**Part 1: Fundamentals of Manual Testing (1–20)**

**1. What is manual testing?**  
Manual testing is the process of executing test cases without using automation tools.

**2. Why is manual testing important?**  
It helps find bugs in scenarios where automation is impractical or expensive.

**3. What is software testing?**  
The process of evaluating software to ensure it meets requirements and works as expected.

**4. What is the objective of testing?**  
To detect defects, ensure quality, and verify functionality.

**5. What are the types of software testing?**  
Functional testing, non-functional testing, and maintenance testing.

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

* **Verification:** Ensuring the product is built correctly.
* **Validation:** Ensuring the right product is built.

**7. What is a defect?**  
A flaw in the software that causes incorrect or unexpected behavior.

**8. What is a bug report?**  
A document describing a defect, steps to reproduce it, and expected results.

**9. What is a test case?**  
A set of conditions and steps to verify a specific feature.

**10. What is a test scenario?**  
A high-level description of what needs to be tested.

**11. What is a test plan?**  
A document that describes the testing approach, scope, and objectives.

**12. What is a test strategy?**  
A high-level document describing the testing principles and objectives for a project.

**13. What is exploratory testing?**  
An informal testing approach where testers explore the application without predefined test cases.

**14. What is ad-hoc testing?**  
Unstructured testing without documentation, performed to quickly find defects.

**15. What is smoke testing?**  
A basic check to ensure the main functions of the application work.

**16. What is sanity testing?**  
Testing to verify that a small change hasn’t broken existing functionality.

**17. What is regression testing?**  
Re-testing to ensure new changes haven’t introduced defects in existing functionality.

**18. What is functional testing?**  
Testing features against functional requirements.

**19. What is non-functional testing?**  
Testing aspects like performance, usability, and security.

**20. What is usability testing?**  
Checking how user-friendly the application is.

**Part 2: Testing Process & Documentation (21–40)**

**21. What is the Software Development Life Cycle (SDLC)?**  
A framework defining stages from requirement gathering to maintenance.

**22. What is the Software Testing Life Cycle (STLC)?**  
A process defining the stages of testing from planning to closure.

**23. What are the phases of STLC?**  
Requirement Analysis → Test Planning → Test Case Design → Test Execution → Defect Reporting → Test Closure.

**24. What is a requirement traceability matrix (RTM)?**  
A document mapping requirements to test cases.

**25. What is a test environment?**  
A setup of hardware, software, and configurations used for testing.

**26. What is the difference between test data and production data?**

* **Test data:** Used for testing.
* **Production data:** Used in the live environment.

**27. What is boundary value analysis (BVA)?**  
Testing at the edge values of input ranges.

**28. What is equivalence partitioning?**  
Dividing inputs into valid and invalid partitions for testing.

**29. What is positive testing?**  
Testing with valid inputs to verify expected results.

**30. What is negative testing?**  
Testing with invalid inputs to check error handling.

**31. What is acceptance testing?**  
Testing to ensure the product meets business requirements.

**32. What is alpha testing?**  
Testing performed by internal teams before release.

**33. What is beta testing?**  
Testing performed by external users before release.

**34. What is compatibility testing?**  
Testing the software on different browsers, devices, or OS.

**35. What is cross-browser testing?**  
Checking that the application works across different browsers.

**36. What is performance testing?**  
Measuring how the system performs under different loads.

**37. What is load testing?**  
Testing how the application performs under expected load.

**38. What is stress testing?**  
Testing the system beyond expected limits.

**39. What is recovery testing?**  
Testing the system’s ability to recover from crashes or failures.

**40. What is installation testing?**  
Verifying that the software installs and uninstalls correctly.

**Part 3: Defects & Quality (41–60)**

**41. What is defect severity?**  
How serious the defect is in terms of impact.

**42. What is defect priority?**  
How quickly the defect should be fixed.

**43. What is the difference between severity and priority?**

* **Severity:** Impact on system.
* **Priority:** Urgency of fix.

**44. What is a defect life cycle?**  
The stages a defect goes through from detection to closure.

**45. What is the difference between defect, error, and failure?**

* **Error:** Mistake in code.
* **Defect:** Found in testing.
* **Failure:** Defect found in production.

