**Section 1: Testing Fundamentals & Principles**

**1. Which of the following is NOT a primary objective of testing?**

a) Finding defects

b) Gaining confidence in the level of quality

c) Providing information for decision-making

d) Fixing all the identified defects

**Answer:** d (Testers find defects; developers usually fix them)

**2. "Testing shows the presence of defects, not their absence." This principle implies:**

a) If no defects are found, the software is defect-free.

b) Testing can prove that software is 100% correct.

c) Even after thorough testing, defects may still exist.

d) Testing should be stopped once a certain number of defects are found.

**Answer:** c

**3. Exhaustive testing (testing all possible inputs and conditions) is:**

a) Always recommended for critical applications.

b) Practically impossible for most real-world software.

c) The only way to ensure 100% defect-free software.

d) Primarily focused on during unit testing.

**Answer:** b

**4. The "Pesticide Paradox" in software testing suggests that:**

a) Using too many testing tools can be harmful.

b) If the same tests are repeated many times, they become less effective at finding new defects.

c) Defects tend to cluster in specific modules.

d) Testing should be performed by an independent team.

**Answer:** b

**5. What does "Defect Clustering" principle state?**

a) Defects are usually introduced by a small number of developers.

b) A small number of modules usually contain most of the defects discovered.

c) Defects found late in the lifecycle are more expensive to fix.

d) Testers should focus on areas where no defects have been found yet.

**Answer:** b

**6. "Testing is context-dependent" means:**

a) The testing approach for an e-commerce site will be different from that for an aviation control system.

b) All software should be tested using the same set of standard test cases.

c) The context of a defect report is more important than the defect itself.

d) Testing should only depend on the available requirements.

**Answer:** a

**7. Which activity is part of the fundamental test process?**

a) Code compilation

b) Requirements gathering

c) Test planning and control

d) Database administration

**Answer:** c

**Section 2: Test Levels & Types**

**8. Which test level focuses on verifying the interaction between different software components or modules?**

a) Unit Testing

b) Integration Testing

c) System Testing

d) Acceptance Testing

**Answer:** b

**9. System Testing is primarily concerned with:**

a) Testing individual functions or methods within a module.

b) Verifying that the integrated system meets specified requirements.

c) Testing if the system is fit for use by the end-users in their operational environment.

d) Ensuring that code changes in one part do not affect other parts.

**Answer:** b

**10. User Acceptance Testing (UAT) is typically performed by:**

a) Developers

b) Independent test team

c) End-users or customers

d) System architects

**Answer:** c

**11. What is the main purpose of Smoke Testing?**

a) To perform in-depth testing of all features.

b) To verify that the most critical functionalities of the application are working fine before detailed testing.

c) To test the system under heavy load.

d) To find obscure defects in edge cases.

**Answer:** b

**12. Regression Testing aims to:**

a) Test a specific module after it has been fixed.

b) Ensure that new changes or defect fixes have not negatively impacted existing functionality.

c) Test the system for the first time.

d) Test the user interface for usability issues.

**Answer:** b

**13. Re-testing is performed to:**

a) Ensure that the original defect has been successfully fixed.

b) Find new defects in unchanged areas of the software.

c) Test the system's performance.

d) Check if the build is stable enough for further testing.

**Answer:** a

**14. Which type of testing verifies if the software can handle expected and unexpected increases in load?**

a) Usability Testing

b) Security Testing

c) Performance Testing (specifically Stress/Load testing)

d) Compatibility Testing

**Answer:** c

**15. Alpha testing is conducted at:**

a) The developer's site by internal staff.

b) The end-user's site by end-users.

c) The developer's site by potential customers.

d) A third-party testing facility.

