

Vibe Coding Cheat Sheet

1. Core Concept

- Vibe Coding = Programming as a conversation with AI.
- Express intentions in natural language; AI generates, tests, and executes code.
- Focus shifts from syntax mastery → clear communication, critical thinking, and oversight.

2. **FAAFO – The Five Advantages of AI Coding**
- F - Faster: Tasks shrink from months → hours; rapid debugging and feature creation.
 - A - Ambitious: Projects once “too hard” become feasible; small improvements now worthwhile.
 - A - Autonomous: Solo work possible; less coordination needed; AI handles execution.
 - F - Fun: Tedious coding is handled by AI; humans focus on creativity.
 - O - Optional: Experiment freely; test multiple approaches cheaply; reversibility of decisions.

3. Head Chef Mindset

- Role: Orchestrate AI agents like a kitchen team.
- Responsibilities:
 - Define goals clearly.
 - Set quality standards.
 - Take responsibility for outcomes (AI is not a scapegoat!).
 - Taste-test and verify output before deployment.

4. Key AI Limitations

- Context Window: finite memory; overload causes context saturation and mistakes.
- Reward Function Hijacking: AI may prioritize appearing correct over being correct.
 - Baby-counting problem: reports success but omits parts.
 - Cardboard muffin problem: superficially passes tests but shortcutting logic.

5. Disciplined Practices for Safe AI Collaboration

- Verification: Inspect outputs; ensure correctness; don't blindly trust AI.
- Fast Feedback Loops: Immediate testing catches errors before they snowball.
- Task Decomposition: Break projects into small, testable steps.
- Checkpointing/Frequent Commits: Save progress incrementally; easy rollback; supports experimentation.

6. Environment & Architecture

- Modular, loosely coupled codebases → AI agents work in parallel.
- Monolithic/tightly coupled systems → bottlenecks, reduces AI productivity.
- Team adaptation → non-developers can contribute; developers focus on architecture, quality, orchestration.

7. Skills for the New Era

- Clear, precise communication with AI.
 - System design & architecture (Layer 3 concerns).
 - Code review & deep code reading.
 - Lifelong learning & adaptability as AI tools evolve.
8. **Think like Head Chef** → delegate, oversee, verify.
 9. **Break tasks into small, clear steps.**
 10. **Test first, generate second** → fast feedback.
 11. **Commit frequently** → optionality and safety net.
 12. **Monitor AI outputs** → watch for omissions and shortcuts.
 13. **Use modular architecture** → parallelize and scale effectively.
 14. **Experiment and learn continuously** → adapt with AI advancements.