**46. What is root cause analysis?**  
Identifying the main reason for a defect.

**47. What is quality assurance (QA)?**  
Ensuring processes are in place to deliver quality products.

**48. What is quality control (QC)?**  
Checking the actual product for defects.

**49. What is the difference between QA and QC?**  
QA is process-oriented, QC is product-oriented.

**50. What is test coverage?**  
The percentage of requirements or code covered by testing.

**51. What is 100% test coverage?**  
When all planned requirements or code paths are tested.

**52. What is defect leakage?**  
When defects are missed in testing and found later.

**53. What is defect density?**  
Number of defects per size of the software module.

**54. What is the cost of quality?**  
Total cost to ensure the product meets quality standards.

**55. What is software reliability?**  
The probability that software will work without failure for a given period.

**56. What is maintainability in software?**  
Ease of modifying the system after release.

**57. What is reusability in testing?**  
Ability to use the same test cases in different projects.

**58. What is interoperability testing?**  
Testing how well the software interacts with other systems.

**59. What is localization testing?**  
Checking language and region-specific features.

**60. What is globalization testing?**  
Ensuring the software works worldwide without modifications.

**Part 4: Testing Techniques & Best Practices (61–100)**

**61. What is static testing?**  
Reviewing code, documents, or requirements without execution.

**62. What is dynamic testing?**  
Executing the application to find defects.

**63. What is black-box testing?**  
Testing without knowing the internal code.

**64. What is white-box testing?**  
Testing with full knowledge of the code.

**65. What is grey-box testing?**  
Testing with partial knowledge of the code.

**66. What is unit testing?**  
Testing individual components in isolation.

**67. What is integration testing?**  
Testing combined modules to ensure they work together.

**68. What is system testing?**  
Testing the entire system as a whole.

**69. What is end-to-end testing?**  
Testing the complete workflow of the application.

**70. What is monkey testing?**  
Random testing without predefined inputs.

**71. What is pair testing?**  
Two testers work together on the same functionality.

**72. What is session-based testing?**  
Time-boxed exploratory testing sessions.

**73. What is checklist-based testing?**  
Testing based on a predefined checklist.

**74. What is domain testing?**  
Testing values within a particular domain range.

**75. What is use case testing?**  
Testing based on documented use cases.

**76. What is decision table testing?**  
Testing using conditions and actions in a tabular form.

**77. What is state transition testing?**  
Testing changes in application state based on events.

**78. What is error guessing?**  
Using tester’s experience to guess where defects may be.

**79. What is a test harness?**  
A setup of tools and data to execute tests.

**80. What is defect clustering?**  
Defects tend to cluster in certain modules.

**81. What is pesticide paradox?**  
If the same tests are run repeatedly, they stop finding new defects.

**82. How to overcome pesticide paradox?**  
Regularly update and improve test cases.

**83. What is exploratory vs ad-hoc testing?**

* **Exploratory:** Structured but not scripted.
* **Ad-hoc:** Completely unstructured.

**84. What is a test oracle?**  
The source of truth to verify expected results.

**85. What is acceptance criteria?**  
Conditions that must be met for a feature to be accepted.

**86. What is shift-left testing?**  
Starting testing earlier in the development process.

**87. What is shift-right testing?**  
Testing after deployment, in production.

**88. What is continuous testing?**  
Running tests as part of the CI/CD pipeline.

**89. What is a bug triage?**  
Prioritizing and assigning bugs for fixing.

**90. What is defect prevention?**  
Taking measures to avoid introducing defects.

**91. What is software configuration management in testing?**  
Tracking and controlling software changes during testing.

**92. What is a test summary report?**  
A document summarizing testing activities and results.

**93. What is a test closure report?**  
A final document confirming testing is complete.

**94. What is a walk-through in testing?**  
An informal review of documents or code.

**95. What is an inspection in testing?**  
A formal, detailed review process.

**96. What is a peer review in testing?**  
Reviewing work products by colleagues.

**97. What is the difference between static and dynamic testing?**

* **Static:** No execution.
* **Dynamic:** Requires execution.

**98. What is a risk-based testing approach?**  
Focusing on high-risk areas first.

**99. What is a test readiness review?**  
A meeting to ensure the test environment and data are ready.

**100. What is the role of a manual tester?**  
Designing, executing, and documenting tests to ensure software quality.

**Part 1: Advanced Concepts & Best Practices (101–130)**

**101. What is exploratory testing’s biggest advantage?**  
It uncovers unexpected defects quickly without heavy documentation.

**102. What is the difference between QA and Testing?**

* **QA:** Process-oriented, ensures quality in the process.
* **Testing:** Product-oriented, finds defects in the product.