**Answer:** a or c (Often internal staff, sometimes with select customers at developer's site)

**16. Which of the following is a non-functional testing type?**

a) Sanity Testing

b) Usability Testing

c) Database Testing (functional aspects)

d) White-box Testing

**Answer:** b (Others can have non-functional aspects, but Usability is inherently non-functional)

**17. Testing without any formal test cases or documentation, relying on the tester's intuition and experience, is known as:**

a) Ad-hoc Testing

b) Regression Testing

c) User Acceptance Testing

d) Performance Testing

**Answer:** a

**Section 3: Test Design Techniques**

**18. Equivalence Class Partitioning (ECP) aims to:**

a) Test every possible input value.

b) Divide input data into groups from which test cases can be derived, assuming all values in a group will behave similarly.

c) Test only the boundary values of input domains.

d) Focus on the transitions between different system states.

**Answer:** b

**19. Boundary Value Analysis (BVA) is a test design technique that focuses on:**

a) The center values of an equivalence partition.

b) The edges or boundaries of input and output equivalence partitions.

c) All possible combinations of input conditions.

d) The paths through the code.

**Answer:** b

**20. Which technique is most suitable for testing scenarios with multiple conditions that result in different actions?**

a) State Transition Testing

b) Boundary Value Analysis

c) Decision Table Testing

d) Use Case Testing

**Answer:** c

**21. State Transition Testing is best used for systems that:**

a) Have a large number of input parameters.

b) Respond differently to an event depending on its current condition or state.

c) Require checking of specific boundary conditions.

d) Involve complex calculations.

**Answer:** b

**22. Which of these is primarily a black-box test design technique?**

a) Statement Coverage

b) Decision Coverage

c) Equivalence Partitioning

d) Path Coverage

**Answer:** c

**23. Error Guessing is an example of which type of test design technique?**

a) Specification-based

b) Structure-based

c) Experience-based

d) Model-based

**Answer:** c

**24. Use Case Testing is primarily focused on:**

a) Testing individual UI elements.

b) Verifying business scenarios or user interactions from start to finish.

c) Analyzing the internal logic of the code.

d) Identifying all possible states of the system.

**Answer:** b

**Section 4: Test Management & Control**

**25. What is the primary purpose of a Test Plan document?**

a) To list all the defects found during testing.

b) To describe the scope, approach, resources, and schedule of intended test activities.

c) To provide detailed steps for executing specific tests.

d) To summarize the results of the testing phase.

**Answer:** b

**26. Which of the following is a key element of a Test Strategy?**

a) Detailed schedule of test execution for specific modules.

b) High-level approach to testing and selection of test levels/types.

c) Specific test data to be used for each test case.

d) List of all testers assigned to the project.

**Answer:** b

**27. Entry criteria in a test plan define:**

a) The conditions that must be met to stop testing.

b. The conditions that must be met to start a specific test phase.

c) The format for reporting defects.

d) The tools to be used for testing.

**Answer:** b

**28. Exit criteria in a test plan often include:**

a) Availability of the test environment.

b) Completion of test case design.

c) Achieving a certain level of test coverage or defect density.

d) Budget approval for the testing phase.

**Answer:** c

**29. What is a Test Case?**

a) A document describing the overall approach to testing.

b) A set of inputs, execution preconditions, expected results, and execution post conditions, developed for a particular objective.

c) A summary of all defects found.

d) A tool used for automating tests.

**Answer:** b

**30. A good test case should be:**

a) Long and descriptive, covering multiple scenarios at once.

b) Clear, concise, traceable to requirements, and have a clear expected result.

c) Vague to allow for tester flexibility.

d) Focused only on positive scenarios.

**Answer:** b

**31. Test Data is:**

a) The output generated by the system after test execution.

b) The specific inputs used to execute a test case.

c) The environment configuration for testing.

d) The criteria used to evaluate test results.

**Answer:** b

**Section 5: Defect Management**

**32. What is the first step in a typical defect lifecycle?**

a) Fixed

b) Verified

c) New/Open

d) Retest

**Answer:** c

**33. A defect report should ideally include:**

a) Only the steps to reproduce and the actual result.

b) Steps to reproduce, expected result, actual result, severity, and priority.

c) A proposed solution for the defect.

d) The name of the developer who introduced the defect.

**Answer:** b

**34. Severity of a defect refers to:**

a) The urgency to fix the defect from a business perspective.

b) The impact of the defect on the functionality or system.

c) The number of times the defect has been encountered.

d) The ease with which the defect can be fixed.

