**1) What is Manual Testing?**

* Manual Testing is the process of manually executing test cases without using automation tools.
* It ensures the application behaves as expected by simulating end-user actions.
* It helps find UI/UX issues, logical errors, and unexpected behavior.

**2) What are the different types of manual testing?**

* **Smoke Testing**
* **Sanity Testing**
* **Functional Testing**
* **Integration Testing**
* **System Testing**
* **User Acceptance Testing (UAT)**
* **Regression Testing**
* **Exploratory Testing**
* **Ad-hoc Testing**
* **Compatibility Testing**

**3) What is the difference between functional and non-functional testing?**

* **Functional Testing**: Validates what the system does; checks business logic and features.
* **Non-functional Testing**: Validates how the system performs; checks performance, usability, scalability, etc.

**4) What are test cases, and why are they important?**

* A **test case** is a documented set of conditions and steps to validate a specific functionality.
* Important for ensuring coverage, repeatability, tracking defects, and improving test efficiency.

**5) What is the Software Testing Life Cycle (STLC)?**

* STLC is the step-by-step process followed in testing:
  1. Requirement Analysis
  2. Test Planning
  3. Test Case Design
  4. Test Environment Setup
  5. Test Execution
  6. Test Closure (Reporting & Retrospective)

**6) What is the difference between verification and validation?**

* **Verification**: Are we building the product right? (Static testing: reviews, walkthroughs)
* **Validation**: Are we building the right product? (Dynamic testing: actual test execution)

**7) What are the key components of a test plan?**

* Test Plan ID
* Objectives & Scope
* Test Strategy
* Resources & Roles
* Entry & Exit Criteria
* Test Environment
* Deliverables
* Schedule
* Risks & Mitigation

**8) What is exploratory testing, and when is it useful?**

* It is simultaneous learning, test design, and execution without predefined test cases.
* Useful when requirements are unclear, during early stages, or for identifying hidden defects.

**9) What is the purpose of smoke testing?**

* To check whether the critical functionalities are working after a new build.
* Acts as a basic stability check before proceeding with deeper testing.

**10) What is regression testing?**

* Re-testing existing functionalities to ensure that new changes have not broken them.
* Done after bug fixes, enhancements, or integration of new features.

**11) What is unit testing, and how is it performed manually?**

* Unit Testing checks individual components or functions of code.
* Typically done by developers using code-level inputs and expected outputs.
* In manual unit testing (rare), test data is given manually, and outputs are verified via print/debug statements or UI logs.

**12) What is integration testing?**

* It verifies the interaction between integrated units/modules.
* Ensures data flows correctly between modules and they work together.
* Done after unit testing, either incrementally or all modules at once (Big Bang).

**13) What is system testing?**

* It validates the entire application as a whole against the requirements.
* Includes functional and non-functional testing.
* Performed in an environment that closely resembles production.

**14) What is acceptance testing?**

* Final level of testing to verify if the system meets business needs.
* Usually done by the client or end-user.
* Types: Alpha testing (by in-house users), Beta testing (by external users).

**15) What is sanity testing?**

* A quick, focused test to check if a specific functionality works after minor changes.
* Performed before deep testing like regression.
* Example: After a small bug fix, testing only that feature.

**16) What is a test scenario?**

* A high-level idea of what to test.
* Describes what needs to be tested, not detailed steps.
* Example: “Verify login with valid credentials.”

**17) What is a test script?**

* A set of detailed instructions to execute a test.
* Can be manual (step-by-step document) or automated (code).
* Includes steps, data, and expected results.

**18) What is a test log?**

* A document that records the execution details of test cases.
* Captures info like start/end time, test steps executed, pass/fail status, and issues faced.
* Helps in debugging and audit trails.

**19) What is a test report?**

* A summary of all testing activities and results.
* Includes number of test cases passed/failed, defect details, environment, and overall quality status.
* Shared with stakeholders to make release decisions.

**20) What is defect tracking?**

* Process of identifying, recording, managing, and closing defects.
* Tools like JIRA, Bugzilla, and Mantis are used.
* Tracks defect life cycle: New → Assigned → Open → Fixed → Retested → Closed/Reopen.

**21) What is the difference between severity and priority?**

* **Severity**: Impact of the defect on the application (set by tester).
  + Example: System crash = High Severity.
* **Priority**: Urgency to fix the defect (set by developer/manager).
  + Example: A typo on the homepage = High Priority, Low Severity.

**22) What are different test techniques in manual testing?**

* **Equivalence Partitioning**
* **Boundary Value Analysis (BVA)**
* **Decision Table Testing**
* **State Transition Testing**
* **Error Guessing**
* **Use Case Testing**
* These techniques help design effective test cases.

