

⚙️ SYSTEM_CONFIG_v10_FINAL.md — Mental-Model-First Configuration

Version: 10.0 (Unified Mental-Model-First) Scope: All instructional and support files in DSA Master Curriculum v9.2 Last Updated: December 31, 2025 Status: OFFICIAL SYSTEM CONFIGURATION

⌚ SYSTEM IDENTITY & PHILOSOPHY

Curriculum Name: DSA Master Curriculum

Version: 9.2 (Content), Config v10.0 (Structure & Philosophy)

Status: FINAL — OPERATIONAL

Core Philosophy:

Data Structures & Algorithms is **not** about memorizing code or copying LeetCode answers. It is about:

- Building **mental models** of how structures and algorithms behave mechanically
- Understanding **computational trade-offs** (time, space, locality, simplicity vs complexity)
- Recognizing **patterns** across problems and systems
- Developing **engineering intuition**: "What should I use here, and why?"

This configuration enforces:

- **Understanding first, code second (or never)**
 - **Systems-level thinking**: RAM model, caches, virtual memory, real systems
 - **Visual-first explanations**: diagrams, tables, flows, traces
 - **Graduate-level engineer tone**: mentor-style, not textbook-style
-

📄 INSTRUCTIONAL FILE REQUIREMENTS (v10)

STRUCTURE Structure & Sections

Every instructional file MUST follow the template [Template_v10.md](#) and include:

- **Header Block** with:
 - Week, Day, Topic
 - Category
 - Difficulty (🟡 / 🔵 / 🟣)
 - Prerequisites
 - Interview Frequency (approximate)
 - Real-World Impact (short)
- **11 Sections in Order:**
 1. 🧠 The Why — Engineering Motivation
 2. 💡 The What — Mental Model & Core Concepts
 3. 🚧 The How — Mechanical Walkthrough
 4. 📈 Visualization — Simulation & Examples
 5. 📊 Critical Analysis — Performance & Robustness

6. Real Systems — Integration in Production
7. Concept Crossovers — Connections & Comparisons
8. Mathematical & Theoretical Perspective
9. Algorithmic Design Intuition
10. Knowledge Check — Socratic Reasoning
11. Retention Hook — Memory Anchors

- **Mandatory Blocks (after Section 11):**

- 5 Cognitive Lenses (Computational, Psychological, Trade-off, AI/ML, Historical)
- Supplementary Outcomes:
 - Practice Problems (8–10)
 - Interview Questions (6+ with follow-ups)
 - Common Misconceptions (3–5)
 - Advanced Concepts (3–5)
 - External Resources (3–5)

Content Expectations

- **Mental Model First:**

- Clear analogy and visual representation (Section 2)
- Explicit invariants and core variations listed
- Mechanical, step-by-step operations (Section 3)
- Worked traces with diagrams (Section 4)

- **Coverage:**

- All major **subtopics / variations / operations / patterns** for that day's topic must appear in Section 2
- 3+ examples:
 - Simple
 - Medium
 - Edge / stress case
- **5–10 real systems** in Section 6, across:
 - Operating Systems (Linux, Windows, macOS)
 - Databases (PostgreSQL, Redis, MongoDB, etc.)
 - Networks & Web (TCP/IP stack, Nginx, DNS, CDNs)
 - Applications (Browsers, Search Engines, Compilers)
 - Cloud/Distributed (AWS, Kafka, Docker, Kubernetes)

Visual & Structural Requirements

To keep files **visual and scannable**:

- Each instructional file MUST include at least:
 - **1 concept summary or comparison table** (typically Section 2 or 7)
 - **1 complexity table** (Section 5)
 - **2–3 visuals overall**, which can be:

- ASCII diagrams
- Markdown tables
- Simple Mermaid flowcharts / graphs (where helpful)
- Recommended visual placements:
 - Section 2–4: at least 1 diagram or trace
 - Section 5: complexity table
 - Section 7 or 9: a decision/comparison table or minimal flowchart

Word Count

- **Per instructional file (total):**
 - Target: **7,500–15,000 words**
 - No per-section quotas; use space where it adds insight and clarity.
-

FORMAT & CODE POLICY

- **File Format:**
 - Markdown `.md` only
 - UTF-8 encoding
 - LF line endings
 - No embedded HTML unless absolutely necessary for clarity
 - **No LaTeX:**
 - Do not use LaTeX syntax or math encoding (`\frac`, `\(\)`, etc.)
 - Use plain text math: `O(n log n)`, "n squared", etc.
 - **Code Policy:**
 - No code by default — explanations must be logic-first, language-neutral
 - If absolutely required for clarity:
 - Use **C# only**, minimal and focused on the algorithm's logic
 - No Python, Java, C++, or other languages in code blocks
-

COGNITIVE LENSES REQUIREMENTS

Every instructional file must include a **Cognitive Lenses** block:

-  Computational Lens
-  Psychological Lens
-  Design Trade-off Lens
-  AI/ML Analogy Lens
-  Historical Context Lens

Each lens:

- 1 short paragraph or a small bullet list/table

- Focuses on:
 - Hardware reality (computational)
 - Human intuition/misconceptions (psychological)
 - Engineering decisions (trade-off)
 - Modern ML analogies (AI/ML)
 - Origin and evolution (historical)
-

REAL SYSTEMS INTEGRATION REQUIREMENTS

For each instructional file:

- Mention **5–10 specific systems**.
- For each system:
 - **Name & Domain**: e.g., "PostgreSQL (Relational Database)"
 - **Problem Solved**: what this concept helps with
 - **Implementation Detail**: how the concept is used internally (at a conceptual level)
 - **Impact**: performance, scalability, robustness, or simplicity improvements

Categories to spread across:

-  OS Kernels & Runtimes (Linux, Windows, JVM, .NET)
 -  Databases & Storage (PostgreSQL, MySQL, Redis, RocksDB)
 -  Networking & Web (TCP/IP, Nginx, CDNs, DNS)
 -  High-level Apps (browsers, search engines, compilers, analytics engines)
 -  Cloud/Distributed (AWS services, Kafka, Docker, Kubernetes, load balancers)
-

SUPPORT FILE REQUIREMENTS (PER WEEK)

For each week, generate at least **5 support files**, using the mental-model-first style:

1. Week_X_Guidelines.md

- Weekly learning objectives
- Key concepts overview
- Learning approach & methodology
- Common mistakes & pitfalls
- Time & practice strategy
- Weekly checklist

2. Week_X_Summary_Key_Concepts.md

- Week overview (short)
- Per-day key concept summaries
- Concept map (ASCII)
- Comparison and relationship tables
- Key insights & misconceptions fixed

3. Week_X_Interview_QA_Reference.md

- 30–50 interview-style questions (no answers, unless explicitly chosen)
- Grouped by topic/day
- Each with follow-up variations

4. Week_X_Problem_Solving_Roadmap.md

- Simple → complex practice progression
- Strategy for mixing problems and reviewing
- Common problem-solving pitfalls
- Pattern templates for major techniques

5. Week_X_Daily_Progress_Checklist.md

- Practical day-by-day checklists
- Activities: reading, tracing, drawing diagrams, practicing problems
- Weekly integration/reflection section

All support files:

- Must be **Markdown**, no LaTeX, no code blocks.
 - Should be **concise and practical**, not essay-heavy.
-

📁 FILE NAMING & FOLDER STRUCTURE

Instructional Files:

- Week_X_Day_Y_[Topic_Name]_Instructional.md
 - Example: Week_1_Day_1_RAM_Model_And_Pointers_Instructional.md

Support Files:

- Week_X_Guidelines.md
- Week_X_Summary_Key_Concepts.md
- Week_X_Interview_QA_Reference.md
- Week_X_Problem_Solving_Roadmap.md
- Week_X_Daily_Progress_Checklist.md

Folder Layout:

```
WEEKS/
Week_01/
  Instructional_Files/
    Week_1_Day_1_RAM_Model_And_Pointers_Instructional.md
    ...
  Support_Files/
    Week_1_Guidelines.md
    Week_1_Summary_Key_Concepts.md
    Week_1_Interview_QA_Reference.md
    Week_1_Problem_Solving_Roadmap.md
    Week_1_Daily_Progress_Checklist.md
```

```
...
CORE_CURRICULUM/
    TEMPLATE_v10.md
    SYSTEM_CONFIG_v10_FINAL.md
    MASTER_PROMPT_v10_FINAL.md
    COMPLETE_SYLLABUS_v10_FINAL.md
```

📞 QUALITY GATES & CHECKLIST

An instructional file is **REJECTED** if it:

- ✗ Is missing any of the 11 main sections
- ✗ Lacks the Cognitive Lenses block
- ✗ Lacks the Supplementary Outcomes block
- ✗ Contains LaTeX or non-C# code
- ✗ Has no complexity table
- ✗ Mentions fewer than 5 real systems
- ✗ Has fewer than 8 practice problems or 6 interview Qs
- ✗ Has obviously shallow or purely surface-level explanations
- ✗ Does not list all core subtopics/variations in Section 2

Recommended acceptance criteria:

- Total length ~7,500–15,000 words
- 3+ worked examples with diagrams/traces
- Clear mental model and invariants
- Concrete mechanical walkthroughs (state changes, operations)
- Real systems connections are specific and believable
- Misconceptions and edge cases are addressed explicitly

⌚ MAINTENANCE & EVOLUTION

- **Weekly:**
 - Incorporate learner feedback, fix any confusing explanations or diagrams.
- **Monthly:**
 - Update external links and resources.
 - Refresh real systems examples to stay current.
- **Quarterly:**
 - Audit a selection of instructional files for depth and clarity.
 - Refine cognitive lenses and design trade-off discussions.

Success Metrics:

- 100% instructional files have all 11 sections + lenses + supplementaries

- 100% support files exist for each week
- High learner ratings on clarity, intuition, and perceived real-world relevance

Status: Configuration aligned with Template_v10 (Mental-Model-First)