 min read

Free Online Tools

Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

DSA/Placements

DSA - Self Paced Course
DSA in JavaScript - Self Paced Course
DSA in Python - Self Paced
C Programming Course Online - Learn C with Data Structures
Complete Interview Preparation
Master Competitive Programming
Core CS Subject for Interview Preparation
Mastering System Design: LLD to HLD
Tech Interview 101 - From DSA to System Design [LIVE]
DSA to Development [HYBRID]
Placement Preparation Crash Course [LIVE]

Development/Testing

JavaScript Full Course
React JS Course
React Native Course
Django Web Development Course
Complete Bootstrap Course
Full Stack Development - [LIVE]
JAVA Backend Development - [LIVE]
Complete Software Testing Course [LIVE]
Android Mastery with Kotlin [LIVE]

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]
Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
Data Science Training Program - [LIVE]
Mastering Generative AI and ChatGPT
Data Science Course with IBM Certification

Programming Languages

C Programming with Data Structures
C++ Programming Course
Java Programming Course
Python Full Course

Clouds/Devops

DevOps Engineering
AWS Solutions Architect Certification
Salesforce Certified Administrator Course

GATE

GATE CS & IT Test Series - 2025
GATE DA Test Series 2025
GATE CS & IT Course - 2025
GATE DA Course 2025
GATE Rank Predictor