**23) What is the difference between black-box and white-box testing?**

* **Black-box Testing**: Tester tests without knowing internal code. Focus on input/output.
* **White-box Testing**: Tester has access to source code; tests logic, loops, paths. Usually done by developers.

**24) What is positive and negative testing?**

* **Positive Testing**: Test with valid inputs to check expected behavior.
* **Negative Testing**: Test with invalid inputs to ensure app handles errors properly.

**25) What is usability testing?**

* Tests how user-friendly and intuitive the application is.
* Focuses on layout, navigation, and ease of use.
* Usually done with feedback from end-users or usability experts.

**26) What is the difference between test case and test scenario?**

* **Test Case**: Detailed steps, input, expected result for one condition.
* **Test Scenario**: High-level idea of what to test (broader).
  + Example:
    - Scenario: Test login functionality.
    - Case: Verify login with valid email and password.

**27) What is compatibility testing?**

* Tests application behavior across different:
  + Browsers (Chrome, Firefox, etc.)
  + Operating Systems (Windows, macOS, Linux)
  + Devices (Mobile, Tablet, Desktop)
* Ensures consistent performance and UI.

**28) What are different levels of testing?**

1. **Unit Testing**
2. **Integration Testing**
3. **System Testing**
4. **Acceptance Testing**

**29) What is ad-hoc testing?**

* Informal, unstructured testing without test cases.
* Based on tester’s understanding and intuition.
* Useful for quickly finding unexpected issues.

**30) What is end-to-end testing?**

* Validates the complete flow of the application from start to finish.
* Covers integration of all components and data flow across systems.
* **Example:** Place an order in an e-commerce app → confirm email → payment → delivery.

**31) What is exploratory testing?**

* Simultaneous learning, test design, and test execution.
* No predefined test cases; tester explores the app dynamically.
* Best used when documentation is limited or for finding hidden bugs.

**32) What is retesting?**

* Testing a specific defect again after it has been fixed.
* Verifies the bug is actually resolved.
* Done with the same input and environment where the bug occurred.

**33) What is the difference between retesting and regression testing?**

* **Retesting**: Re-validates fixed defects using the same data.
* **Regression Testing**: Ensures new changes haven’t broken existing functionality.
* Retesting is bug-specific; regression is system-wide.

**34) What is recovery testing?**

* Tests the system’s ability to recover from crashes, hardware failures, or network issues.
* Example: Simulating power failure to check auto-save or restart recovery.

**35) What is maintainability testing?**

* Checks how easily the application can be updated, modified, or enhanced.
* Focus on code structure, modularity, and documentation.

**36) What is portability testing?**

* Verifies whether the software can run on different environments or platforms.
* Example: App working across various OS, browsers, or mobile devices.

**37) What is localization testing?**

* Ensures the software behaves correctly in a specific region or language.
* Focuses on UI, date/time formats, currency, and cultural appropriateness.

**38) What is globalization testing?**

* Tests if the application supports multiple languages, regions, and cultural formats.
* Ensures product is ready for international markets without code changes.

**39) What is risk-based testing?**

* Prioritizing testing of features based on risk of failure and business impact.
* High-risk areas (e.g., payment module) are tested first and more thoroughly.

**40) What is the difference between alpha and beta testing?**

* **Alpha Testing**: Done in-house by internal testers before release.
* **Beta Testing**: Done by real users in a live environment before final release.
* Alpha is controlled; beta is real-world feedback.

**41) What is a bug life cycle?**

* Also called **defect life cycle**, it defines the stages a bug goes through from identification to closure:
  1. New
  2. Assigned
  3. Open
  4. Fixed
  5. Retest
  6. Verified
  7. Closed or Reopened (if not fixed)

**42) What is a defect report?**

* A document used to report and track defects found during testing.
* Includes defect ID, title, steps to reproduce, severity, priority, status, environment, and screenshots.

**43) What is an issue tracker?**

* A tool used to log, manage, and track defects or issues in the application.
* Examples: **JIRA**, **Bugzilla**, **Mantis**, **Redmine**.
* Helps in collaboration between developers, testers, and stakeholders.

**44) What are test deliverables?**

* Artifacts created during and after the testing process:
  + Test Plan
  + Test Cases
  + Test Scenarios
  + Test Data
  + Defect Reports
  + Test Summary/Closure Report
  + Traceability Matrix

**45) What are entry and exit criteria in testing?**

* **Entry Criteria**: Conditions to be met before testing begins (e.g., test environment setup, approved test cases).
* **Exit Criteria**: Conditions to be met before testing ends (e.g., 95% test cases passed, all critical bugs fixed).

**46) What is load testing?**

* Tests system performance under expected user load.
* **Example:** Checking how a website behaves with 1,000 simultaneous users.