**103. What is the difference between manual and automated testing?**

* **Manual:** Human execution of tests.
* **Automated:** Scripts and tools execute tests automatically.

**104. Why do some companies still prefer manual testing?**  
Because exploratory, usability, and visual checks are best done manually.

**105. What is risk-based testing?**  
Prioritizing testing efforts on high-risk areas.

**106. What is a showstopper bug?**  
A defect that blocks further testing or product release.

**107. What is test data preparation?**  
Creating the required input data before test execution.

**108. What is a smoke vs sanity testing difference?**

* **Smoke:** Checks if the main build is stable.
* **Sanity:** Checks small changes after a fix.

**109. What is alpha vs beta testing difference?**

* **Alpha:** Internal testers before release.
* **Beta:** Real users before final release.

**110. What is a heuristic in testing?**  
A problem-solving approach using experience-based techniques.

**111. What is equivalence class testing?**  
Grouping inputs that should be treated the same.

**112. What is pairwise testing?**  
Testing combinations of input parameters in pairs to reduce effort.

**113. What is test prioritization?**  
Running the most important tests first when time is limited.

**114. What is defect rejection ratio?**  
The percentage of defects reported that are rejected.

**115. What is defect reproducibility?**  
The ability to reproduce a defect consistently.

**116. What is defect slippage?**  
When defects are found in production after passing testing.

**117. What is defect turnaround time?**  
The time taken from reporting to fixing a defect.

**118. What is exploratory testing charter?**  
A statement defining scope, objectives, and time for exploratory testing.

**119. What is domain knowledge in testing?**  
Understanding the business area the application serves.

**120. Why is domain knowledge important for testers?**  
It improves the ability to identify business-critical defects.

**121. What is API testing in manual QA?**  
Testing APIs using tools like Postman without automation frameworks.

**122. What is mobile application manual testing?**  
Testing mobile apps on real devices or simulators.

**123. What is desktop application manual testing?**  
Testing software installed locally on PCs or laptops.

**124. What is localization vs globalization testing difference?**

* **Localization:** Adapting to specific regions.
* **Globalization:** Ensuring the app works worldwide.

**125. What is test optimization?**  
Reducing test cases while maintaining coverage.

**126. What is testing debt?**  
Unfinished or skipped tests that need to be addressed later.

**127. What is shift-left testing?**  
Starting testing earlier in the development cycle.

**128. What is shift-right testing?**  
Testing in production or post-deployment.

**129. What is the main role of a test lead?**  
To plan, coordinate, and ensure the quality of the testing process.

**130. What is the main role of a test manager?**  
To oversee overall testing strategy, resources, and quality goals.

**Part 2: Real-World Scenarios (131–160)**

**131. What will you do if you find a critical bug just before release?**  
Immediately report it to stakeholders and assess the impact before release.

**132. How do you test without requirements?**  
Use exploratory testing, talk to stakeholders, and review similar systems.

**133. How do you handle vague requirements?**  
Seek clarification, write assumptions, and confirm with the client.

**134. What do you do if developers disagree with your bug?**  
Reproduce the defect with proof, logs, and steps.

**135. How do you test under tight deadlines?**  
Prioritize high-risk and critical test cases first.

**136. How do you test a login page?**  
Check valid/invalid credentials, security, UI, and performance.

**137. How do you test a search feature?**  
Check results accuracy, filters, sorting, and edge cases.

**138. How do you test a payment gateway?**  
Check success, failure, network issues, and security.

**139. How do you test without access to the backend?**  
Focus on front-end validations, API calls, and UI behavior.

**140. How do you test a mobile app with limited devices?**  
Test on critical devices and use emulators/simulators.

**141. How do you test for accessibility?**  
Check for screen reader compatibility, keyboard navigation, and contrast.

**142. How do you test a new feature in production?**  
Use feature flags and perform smoke tests in the live environment.

**143. How do you test in an agile environment?**  
Collaborate closely with developers, perform exploratory testing, and keep test cases short.

**144. How do you test a bug fix?**  
Perform regression tests around the changed area.

**145. How do you handle repetitive test cases?**  
Document them and evaluate for automation later.

**146. How do you test performance manually?**  
Observe response times and simulate load with multiple users if possible.

