

# Part A — Theory Questions

## Q1. Nine Pillars Understanding

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

**Answer:**

AI Development Agents repetitive aur basic setup kaam bohot fast aur accurate kar dete hain. Jab AI yeh routine tasks handle kar leta hai, developer ka time bach jata hai aur woh apna focus high-level design, architecture planning aur system thinking par laga sakta hai. Yeh approach developer ko coding ke chote kaam se utha kar system architect level ki soch develop karne mein help karti hai.

---

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

**Answer:**

Nine Pillars developer ko multiple areas mein strong banate hain — jaise AI coding, specifications, testing, cloud deployment, reusable skills, aur tool integration. In pillars ki wajah se developer sirf coding nahi, balkay architecture, infra, testing, aur domain-level tasks bhi samajhne lagta hai. Is tarah developer aik field nahi, balke kai fields mein depth hasil karta hai, jise M-Shaped developer kehte hain.

## Q2. Vibe Coding vs Specification-Driven Development

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

**Answer:**

Vibe Coding bina plan ya structure ke hota hai. Developer bas flow ke hisaab se code likhta jata hai. Ek week baad code messy ho jata hai, bugs trace nahi hotay, logic confuse ho jati hai, aur naye features add karna mushkil ho jata hai. Project ka control dheere-dheere khatam ho jata hai.

---

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

**Answer:**

Specification-Driven Development mein pehle clear specs likhi jati hain, jisme requirements, rules aur expected behavior define hota hai. AI aur developer dono ussi spec ko follow karte hain, isliye project organized, predictable aur easy to maintain rehta hai. Bugs kam hotay hain, aur long-term structure strong rehta hai.

---

## Q3. Architecture Thinking

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

**Answer:**

Architecture-first thinking developer ka role coder se system designer tak shift kar deta hai. AIDD mein AI coding ka bohot hissa karta hai, isliye developer ka main kaam architecture plan karna, systems ko structure dena aur high-level decisions lena hota hai. Developer execution se zyada thinking aur design par focus karta hai.

---

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

**Answer:**

Raw code sirf choti details hoti hain. Layers aur systems mein sochne se developer ko poore project ka big picture samajh aata hai. Yeh approach complexity ko control mein rakhti hai, changes easy banati hai, bugs identify karne mein help karti hai, aur architecture long-term stable rehta hai. System thinking ek scalable aur clean software banane ke liye zaroori hai.

## Part B — Practical Task

**Prompt:**

```
Using: 1 GEMINI.md file
> generate 1 paragraph specification for email validation . it must contain @ , a valid domain (e.g. , .com , .org) , it should returns clear error message.
~                                         no sandbox (see /docs)                               auto
```

**Answer:**

■ An effective email validation system must ensure that a given email address string conforms to standard formatting rules. The validation logic should first confirm the presence of exactly one "@" symbol, which acts as a separator between the local part and the domain. It must then verify that the domain part contains a valid top-level domain (TLD), such as ".com", ".org", or ".net", which is preceded by a domain name and a period. If the validation fails at any step, the system should return a clear and specific error message that guides the user, such as "Error: Invalid email format, missing '@' symbol" or "Error: Invalid or missing domain, please check the address."

## Part C — Multiple Choice Questions (Corrected Answers)

- What is the main purpose of Spec-Driven Development?  
✓ B. Clear requirements before coding begins
- What is the biggest mindset shift in AI-Driven Development?  
✓ B. Thinking in systems and clear instructions
- Biggest failure of Vibe Coding?  
✓ B. Architecture becomes hard to extend
- Main advantage of using AI CLI agents (like Gemini CLI)?  
✓ B. Handle repetitive tasks so dev focuses on design & problem-solving
- What defines an M-Shaped Developer?  
✓ C. Deep skills in multiple related domains