**47) What is stress testing?**

* Tests the system beyond its normal limits to see how it behaves under extreme conditions.
* Identifies the breaking point or how system fails gracefully.

**48) What is volume testing?**

* Tests system performance by feeding it a large volume of data.
* **Example:** Uploading 10GB of data to see database performance.

**49) What is soak testing?**

* Also called **endurance testing**.
* Checks system stability over a long period of continuous usage.
* **Example:** Running an app continuously for 48 hours to check memory leaks or crashes.

**50) What is failover testing?**

* Tests how the system behaves during a failure and whether it can recover.
* **Example:** If one server goes down, does the app switch to a backup server smoothly?

**51) What is security testing?**

* Validates that the application is secure from external threats.
* **Focus areas:** data protection, authentication, authorization, session management, SQL injection, XSS, etc.
* **Goal:** Ensure confidentiality, integrity, and availability.

**52) What is compliance testing?**

* Ensures that the application meets industry regulations, standards, or internal policies. Compliance testing is a process that verifies whether a software application or system meets a specific set of regulations, standards, or internal policies it is required to follow.
* **Example:** GDPR, HIPAA, PCI-DSS compliance checks.
* **GDPR (General Data Protection Regulation):** Testing that the application correctly handles the data privacy and rights of EU citizens.
* **HIPAA (Health Insurance Portability and Accountability Act):** Ensuring software that manages U.S. patient health information meets strict security and privacy rules.
* **PCI-DSS (Payment Card Industry Data Security Standard):** Verifying that an application securely handles credit card data to prevent fraud.
* The primary goal is to ensure the product is legally and contractually sound, avoiding potential fines, security breaches, and loss of reputation. Also called **Conformance Testing**.

**53) What is a test matrix?**

* A table showing the relationship between requirements and test coverage.
* Helps identify which requirements have been tested and which haven't.
* Also called **Requirement Traceability Matrix (RTM)**.

**54) What is the role of a manual tester?**

* Analyze requirements and prepare test scenarios/test cases.
* Execute test cases manually and report defects.
* Collaborate with developers and business analysts for defect resolution and validation.
* Ensure the product meets functional and business requirements.

**55) What is acceptance criteria?**

* Predefined conditions that a product must meet to be accepted by the client or user.
* Derived from business requirements.
* Example: "User must be able to log in with email and password."

**56) What is test data?**

* Input values used to execute test cases.
* Types: valid, invalid, boundary, large data sets.
* Test data can be manually created or fetched from production-like environments.

**57) What is the role of environment in testing?**

* A controlled setup where the application is tested (hardware, software, network, and configurations).
* Types: Development, Testing/QA, Staging, Production.
* Affects test results and defect reproducibility.

**58) What is defect masking?**

* When one defect hides the presence of another defect.
* Occurs when fixing or identifying a bug prevents other bugs from being detected.
* **Example:** A login crash prevents testing of profile features.

**59) What is defect leakage?**

* When a defect is missed during testing and found in production by the end-user.
* Indicates gaps in test coverage or test effectiveness.
* Metric to evaluate test team performance.

**60) What is test coverage?**

* Measures the amount of testing performed on the application.
* Can refer to requirements, code, functionalities, or paths tested.
* High coverage = reduced risk of undetected bugs.

**61) What are test metrics?**

* Quantitative measures used to track and assess the efficiency and effectiveness of the testing process.
* Examples:
  + Test case execution rate
  + Defect density
  + Defect leakage rate
  + Test coverage
  + Test pass/fail percentage

**62) What is defect clustering?**

* A principle from **Pareto's Law**: 80% of defects are found in 20% of the modules.
* Helps prioritize testing on high-risk, defect-prone areas.
* Suggests focused testing on complex or frequently changed modules.

**63) What is shift-left testing?**

* An approach where testing activities start early in the software development lifecycle.
* Helps in early defect detection and reduces rework cost.
* Involves activities like requirement reviews, unit tests, and static code analysis.

**64) What is exploratory testing?**

* Testing without predefined test cases; the tester learns, designs, and executes tests simultaneously.
* Based on the tester’s intuition, experience, and product knowledge.
* Ideal for discovering unexpected bugs or during early test cycles.

**65) What is session-based testing?**

* A type of exploratory testing conducted in time-boxed sessions.
* Each session has a clear goal, charter, and documented observations.
* Combines structure with flexibility for better traceability and accountability.

**66) What is the difference between bug and defect?**

* **Defect**: A variance from the expected result found during testing.
* **Bug**: Informal term for a defect, commonly used in development and testing.
* Technically, both refer to issues in the software, but "defect" is more formal.

