📋 MASTER_PROMPT_v10_FINAL.md — Mental-Model-First Generation Prompt

Version: 10.0 (Aligned with Template_v10 & SYSTEM_CONFIG_v10)
Status: ☑ OFFICIAL — Master control document for instructional file generation
Last Updated: December 31, 2025

🎯 PURPOSE

This master prompt defines **how AI should generate instructional files** for the DSA Master Curriculum v9.2 under the **mental-model-first** philosophy.

All generated instructional files MUST:

- Follow `Template_v10.md`
- Satisfy `SYSTEM_CONFIG_v10_FINAL.md`
- Emphasize **mental models, mechanical understanding, trade-offs, and pattern recognition**
- Use **visuals (tables, diagrams, flows)** to clarify, not clutter

---

🔍 QUALITY STANDARDS (PER INSTRUCTIONAL FILE)

☑ Structural Requirements

Each instructional file MUST have:

1. Header (Week, Day, Topic, Category, Difficulty, Prerequisites, Interview Frequency, Real-World Impact)
2. 11 sections in order:
    - 🧠 The Why — Engineering Motivation
    - 📌 The What — Mental Model & Core Concepts
    - ⚙ The How — Mechanical Walkthrough
    - 🎨 Visualization — Simulation & Examples
    - 📊 Critical Analysis — Performance & Robustness
    - 🏭 Real Systems — Integration in Production
    - 🔗 Concept Crossovers — Connections & Comparisons
    - 📐 Mathematical & Theoretical Perspective
    - 💡 Algorithmic Design Intuition
    - ❓ Knowledge Check — Socratic Reasoning
    - 🎯 Retention Hook — Memory Anchors
3. 🧩 Cognitive Lenses block with 5 lenses
4. ⚔ Supplementary Outcomes block:
    - 8–10 practice problems
    - 6+ interview questions (with follow-ups)
    - 3–5 misconceptions
    - 3–5 advanced concepts

    - 3–5 external resources

🖊 Content & Coverage

- **Mental Model & Variations:**
    - Section 2 must list **all core concepts/variations/subtypes/operations** for that topic.
    - Include invariants and an intuitive analogy.
- **Mechanics & Simulation:**
    - Section 3 must walk through core operations step-by-step, describing state changes.
    - Section 4 must include at least **3 examples**:
        - simple
        - medium
        - edge/stress case
    - At least one counter-example showing a common incorrect approach.
- **Real Systems:**
    - Section 6 must mention **5–10 real systems** with context, implementation detail (at conceptual level), and impact.
- **Trade-offs & Decisions:**
    - Section 5 must include a **complexity table** and discuss where Big-O misleads.
    - Section 9 must offer a **decision framework** (table or small flow): when to use vs when not to use this concept.
- **Supplementary:**
    - Practice problems must cover all major variations and patterns in the topic.
    - Interview questions must be realistic and include follow-ups probing deeper understanding.
    - Misconceptions must be meaningful (not trivial) and tied to actual confusion points.

🎨 Visual Requirements

Each instructional file should contain:

- **1+ concept summary / comparison table** (often in Section 2 or 7)
- **1 complexity table** in Section 5
- **2–3 visuals overall**:
    - ASCII diagrams showing structures or traces
    - Markdown tables summarizing states or comparisons
    - Simple Mermaid flows for algorithm steps or decision frameworks (if useful)

⏱ Word Count

- Target total: **7,500–15,000 words** per instructional file
- No per-section word quotas — use length where it adds clarity, depth, or visual explanation.

📋 Format & Code

- Output must be **Markdown only**.
- ❌ No LaTeX math or LaTeX-like encoding.
- ❌ No code by default.
- If absolutely necessary for clarity:
  - ☑ Minimal **C#** code only, logic-first
  - ❌ No other languages

🧠 GENERATION WORKFLOW (INSTRUCTIONAL FILES)

Step 1: Topic Selection & Concept Mapping

1. Read `COMPLETE_SYLLABUS_v10_FINAL.md` and identify:
   - Week X, Day Y, Topic Name
   - Category (Foundations / Patterns / Data Structure / Graph / DP / etc.)
2. Enumerate **all core concepts/variations** for the topic:
   - Types, variants, patterns, operations
   - Make sure none are omitted
3. Identify the **role** of this topic:
   - What problems it solves
   - What earlier topics it builds on
   - What later topics depend on it

Step 2: Template Initialization

Use `Template_v10.md`:

- Fill in header:
  - Week, Day, Topic, Category, Difficulty, Prerequisites, Interview Frequency, Real-World Impact
- Keep all section headings and emojis intact:
  - Do **not** reorder or remove sections

Step 3: Content Generation by Section

Section 1 – The Why (Engineering Motivation):

- Start with **2–3 real-world problems** where this concept is crucial.
- Explain the **design problem** the concept solves and associated trade-offs.
- Clarify **interview relevance** and typical question archetypes.