**Answer:** b

**35. Priority of a defect refers to:**

a) The technical complexity of the defect.

b) The impact of the defect on system performance.

c) The order in which defects should be fixed, often based on business impact.

d) The stage in which the defect was found.

**Answer:** c

**36. When a defect is fixed by the developer, its status typically changes to:**

a) Reopened

b) Closed

c) Fixed/Resolved

d) Deferred

**Answer:** c

**37. If a tester finds that a "Fixed" defect is still reproducible, the defect status should be changed to:**

a) Closed

b) Verified

c) Reopened/Rejected

d) Deferred

**Answer:** c

**38. A "Deferred" defect is one that:**

a) Has been fixed and is awaiting re-testing.

b) Cannot be reproduced by the development team.

c) Will be fixed in a future release.

d) Is considered a duplicate of another defect.

**Answer:** c

**Section 6: Testing in Different Lifecycles**

**39. In Agile methodologies like Scrum, when does testing typically occur?**

a) Only at the end of the entire project.

b) As a separate phase after all development sprints are completed.

c) Continuously throughout each sprint, alongside development.

d) Primarily before the sprint planning meeting.

**Answer:** c

**40. Which statement is TRUE about testing in a V-Model?**

a) Testing activities are planned only after coding is complete.

b) Each development phase has a corresponding test phase planned in parallel.

c) It is best suited for projects with unclear or changing requirements.

d) UAT is performed before System Testing.

**Answer:** b

**41. A key characteristic of testing in a Waterfall model is:**

a) Overlapping of testing and development phases.

b) Testing is performed sequentially after the completion of the development phase.

c) Frequent delivery of testable software increments.

d) Heavy reliance on exploratory testing.

**Answer:** b

**42. "Shift Left" in testing refers to:**

a) Delaying testing activities until the end of the development cycle.

b) Involving testers and starting test activities earlier in the software development lifecycle.

c) Shifting all testing responsibility to the development team.

d) Focusing testing only on the left-most modules in an architecture diagram.

**Answer:** b

**Section 7: Static vs. Dynamic Testing & Verification vs. Validation**

**43. Static testing involves:**

a) Executing the code to find defects.

b) Analyzing documents (like requirements, design, code) without executing the software.

c) Testing the system with a large volume of data.

d) Testing the system's response time.

**Answer:** b

**44. Which of the following is an example of a static testing technique?**

a) Unit Testing

b) Code Review

c) Performance Testing

d) User Acceptance Testing

**Answer:** b

**45. Dynamic testing involves:**

a) Reviewing the source code for syntax errors.

b) Executing the software with test data to observe its behavior.

c) Analyzing the system architecture documents.

d) Conducting walkthroughs of requirement specifications.

**Answer:** b

**46. Verification in software testing primarily answers the question:**

a) "Are we building the right product?"

b) "Are we building the product right?"

c) "Is the product usable by the customer?"

d) "Is the product fast enough?"

**Answer:** b (Checks if software conforms to its specification)

**47. Validation in software testing primarily answers the question:**

a) "Are we building the product right?"

b) "Does the code have any syntax errors?"

c) "Are we building the right product?"

d) "Are individual modules working correctly?"

**Answer:** c (Checks if software meets user needs and requirements)

**Section 8: Miscellaneous & QA/QC**

**48. What is Quality Assurance (QA)?**

a) The process of executing test cases to find defects.

b) A set of activities designed to ensure that the development and maintenance processes are adequate to produce high-quality software.

c) The process of fixing defects found during testing.

d) The final check performed by end-users before release.

**Answer:** b (It's process-oriented)

**49. What is Quality Control (QC)?**

a) The establishment of processes and standards to prevent defects.

b) A set of activities designed to evaluate a developed work product (like testing).

c) Defining the roles and responsibilities within a test team.

d) The act of reviewing requirement documents.

