Competency Area	Skill/Attribute	Junior Developer (L1)	Mid-Level Developer (L2)	Senior Developer (L3)
Technical Proficiency	Programming Fundamentals	Understands basic syntax, data types, control flow.	Writes clean, efficient code; good grasp of OOP/FP concepts.	Masters advanced concepts; understands language nuances & trade-offs.
	Data Structures & Algorithms	Knows common structures (lists, maps); basic sorting/searching.	Selects appropriate DS&A for problems.	Designs custom DS or complex algorithms; optimizes for performance.
	System Design	Understands component interactions within a single service.	Contributes to feature design; understands API design principles.	Designs multi-service features/systems; considers scalability, reliability.
	Databases	Basic CRUD operations; simple SQL queries.	Writes complex queries; understands indexing; basic NoSQL concepts.	Designs database schemas; optimizes queries; selects appropriate DB type.
	Testing	Writes basic unit tests.	Writes comprehensive unit/integration tests; understands mocking.	Designs testing strategies; familiar with TDD/BDD; performance testing.
	Development Tools	Uses IDE, basic Git commands (commit, push, pull).	Proficient with Git (branching, merging); uses debugger effectively.	Masters build tools, CI/CD pipelines; proficient with containerization (Docker).
Problem Solving	Analysis & Decomposition	Can break down simple problems with guidance.	Independently breaks down features into tasks.	Decomposes complex problems into manageable sub-problems.
	Debugging & Troubleshooting	Can debug simple issues using logs/debugger.	Systematically finds root causes of bugs in own/others' code.	Troubleshoots complex system-level issues; anticipates problems.
	Solution Design & Creativity	Implements straightforward solutions based on specs.	Proposes alternative solutions; considers trade-offs.	Designs elegant, robust, and scalable solutions; innovates.
Collaboration & Comms	Code Reviews	Participates constructively in reviews (receives feedback).	Provides thoughtful code review feedback.	Leads code reviews; establishes best practices.
	Documentation	Writes clear comments in code.	Writes basic technical documentation (READMEs, API usage).	Writes comprehensive design docs; documents architectural decisions.
	Technical Communication	Explains basic technical concepts clearly.	Explains technical decisions & trade-offs to peers/team lead.	Communicates complex ideas effectively to technical & non-technical audiences.
	Teamwork	Collaborates effectively on assigned tasks.	Works well within the team; shares knowledge proactively.	Mentors junior developers; helps drive team alignment & goals.
SDLC & Best Practices	Agile/Scrum	Understands basic ceremonies & principles.	Actively participates & contributes to Agile processes.	Helps improve team processes; understands Agile values deeply.
	Code Quality & Maintainability	Writes code that follows basic style guides.	Writes clean, readable, maintainable code; applies SOLID principles.	Champions code quality; refactors effectively; understands design patterns.
	Security Awareness	Understands basic security concepts (e.g., injection).	Writes code defensively; aware of common vulnerabilities (OWASP Top 10).	Proactively identifies & addresses security risks in design/code.