

## Part A — Theory (Short Questions)

### 1. Nine Pillars Understanding

**Q1:** Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?

**Answer:** AI agents boring aur repeat hone wale kaam bohot jaldi kar dete hain. Is se aapka time bach jata hai aur aap zyada important cheezon par focus kar sakti ho, jaise system ka design banana, planning karna aur sahi structure sochna. Ye skills hi aapko strong system architect banati hain.

**Q2:** Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer?

**Answer:** Nine Pillars developer ko sirf coding nahi, balki planning, architecture, tools ka sahi istemal, AI agents ke sath kaam, aur system level sochna sikhate hain. Jab developer multiple areas samajhne lagta hai to woh M-Shaped ban jata hai — yaani ek se zyada fields mein strong.

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

**Q1:** Why does Vibe Coding usually create problems after one week?

**Answer:** Vibe Coding mood ke hisaab se hota hai, bina plan ke. Pehle din sab theek lagta hai, lekin kuch dino baad code messy ho jata hai, cheezein samajh nahi aati, bugs aa jate hain aur project handle karna mushkil ho jata hai.

**Q2:** How would Specification-Driven Development prevent those problems?

**Answer:** Specification-Driven Development pehle hi clear plan bana deta hai. Jab requirements likhi hoti hain to code saaf, stable aur future mein change karna easy ho jata hai. Is wajah se Vibe Coding wali problems paida nahi hoti.

### 3. Architecture Thinking

Q1: How does architecture-first thinking change the role of a developer in AIDD?

Answer: Architecture-first thinking developer ko sirf code writer se hata kar system designer bana deta hai. Developer pehle pura system ka structure sochta hai, phir AI agents ko kaam deta hai. Is se developer ki value aur role dono grow karte hain.

Q2: Explain why developers must think in layers and systems instead of raw code.

Answer: Layers aur systems mein sochna se project organized rehta hai, manage karna easy hota hai, aur future updates smoothly ho jati hain. Raw code thinking se code jaldi bigar jata hai aur system toot-phoot jaisa ho jata hai. System thinking se sab controlled rehta hai.

## Part B — Practical Task



```
Gemini - email-validation-function
```

**> GEMINI**

Gemini 3 is now available.  
Join the waitlist at <https://goo.gle/enable-preview-features>

Tips for getting started:

1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

> Generate a one-paragraph specification for an email validation function. It should check "@" and a valid domain like .com or .org and return clear error messages.

▣ The email validation function should accept a string input representing an email address and determine its validity. It must perform essential checks, including verifying the presence of an "@" symbol and ensuring that the domain part contains a valid top-level domain (e.g., ".com", ".org", ".net"). Upon identifying an invalid email, the function should return a clear, user-friendly error message indicating the specific reason for the failure.

## 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

2. 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