Zoho Drive Pattern — 2025

Overall Structure

Round	Focus	Expected Format	Difficulty
Round 1: Online MCQ	C fundamentals + Aptitude	20 questions (10 C, 10 Aptitude)	Easy–Medium
Round 2: Short Programming (L2)	Logical pattern + basic algorithm	Live coding with interviewer (C, or Java)	Medium
Round 3: Long Programming (L3)	Advanced logic + parsing/ simulation problem	1–2 coding problems, 2–3 hrs	Hard
Round 4: Technical Interview	OOP + Core CS + Resume/ Projects	60–90 min	Medium–Hard
Round 5: HR Interview	Personality + Company fit	10–15 min	Easy

Round 1 — Online Aptitude + C Basics

Pattern (based on 2024–25 reports):

- 10 C Programming MCQs → pointers, arrays, functions, macros, recursion.
- 10 Aptitude → number systems, time-speed-distance, probability, logic puzzles.

Preparation Tip

- Practice from SkillRack / Zoho-type C sheets.
- Focus on:
  - Function return & scope
  - Pointer to arrays vs array of pointers
  - #define pitfalls
  - Dry run small recursion logic
- 20 Qs in 60 mins (negative marking possible).

Round 2 — Short Programming (L2)

Structure:

- **Section A (Pattern Logic)** → 30 min cutoff.
- **Section B (2 Algorithmic Qs)** → 2 hr 30 min (max).
- Language of choice (Java/C++).
- Live screen sharing with discussion.

Type	Example	Concept Tested
Pattern Problem (no array)	Number or diagonal pattern (like Ragav's "triangle increasing" pattern)	Nested loops, controlled spacing, integer formatting
String Subproblem	"Longest substring between vowels" / "Remove vowels and get max substring"	String traversal, condition checking
Array Traversal	"Next greater element on right"	Nested loops / stack optimization
Prefix-Sum Logic	"Equal 0s and 1s → max subarray"	HashMap / prefix-sum technique

Prediction:

1 Pattern Problem (nested loop) + 1 String or Array manipulation (2 pointer or prefix-sum based).
Languages — interviewer may prefer C/Java for L3.

Round 3 — Long Programming (L3)

Trend: Real-time simulation or algebraic expression parser.
Sample question → *(2x+y)*(3x–5y)* → expansion logic.*

Possible Topics

Category	Example Problem Type
Expression Evaluation / Expansion	Multiply or simplify polynomial expressions
Parser / String Simulation	Evaluate arithmetic or nested expressions
Matrix / Equation Simulation	Sparse matrix multiplication
Custom DS Simulation	LFU/LRU Cache, mini DB query interpreter
String→Token→Evaluate	Basic calculator or postfix evaluator

Prediction:

One **large, real-world algebraic or simulation-style problem** — requiring parsing + logic formulation + modular coding.
Language → likely **Java or C++**, Python often disallowed.

Round 4 — Technical Interview

Focus Areas:

- Core CS: OS concepts (process vs thread, scheduling), OOP (Java), DBMS (joins, relationships).
- Resume-based project discussion (GitHub expected).
- Basic DSA (arrays, linked list, recursion).
- Behavioral logic: "Why this approach?" / "Can you optimize?"

Revise Before :

- **OOP:** encapsulation, inheritance, polymorphism (with example).
- **DBMS:** normalization, joins, real ER mapping.
- **OS:** deadlock, scheduling, semaphores.
- **Code Demo:** at least 2 mini projects ready to explain.

Round 5 — HR Interview

Pattern

Friendly tone, short, value-based.

- “Why Zoho?”, “Where do you see yourself in 5 years?”, “Higher studies?”, “If offered lesser package, would you join?”

Advice:
Be natural, confident, emphasize *learning and long-term growth over package.*

Key Takeaways (Zoho 2025)

Priority	What to Expect	Your Prep Focus
🥇	**Pattern + String + Array problem** (L2)	Nested loops, dry run without arrays, substring logic
🥈	**Algebraic Expression Parsing (L3)**	String tokenization, maps, expression multiplication
🥉	**Core CS (Technical)**	Java OOP, DBMS, OS basics
⭐	**C Fundamentals (Online Round)**	Pointers, macros, recursion flow
💬	**Soft skills (Interview)**	Confidence + authenticity + curiosity

Strategic Move:
Tonight, quickly revise:

- Pattern printing without arrays
- Substring between vowels
- Next greater element logic
- Equal 0s & 1s subarray (prefix-sum HashMap)
- Expression multiplication parser (tokens × coefficients)