**Answer:** b (It's product-oriented, testing is a QC activity)

**50. Which of the following is a primary benefit of using test management tools?**

a) They automatically write all test cases.

b) They eliminate the need for manual testing.

c) They help in organizing test assets, tracking execution, and managing defects.

d) They guarantee that the software will be defect-free.

**Answer:** c

**Section 9: Test Design Techniques (Continued & Advanced)**

**51. Pairwise testing (or All-Pairs Testing) is a technique used to:**

a) Test all possible combinations of input parameters.

b) Test each input parameter with every other input parameter at least once.

c) Reduce the number of test cases by testing only pairs of interacting parameters.

d) Test software in pairs of testers.

**Answer:** c

**52. Orthogonal Array Testing (OAT) is a statistical test design technique that aims to:**

a) Ensure every possible input is tested.

b) Achieve a balanced and minimal set of test cases when dealing with multiple parameters and their interactions.

c) Test the system by transitioning between all possible states.

d) Focus only on boundary conditions.

**Answer:** b

**53. In Cause-Effect Graphing, nodes representing causes and effects are linked by:**

a) Flow lines

b) Logical operators (AND, OR, NOT)

c) Arrows indicating transitions

d) Use case steps

**Answer:** b

**54. When is Exploratory Testing most effectively used?**

a) When requirements are fully defined and stable.

b) When there is limited time, or to complement formal testing by leveraging tester experience and intuition.

c) Only for unit testing by developers.

d) As a replacement for all scripted testing.

**Answer:** b

**55. A checklist-based testing approach is most useful when:**

a) The system is highly complex and requires deep analysis.

b) Testers have limited domain knowledge but need to cover standard functionalities systematically.

c) The primary goal is to find obscure defects.

d) Requirements are still evolving rapidly.

**Answer:** b

**Section 10: Test Types (More Granular)**

**56. Which aspect is primarily evaluated during Usability Testing?**

a) The system's ability to handle a large number of concurrent users.

b) The ease of use, learnability, and user satisfaction with the software.

c) The protection of data from unauthorized access.

d) The software's ability to run on different operating systems.

**Answer:** b

**57. Security testing often involves looking for vulnerabilities related to:**

a) Page load times and server response.

b) SQL injection, cross-site scripting (XSS), and authentication/authorization flaws.

c) How well the application recovers from hardware failures.

d) The visual appeal and layout of the user interface.

**Answer:** b

**58. Compatibility Testing ensures that the software:**

a) Meets all the specified functional requirements.

b) Works correctly in different environments (e.g., browsers, OS, hardware, network).

c) Is easy for new users to learn.

d) Can handle a high volume of transactions.

**Answer:** b

**59. Load Testing, a type of performance testing, primarily checks:**

a) The system's behavior under normal and anticipated peak load conditions.

b) The system's breaking point when subjected to extreme loads.

c) The system's stability over a long period under normal load.

d) The system's response to sudden bursts of high activity.

**Answer:** a

**60. Stress Testing is designed to evaluate the system's behavior:**

a) During typical daily usage.

b) When resources like memory or CPU are scarce, or load is beyond normal operating limits.

c) Over an extended duration of continuous operation.

d) Across different mobile devices.

**Answer:** b

**61. Endurance Testing (or Soak Testing) focuses on:**

a) The system's ability to handle a brief, massive increase in load.

b) Identifying performance issues like memory leaks that appear only after the system has been running for a long time.

c) The initial response time of the application under light load.

d) How quickly a user can complete a specific task.

**Answer:** b

**62. Accessibility Testing (A11y) aims to ensure that the application is usable by:**

a) Only expert users.

b) People with disabilities (e.g., visual, auditory, motor, cognitive).

c) Users in different geographical locations.

d) Users with older hardware configurations.

**Answer:** b

**63. Localization Testing verifies that the software:**

a) Can be easily translated into multiple languages.

b) Is adapted correctly for a specific target locale, including language, cultural conventions, and regional settings.

c) Functions correctly across different time zones.

d) Has a globally appealing user interface design.