Section 2 – The What (Mental Model & Core Concepts):

- Introduce a **core analogy** that makes the concept intuitive.
- Provide a **main diagram** (ASCII or Mermaid) showing its “shape”.
- List **invariants** that must always hold.

- List all **core concepts/variations** with short descriptions and rough complexities.
- Include a **concept summary table**.

Section 3 – The How (Mechanical Walkthrough):

- Define the **state/data structure** clearly (what is stored, where, how).
- For each core operation:
  - Explain step-by-step what changes in state.
  - Mention time/space cost.
- Describe **memory behavior**: stack/heap, pointer use, locality.
- List **edge cases** and how operations handle them.

Section 4 – Visualization (Simulation & Examples):

- Provide at least **3 examples**:
  - Simple (warm-up)
  - Medium (more representative)
  - Edge/stress case
- For each:
  - Show input
  - Show initial state diagram
  - Provide a **trace table** of steps and state changes
- Include a **counter-example**: common mistake and how it fails.

Section 5 – Critical Analysis (Performance & Robustness):

- Fill out a **complexity table** for core operations/variants.
- Explain where Big-O is misleading (e.g., same complexity, different constants or memory behavior).
- Discuss **real memory behavior**: caches, TLB, allocation patterns.
- List 3–5 **failure modes** and how to avoid them.

Section 6 – Real Systems:

- Identify 5–10 real-world systems using this concept.
- For each:
  - Problem solved
  - Implementation detail
  - Impact on performance/robustness

Section 7 – Concept Crossovers:

- List 3–5 **prerequisites** and how they show up inside this topic.
- List 3–5 **successor concepts** and how they build on this one.
- Add a **comparison table** with key alternatives.

Section 8 – Mathematical & Theoretical:

- Provide a formal-ish definition if applicable.
- State one **key theorem/property** and give a short proof sketch.
- Optionally mention RAM/external memory or other models if relevant.

Section 9 – Algorithmic Design Intuition:

- Build a **decision framework** (table/flow): when to use vs avoid this concept.
- Highlight **red flags** (obvious cues) and **blue flags** (subtle cues) in interviews.

Section 10 – Knowledge Check:

- Write 3–5 **Socratic questions**:
  - Why this works
  - Edge cases
  - Comparisons
  - Real-system tie-ins
- Do not provide answers.

Section 11 – Retention Hook:

- One-sentence essence.
- Mnemonic (acronym or phrase) with quick explanation.
- One **visual cue** (small diagram / "logo").
- Short interview story showing the concept in action.

Cognitive Lenses Block:

- For each lens (Computational, Psychological, Trade-off, AI/ML, Historical):
  - 1 short paragraph or bullet list
  - Provide at least one concrete insight tied to the topic.

Supplementary Outcomes:

- **Practice Problems (8–10):**
  - Title, Source, Difficulty, Key Concepts, Constraints
  - No solutions.
- **Interview Questions (6+):**
  - Question + 2+ follow-ups
  - No answers in instructional file.
- **Common Misconceptions (3–5):**
  - Misconception, why it seems plausible, reality, memory aid, impact.
- **Advanced Concepts (3–5):**
  - Title, relation to core concept, when/why it's useful.
- **External Resources (3–5):**
  - Type, author, why useful, difficulty, link/citation.

Step 4: Word Count & Visual Check

- Ensure estimated total length is **within 7,500–15,000 words**.
- Verify:
  - At least 1 **concept summary/comparison table**
  - 1 **complexity table**
  - 2–3 **visuals** (diagrams, traces, or flows)

Step 5: Quality Checklist

Verify against `SYSTEM_CONFIG_v10_FINAL.md`:

- 11 sections + Cognitive Lenses + Supplementary present
- All core subtopics/variations listed in Section 2
- 5–10 real systems in Section 6
- 8–10 practice problems, 6+ interview Qs
- 3–5 misconceptions, 3–5 advanced concepts, 3–5 resources
- No LaTeX, no non-C# code; Markdown only
- Visuals present (tables, diagrams, flows)

Step 6: File Naming & Submission

- Name the file: `Week_X_Day_Y_[Topic_Name]_Instructional.md`
- Ensure:
  - Correct Week and Day
  - Topic name with underscores, capitalized words
- Place under the appropriate week's `Instructional_Files` folder.

📓 SUPPORT FILE GENERATION (HIGH-LEVEL)

For each week:

- Generate the 5 support files using the templates in `Template_v10.md`:
  - `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`

Guidelines:

- Keep them **concise, practical, and visual**.
- Emphasize how to **study**, **integrate**, and **review** concepts, not restating full explanations.
- No LaTeX, no code blocks.

☑ SUMMARY

When generating any instructional file:

- Think like a **senior engineer teaching a strong junior**.

- Start from **why**, build a **mental model**, walk through **mechanics**, show **examples**, then discuss **trade-offs and systems**.
- Only then consider code (if at all), and even then, as minimal C# for clarity.

Status: ☑ MASTER PROMPT ALIGNED WITH Template_v10 & SYSTEM_CONFIG_v10