**147. How do you test third-party integrations?**  
Validate data exchange, error handling, and contract compliance.

**148. How do you test cloud-based apps?**  
Check scalability, security, and multi-location performance.

**149. How do you test an API manually?**  
Use tools like Postman to send requests and verify responses.

**150. How do you ensure test coverage?**  
Map test cases to requirements in a traceability matrix.

**151. How do you test a file upload feature?**  
Check allowed formats, file size limits, and error handling.

**152. How do you test notifications?**  
Check timing, format, content, and delivery method.

**153. How do you test a form?**  
Check validations, error messages, and submission process.

**154. How do you test session management?**  
Verify timeouts, logout functionality, and session persistence.

**155. How do you test email functionality?**  
Check correct recipients, format, and spam folder behavior.

**156. How do you test an API without documentation?**  
Use tools to inspect requests/responses and reverse engineer usage.

**157. How do you test browser compatibility?**  
Run tests across different browsers and devices.

**158. How do you test a multi-language application?**  
Check translations, formatting, and cultural appropriateness.

**159. How do you test a chat application?**  
Check message delivery, typing indicators, and offline messages.

**160. How do you test a data migration?**  
Compare source and destination data for completeness and accuracy.

**Part 3: Interview-Focused (161–200)**

**161. Explain STLC in 2 minutes.**  
Requirement Analysis → Planning → Test Design → Execution → Defect Reporting → Closure.

**162. Difference between regression and retesting?**

* **Regression:** Checks old functionality.
* **Retesting:** Checks if a specific defect is fixed.

**163. Difference between severity and priority with an example?**  
High severity, low priority: A typo in a legal disclaimer.

**164. Give an example of a high priority, low severity bug.**  
A company logo linking to the wrong page before a big launch.

**165. What is boundary value analysis example?**  
If age limit is 18–60, test 17, 18, 60, and 61.

**166. What is equivalence partitioning example?**  
Group ages into <18, 18–60, and >60 for testing.

**167. What are exit criteria in testing?**  
Conditions that must be met to stop testing.

**168. What is entry criteria in testing?**  
Conditions that must be met to start testing.

**169. What is the difference between validation and verification?**  
Verification checks process, validation checks product.

**170. What are non-functional testing types?**  
Performance, usability, security, compatibility.

**171. What are functional testing types?**  
Unit, integration, system, acceptance.

**172. What is a bug triage meeting?**  
A meeting to prioritize and assign defects.

**173. What is a use case?**  
A sequence of steps showing how a user interacts with the system.

**174. What is the difference between positive and negative testing?**  
Positive tests valid inputs; negative tests invalid inputs.

**175. What is a test harness example?**  
A toolset for running tests with mock data.

**176. What is a test oracle example?**  
A database dump used to verify API responses.

**177. What is the pesticide paradox in real life?**  
Running the same login tests daily stops finding new issues.

**178. What is an example of defect clustering?**  
Most bugs in an e-commerce site are found in checkout.

**179. What is exploratory testing example?**  
Randomly navigating a new mobile app to find issues.

**180. What is the difference between alpha and beta testing with example?**

* **Alpha:** Internal testing before release.
* **Beta:** Users test before final release.

**181. How do you test a feature with no UI?**  
Test APIs or backend logs.

**182. How do you report a bug effectively?**  
Include title, steps, expected vs actual, screenshots/logs.

**183. How do you explain a missed defect?**  
Acknowledge, find the cause, and add coverage.

**184. How do you handle conflicts with developers?**  
Stay professional, show proof, and focus on the issue.

**185. How do you handle too many open bugs?**  
Prioritize by severity and business impact.

**186. What is the main skill a manual tester should have?**  
Attention to detail.

**187. Why is communication important in testing?**  
To clearly report defects and collaborate with teams.

**188. Why should testers know SQL?**  
To validate backend data.

**189. Why should testers know basic API testing?**  
To test services without full UI.

**190. How does Agile affect testing?**  
Testing is continuous and iterative.

**191. What is continuous testing example?**  
Running regression tests automatically after each build.

**192. What is shift-left example?**  
Reviewing test cases during design phase.

**193. What is shift-right example?**  
Monitoring live app performance.

**194. Why is exploratory testing useful in Agile?**  
It’s fast and adaptable.

**195. How do you test for security manually?**  
Check input validation, session handling, and access control.

**196. What is compliance testing?**  
Ensuring the app meets legal/regulatory requirements.

**197. What is end-to-end testing example?**  
Placing an order and tracking it to delivery.

**198. What is integration testing example?**  
Testing payment gateway and order management together.

**199. What is the future of manual testing?**  
More focus on usability, exploratory, and domain knowledge.

**200. What advice for a beginner manual tester?**  
Learn basics, practice test cases, and understand the business domain.