**Answer:** b

**64. Internationalization (I18n) testing focuses on ensuring that the software:**

a) Is fully translated into a specific target language.

b) Can be easily adapted to various languages and regions without engineering changes.

c) Is free from culturally offensive content.

d) Has been tested in all target countries.

**Answer:** b (I18n is about designing the software to be adaptable; L10n is the actual adaptation)

**Section 11: Test Management & Control (Advanced)**

**65. Which of the following is a common test estimation technique?**

a) Code Coverage Analysis

b) Defect Density Calculation

c) Work Breakdown Structure (WBS) or Expert Judgment (e.g., Delphi technique)

d) Use Case Point Analysis

**Answer:** c (Use Case Points can be used for estimation, but WBS/Expert Judgment are more direct test estimation techniques)

**66. Risk-Based Testing involves:**

a) Testing only the high-risk modules and ignoring low-risk ones.

b) Allocating testing effort and prioritizing tests based on the assessed level of product risk.

c) Eliminating all risks before starting the testing phase.

d) A testing approach where only senior testers handle risky modules.

**Answer:** b

**67. What is the primary purpose of a Test Summary Report?**

a) To list all the test cases executed.

b) To provide detailed steps for each failed test case.

c) To provide stakeholders with a summary of testing activities, results, and an assessment of quality against exit criteria.

d) To request more resources for the testing team.

**Answer:** c

**68. Test progress monitoring often involves tracking metrics like:**

a) Number of lines of code written by developers.

b) Percentage of test cases executed, passed, failed, and blocked.

c) Number of requirements gathered.

d) Team morale and velocity (Agile context, but less direct for test \*progress\* monitoring).

**Answer:** b

**69. Which factor is LEAST likely to be considered a component of product risk in risk-based testing?**

a) Likelihood of failure

b) Impact of failure

c) The programming language used by developers

d) Complexity of the feature

**Answer:** c (While it \*could\* indirectly influence complexity or likelihood, it's less direct a risk component than the others)

**70. Configuration Management in testing is important for:**

a) Automatically generating test data.

b) Tracking versions of software, testware (test cases, scripts), and test environments to ensure consistency and reproducibility.

c) Managing the test team's schedule.

d) Calculating the return on investment (ROI) of testing.

**Answer:** b

**Section 12: Testing in Agile (More Details)**

**71. In an Agile context, the "Definition of Done" (DoD) for a user story often includes:**

a) Only that the code is written.

b) Code is written, unit tested, integration tested, and passes acceptance criteria.

c) The product owner has reviewed the story.

d) The sprint has ended.

**Answer:** b (It implies a potentially shippable state, which includes thorough testing)

**72. Who is typically responsible for writing acceptance criteria for a user story in Agile?**

a) Primarily the development team

b) Primarily the Scrum Master

c) Primarily the Product Owner, often in collaboration with the team (including testers).

d) Only the testers

**Answer:** c

**73. Continuous Integration (CI) in an Agile environment facilitates testing by:**

a) Eliminating the need for a dedicated test environment.

b) Automatically running tests (e.g., unit, integration) whenever new code is integrated, providing rapid feedback.

c) Replacing all manual testing efforts.

d) Managing the product backlog.

**Answer:** b

**74. Test automation in Agile projects is crucial because it:**

a) Replaces the need for manual exploratory testing.

b) Allows for rapid regression testing of frequently changing code within short iterations.

c) Is the only way to test user stories.

d) Guarantees 100% test coverage.

**Answer:** b

**75. A "Test Pyramid" in Agile testing typically suggests:**

a) Having more UI tests than unit tests.

b) Having a large base of fast unit tests, a smaller layer of service/integration tests, and a very small layer of slow UI/end-to-end tests.

c) Testing only through the user interface.

d) Focusing all testing efforts at the end of the sprint.

**Answer:** b

**Section 13: Defect Management (Advanced)**

**76. Defect Triage meetings are primarily held to:**

a) Fix all the reported defects.

b) Decide on the validity, priority, severity, and assignment of newly reported defects.

c) Write new test cases based on defects found.

d) Train developers on how to avoid defects.

