

## Part A — Theory (Short Questions)

### 1. Nine Pillars Understanding

- Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?

Ans. Using AI Development Agents (like Gemini CLI) for repetitive task better for our growth because it gives us leverage of not thinking about manual or time-taking task like writing code manually rather we can clearly focus on system designs, architectures, complex problem solving. The AI Agents don't replace your skills, but they amplify it, allowing you to spend more time on architectural principles rather than spend time on trivial tasks like manual coding.

- Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer.

Ans. The M-Shaped developer isn't about being good at everything. It's about choosing 2-4 complementary domains where you can maintain deep expertise with AI augmentation. The Nine Pillars of AI-Driven Development help a developer become an M-shaped developer by combining deep expertise in their core area with broad skills across other domains. AI agents and Markdown as code let you handle tasks outside your main expertise quickly. MCP and AI-first IDEs simplify tools and reduce mental load, while a universal Linux environment and cloud deployment make your skills portable. Test-driven and specification-driven development ensure quality, and composable skills let you leverage domain knowledge without mastering everything. Together, these pillars let you work confidently across multiple areas while staying strong in your specialty.

### 2. Vibe Coding vs Specification-Driven Development

Why does Vibe Coding usually create problems after one week?

- As in vibe coding there are no clear specs so in this case in future you may misunderstand how things work or are expected to work. Or even the other developer teams can get confused.
- So many bugs will appear in the code if you modify or extend your code, and all get so much messy that no one understands the generated code now.
- You don't consider all edge cases and due to which after a week or sometime, new bugs arrive in your code.

How would Specification-Driven Development prevent those problems?

- All specs are clearly documented in a file, so in future you will be clear enough that how things are working, and other developers can also understand the workflow by reading the spec file.
- If you want to modify or extend code, you will first modify your spec file and then make changes to code; code and specs will be aligned, no messiness.
- Specification includes all scenarios and edge cases, so less chances of hidden bugs

### 3. Architecture Thinking

- How does architecture-first thinking change the role of a developer in AIDD?

In AI-Driven Development, architecture-first thinking changes a developer's role from just writing code to designing and orchestrating entire systems. Developers focus on how components, AI agents, and workflows interact, ensuring scalability, maintainability, and integration across domains.

- Explain why developers must think in layers and systems instead of raw code. Developers need to think in layers and systems instead of just raw code because big apps are like each piece that must fit together, or the whole thing breaks. Don't think about raw code because AI agents will handle it for you, you just focus on systems designs.

## Part B — Practical Task (Screenshot Required)

**CLI PROMPT -->** Create a GEMINI.md file in root and generate a 1-paragraph specification for an email validation function. Requirements of email are as follows: Must contain "@", Must contain a valid domain (e.g., .com, .org), Should return clear error messages

**The 1-paragraph specification generated by the CLI -->**The email validation function should accept a string as input and return a boolean indicating whether the email is valid, along with an array of error messages if validation fails. A valid email address must contain a "@" symbol and a domain with a valid top-level domain (e.g., ".com", ".org", ".net"). If the email is invalid, the function should provide clear and specific error messages, such as "Email must contain @" if the "@" symbol is missing, or "Invalid domain" if the domain is malformed or uses an unsupported TLD.

## **Part C — Multiple Choice Questions**

1. What is the main purpose of Spec-Driven Development?
  - A. Make coding faster
  - B. Clear requirements before coding begins**
  - C. Remove developers
  - D. Avoid documentation
  
1. What is the biggest mindset shift in AI-Driven Development?
  - A. Writing more code manually
  - B. Thinking in systems and clear instructions**
  - C. Memorizing more syntax
  - D. Working without any tools
  
  
  
  
  
  
  
  
  
  
3. Biggest failure of Vibe Coding?
  - A. AI stops responding
  - B. Architecture becomes hard to extend**
  - C. Code runs slow
  - D. Fewer comments written

4. Main advantage of using AI CLI agents (like Gemini CLI)?

A. They replace the developer completely

B. Handle repetitive tasks so dev focuses on design & problem-solving

C. Make coding faster but less reliable

D. Make coding optional

5. What defines an M-Shaped Developer?

A. Knows little about everything

B. Deep in only one field

C. Deep skills in multiple related domains

D. Works without AI tools