**67) What is test case design?**

* The process of creating effective and efficient test cases based on requirements.
* Involves techniques like:
  + Boundary Value Analysis
  + Equivalence Partitioning
  + Decision Table
  + State Transition
  + Use Case Testing

**68) What are test artifacts?**

* Documents and tools used throughout the testing lifecycle.
* Examples:
  + Test Plan
  + Test Cases
  + Test Scenarios
  + Traceability Matrix
  + Defect Reports
  + Test Summary Reports

**69) What are common challenges in manual testing?**

* Time-consuming for large applications
* Prone to human error
* Limited regression coverage
* Difficult to simulate complex scenarios
* Repetitive and lacks reusability
* Dependency on experienced testers

**70) What is the role of a test manager?**

* Responsible for planning, monitoring, and controlling all testing activities.
* Key duties:
  + Define test strategy and process
  + Allocate tasks and manage resources
  + Track progress and ensure quality metrics
  + Report test status to stakeholders
  + Risk management and team mentoring

**71) What is a testing checklist?**

* A predefined list of activities or conditions to verify during testing.
* Helps ensure important aspects are not missed.
* Commonly used in reviews, regression, or release checklists.

**72) What is parallel testing?**

* Testing multiple versions or modules of an application at the same time.
* Often used to compare old and new systems.
* Ensures consistency and correctness after migration or updates.

**73) What is accessibility testing?**

* Ensures the application is usable by people with disabilities (e.g., visually or hearing impaired).
* Follows standards like **WCAG** (Web Content Accessibility Guidelines).
* Tests screen readers, keyboard navigation, contrast, etc.

**74) What is A/B testing?**

* A technique where two versions (A and B) are shown to users to compare performance.
* Used in UI/UX or feature optimization.
* Helps identify which version performs better based on user behavior.

**75) What is interface testing?**

* Verifies interactions between different software components or systems.
* Checks data exchange, request/response formats, and error handling.
* Example: Testing between front-end and API or between two services.

**76) What is mutation testing?**

* A white-box testing technique where small changes (mutations) are introduced in the code.
* Checks if existing test cases can detect the changes.
* Helps improve test case effectiveness.

**77) What is fault tolerance testing?**

* Ensures the system continues to function properly in the presence of faults or errors.
* Example: If one component fails, the app should switch to a backup without crashing.

**78) What is disaster recovery testing?**

* Validates the system’s ability to recover from catastrophic failures (e.g., server crash, natural disaster).
* Ensures data recovery, system restoration, and business continuity plans work.

**80) What is the difference between static and dynamic testing?**

* **Static Testing**: Performed without executing the code (e.g., reviews, walkthroughs).
* **Dynamic Testing**: Involves actual code execution to find defects during runtime (e.g., functional testing).
* Static = preventive; Dynamic = detective.

**81) What is Agile Testing?**

* Testing that aligns with Agile development practices.
* Involves continuous testing in short iterations (sprints).
* Focuses on collaboration, early feedback, and flexibility.
* Testers work closely with developers and product owners.

**82) What is Test-Driven Development (TDD)?**

* A development approach where tests are written **before** the code.
* Flow:
  1. Write a test
  2. Run it (it fails)
  3. Write code to pass the test
  4. Refactor
* Promotes clean, testable, and reliable code.

**83) What is Behavior-Driven Development (BDD)?**

* Extension of TDD focusing on **behavior** of the application from the user's perspective.
* Uses natural language constructs like “Given-When-Then”.
* Tools: Cucumber, SpecFlow.
* Improves collaboration among developers, testers, and non-technical stakeholders.

**84) What is Defect Severity Classification?**

* Categorizes how **serious** a defect is in terms of system impact:
  + **Critical** – System crash, data loss
  + **High** – Major functionality broken
  + **Medium** – Some issues, but workarounds exist
  + **Low** – Minor UI or cosmetic issues
* Helps prioritize fixes effectively.

**85) What is Acceptance Test-Driven Development (ATDD)?**

* Similar to TDD but driven by **acceptance criteria** written before development.
* Involves collaboration between developers, testers, and business stakeholders.
* Ensures software meets business expectations through shared understanding.

**86) What is ISO/IEC 25010?**

* A quality model standard for software product evaluation.
* Defines **8 quality characteristics**:
  + Functionality, Reliability, Usability, Efficiency, Maintainability, Portability, Compatibility, Security
* Guides quality requirements and testing practices.

**87) What is a Bug Tracking Tool?**

* A software used to log, track, and manage defects throughout the testing lifecycle.
* Features: Assign bugs, prioritize, track status, generate reports.
* Examples: **JIRA**, Bugzilla, Mantis, Redmine.

**88) What is Test Execution Cycle?**

* The phase in testing where test cases are executed against the application.
* Steps include:
  1. Execute test cases
  2. Record results
  3. Log defects (if any)
  4. Re-test and close
* Continues until test exit criteria are met.