**Answer:** b

**77. Root Cause Analysis (RCA) for defects aims to:**

a) Only fix the symptom of the defect.

b) Identify the underlying reason why the defect was introduced in the first place to prevent similar defects in the future.

c) Determine who is responsible for introducing the defect.

d) Calculate the cost of fixing the defect.

**Answer:** b

**78. Which of the following is a common technique used in Root Cause Analysis?**

a) Boundary Value Analysis

b) The "5 Whys" technique or Fishbone (Ishikawa) Diagram

c) Use Case Testing

d) Code Coverage Measurement

**Answer:** b

**79. A defect is classified as "Cannot Reproduce" or "Not Reproducible" when:**

a) The developer does not understand the defect report.

b) The tester made a mistake in reporting the defect.

c) The development team is unable to replicate the defect based on the information provided.

d) The defect has already been fixed.

**Answer:** c

**80. A "Masked Defect" is a defect that:**

a) Is intentionally hidden by developers.

b) Prevents the discovery of another defect.

c) Only appears under very specific, rare conditions.

d) Has been found and fixed in a previous version.

**Answer:** b (Its presence conceals or "masks" another underlying issue)

**Section 14: Test Environment Management & Test Data**

**81. The primary purpose of a dedicated Test Environment is to:**

a) Provide a stable and controlled setting for executing tests that closely mirrors production (where feasible).

b) Allow developers to experiment with new coding techniques.

c) Store all historical test data.

d) Host the project management tools.

**Answer:** a

**82. Which characteristic is desirable for test data?**

a) It should always be real production data.

b) It should be readily available, cover various scenarios (positive, negative, boundary), and be manageable.

c) It should be minimal to speed up test execution.

d) It should be generated only by developers.

**Answer:** b (Using real production data has privacy/security concerns and might not cover all test conditions)

**83. Data-Driven Testing is an approach where:**

a) Test data is hardcoded within test scripts.

b) Test scripts read input values and expected results from external data sources (e.g., spreadsheets, databases).

c) Only database testing is performed.

d) Test data is generated randomly for each test run.

**Answer:** b

**84. If a test environment is not identical to the production environment, what is a potential risk?**

a) Test execution will always be faster.

b) Defects found in testing may not occur in production, or defects missed in testing may appear in production.

c) Test reporting becomes impossible.

d) It will reduce the number of test cases needed.

**Answer:** b

**Section 15: Static Testing Techniques (More Detail)**

**85. A "Walkthrough" as a static review technique is typically:**

a) A formal, documented review process led by a moderator.

b) Led by the author of the document/code, aiming to gather feedback and achieve a common understanding.

c) An independent assessment by an external auditor.

d) Focused solely on finding defects against a checklist.

**Answer:** b

**86. An "Inspection" is considered the most formal review type because it:**

a) Is always led by senior management.

b) Typically involves defined roles, checklists, entry/exit criteria, metrics collection, and a trained moderator.

c) Does not require any preparation from reviewers.

d) Is only performed on completed code.

**Answer:** b

**87. Which of the following is a key benefit of performing static testing early in the lifecycle?**

a) It guarantees that no defects will be found during dynamic testing.

b) It can find defects when they are cheaper to fix and can prevent defect multiplication.

c) It completely replaces the need for dynamic testing.

d) It is primarily used to assess the performance of the code.

**Answer:** b

**Section 16: Test Levels in Context**

**88. Component Testing (Unit Testing) is primarily the responsibility of:**

a) End-users

b) Independent testers

c) Developers

d) Business analysts

**Answer:** c

**89. "Big Bang" integration testing involves:**

a) Integrating modules one by one.

b) Integrating all or most modules simultaneously and then testing the entire system.

c) Using stubs and drivers to test integration incrementally.

d) Testing integrations with external systems only.

**Answer:** b

**90. Which of the following is an advantage of incremental integration testing (e.g., top-down, bottom-up) over Big Bang?**

a) It requires less overall testing effort.

b) Defects can be found earlier and are easier to localize.

c) It doesn't require the use of stubs or drivers.

d) It can only be done after all modules are fully developed.

