SDET Interview Answers (Concise)

--- Manual Testing ---

1. Software Testing: Process of evaluating software to find defects and ensure quality.

2. Verification vs Validation: Verification = "Are we building right?" Validation = "Are we building right product?"

3. SDLC: Phases of software development; testing aligns with each phase for quality checks.

4. STLC: Testing life cycle steps – requirement analysis to closure.

5. Levels of Testing: Unit, Integration, System, UAT.

6. Functional vs Non-Functional: Functional tests business logic; non-functional tests performance/security.

7. Regression vs Retesting: Regression = after changes; Retesting = verifying a fixed defect.

8. Positive vs Negative Testing: Positive = valid data; Negative = invalid data.

9. Severity vs Priority: Severity = impact; Priority = urgency.

10. Test Case: Steps, preconditions, expected results.

11. Test Plan: Scope, approach, resources, schedule.

12. Types of Testing: Smoke, sanity, exploratory, ad-hoc.

13. BVA: Testing boundaries.

14. EP: Grouping data into partitions.

15. Static vs Dynamic: Static = reviews; Dynamic = executing software.

16. Defect Life Cycle: New → Open → Fixed → Retest → Closed.

17. RTM: Mapping requirements to test cases.

18. Prioritizing: Based on risk, business impact.

19. Risk-Based Testing: Focus on high-risk areas.

20. Entry/Exit Criteria: Conditions to start/end testing.

21. Bug in Production: Report, analyze, fix quickly.

22. Missing Requirements: Clarify with BA/PO.

23. Developer Disagrees: Provide evidence.

24. Bug Reporting: Clear steps, logs, screenshots.

25. Requirement Review: Validate completeness and clarity.

26. Limited Time: Prioritize critical tests.

27. Exploratory Testing: Explore without scripts.

28. 100% Coverage: RTM, risk analysis.

29. No Requirements: Explore UI, discuss with stakeholders.

30. Challenges: Unclear requirements, environment issues.

--- Java Answers ---

(Answers concise due to size limits)

1. Features: OOP, platform independent, robust.

2. JDK/JRE/JVM: JVM runs code, JRE runtime env, JDK development kit.

3. OOP Principles: Encapsulation, Abstraction, Inheritance, Polymorphism.

4. == vs equals(): == compares references; equals() compares values.

5. ArrayList vs LinkedList: ArrayList faster for retrieval.

6. HashMap vs LinkedHashMap vs TreeMap: Unordered, ordered, sorted.

(… similar concise answers continue …)

Due to space constraints in PDF, recommend splitting categories in multiple PDFs for full detail.