**Answer:** b

**91. Alpha Testing is a form of:**

a) Component Testing

b) Integration Testing

c) System Testing

d) Acceptance Testing

**Answer:** d (It's a type of acceptance testing, albeit internal or at the developer's site)

**92. Operational Acceptance Testing (OAT) typically focuses on aspects like:**

a) Functional correctness of individual features.

b) Backup/restore procedures, maintainability, and performance in a production-like environment.

c) User interface aesthetics.

d) Adherence to coding standards.

**Answer:** b

**Section 17: Test Tools & Automation (Manual Perspective)**

**93. From a manual tester's perspective, a test execution tool can help by:**

a) Automatically designing all test cases.

b) Capturing and replaying manual test steps, especially for regression.

c) Writing the initial requirements.

d) Eliminating the need for understanding the application.

**Answer:** b (Even if primarily manual, some tools can assist with parts of execution or logging)

**94. When evaluating a new test management tool, a key consideration would be:**

a) The number of defects it can automatically find.

b) Its ability to integrate with other tools (e.g., defect trackers, automation frameworks) and its ease of use.

c) Whether it can write code for the application.

d) Its popularity on social media.

**Answer:** b

**Section 18: Tester Mindset & Skills**

**95. A key characteristic of a good manual tester is:**

a) Only following test cases rigidly without any deviation.

b) Having a critical and inquisitive mindset, attention to detail, and good communication skills.

c) Blaming developers for defects found.

d) Focusing solely on finding as many defects as possible, regardless of severity.

**Answer:** b

**96. Confirmation Bias in testing refers to the tendency to:**

a) Test in a way that confirms the software works correctly, potentially overlooking defects.

b) Prioritize defects based on personal preference.

c) Confirm all test results with the development team.

d) Only test features that one is familiar with.

**Answer:** a

**97. Why is domain knowledge important for a tester?**

a) It allows them to write code for the application.

b) It helps them understand user expectations, business rules, and design more effective test cases.

c) It is only important for developers.

d) It makes test execution faster.

**Answer:** b

**Section 19: Standards & Process Improvement**

**98. ISO/IEC 29119 is a set of international standards related to:**

a) Software project management

b) Software testing processes, documentation, and techniques

c) Information security management

d) Software coding best practices

**Answer:** b

**99. The TMMi (Test Maturity Model integration) is a framework used to:**

a) Automatically generate test cases.

b) Assess and improve the maturity of an organization's testing processes.

c) Manage test data.

d) Define the roles within a test team.

**Answer:** b

**100. Which of the following best describes "Requirement Traceability" in testing?**

a) The ability to trace defects back to the tester who found them.

b) The ability to link test cases back to the specific requirements they are intended to verify.

c) The process of tracing the execution path of a test case through the code.

d) Documenting how requirements change over time.

**Answer:** b

**Section 20: Specific Scenarios & Advanced Test Types**

**101. When testing a login form, which scenario represents a negative test case for the password field based on a "6-12 characters" requirement?**

a) Entering a 7-character password.

b) Entering a 10-character password.

c) Entering a 5-character password.

d) Leaving the password field blank (this is another negative case, but c directly tests the boundary).

**Answer:** c

**102. You are testing a feature that calculates shipping costs based on weight and destination zone. Which test design technique is MOST suitable for ensuring all combinations of rules are covered?**

a) State Transition Testing

b) Equivalence Partitioning

c) Decision Table Testing

d) Exploratory Testing

**Answer:** c

**103. During usability testing for a mobile app, observing a user repeatedly tapping the wrong icon to achieve a common task would indicate an issue with:**

a) System performance

b) Data integrity

c) Learnability or intuitiveness

d) Security vulnerabilities

**Answer:** c

**104. API testing primarily focuses on verifying:**

a) The graphical user interface of the application.

b) The business logic layer through programmatic interfaces, checking requests, responses, and performance.

c) The database schema and stored procedures.

d) The compatibility of the application across different browsers.

**Answer:** b

**105. When testing a "Forgot Password" functionality, a critical security test case would be:**

a) Ensuring the password reset email is sent quickly.

b) Verifying that the reset link correctly allows a user to set a new password.

c) Checking if the reset link can be easily guessed or if it exposes sensitive information.

d) Testing if the UI elements on the reset page are aligned correctly.

**Answer:** c

**106. Mutation Testing is a technique primarily used to:**

a) Evaluate the quality and effectiveness of an existing set of test cases.

b) Test how the system handles mutated (invalid) input data.

c) Introduce intentional faults (mutations) into the code and check if test cases can detect them.

d) Test genetic algorithms.

**Answer:** c (Often more white-box, but understanding the concept is useful for thoroughness)

**107. A/B Testing is a method commonly used to:**

a) Test two different versions of a component in isolation.

b) Compare two versions of a webpage or app feature to see which one performs better in terms of user engagement or conversion.

c) Ensure both Alpha and Beta testing have been completed.

d) Test the application before and after a major code change.

**Answer:** b

**Section 21: Advanced Defect Lifecycle & Attributes**

**108. What does the defect status "Duplicate" imply?**

a) The defect cannot be reproduced.

b) The reported defect is identical or very similar to an already existing defect report.

c) The defect has been fixed twice by mistake.

d) The defect will not be fixed.

**Answer:** b

**109. If a defect is marked as "By Design" or "As Designed," it means:**

a) The reported behavior is an actual defect that needs to be fixed.

b) The development team designed the defect intentionally.

c) The reported behavior, while perhaps unexpected by the tester, is the intended functionality according to specifications or design decisions.

d) The defect will be addressed in a future design iteration.

**Answer:** c

**110. "Defect Age" or "Bug Age" typically refers to:**

a) The experience level of the tester who found the defect.

b) The time elapsed since the defect was first reported.

c) The estimated time it will take to fix the defect.

d) The version of the software in which the defect was introduced.

**Answer:** b

**111. Why is it important to capture the "Detected in Version/Build" for a defect?**

a) To blame the development team for that specific build.

b) To help identify when the defect was introduced and to track fix verification in subsequent builds.

c) To determine the defect's priority.

d) It's only important for automated defect reporting.

**Answer:** b

**Section 22: Test Documentation Details**

**112. Which element is typically NOT part of a detailed Test Case Specification?**

a) Test Case ID

b) Test Data

c) Root Cause Analysis of a potential defect

d) Expected Result

**Answer:** c (RCA is done after a defect is found and analyzed, not as part of the initial test case)

**113. A Test Log or Test Execution Log primarily contains:**

a) The overall test plan and strategy.

b) A chronological record of test execution details, including test cases run, actual results, status (pass/fail), and environment used.

c) Only the list of failed test cases.

d) The business requirements for the system.

**Answer:** b

**114. The "Prerequisites" section in a test case specifies:**

a) The steps to reproduce a defect.

b) The conditions that must be true or actions that must be performed before the test case can be executed.

c) The expected outcome of the test.

d) The tools required for testing.

**Answer:** b

**115. A "Test Charter" used in Exploratory Testing typically includes:**

a) Detailed step-by-step instructions.

b) The mission or goal for the test session, target features, and potential risks to investigate.

c) The expected results for all possible inputs.

d) A list of defects found in previous sessions.

**Answer:** b

**Section 23: Test Coverage Concepts**

**116. Requirements Coverage is a metric that measures:**

a) The percentage of code that has been executed by tests.

b) The percentage of specified requirements for which corresponding test cases have been designed and/or executed.

c) The number of defects found per requirement.

d) The time spent testing each requirement.

**Answer:** b

**117. From a black-box testing perspective, "Decision Coverage" (if requirements are written with clear conditions/decisions) aims to ensure:**

a) Every line of code has been executed.

b) All possible outcomes of logical decisions within the requirements (e.g., IF-THEN-ELSE conditions) have been tested.

c) Every module of the software has been tested.

d) Only the positive scenarios for each decision are tested.

**Answer:** b

**118. Achieving 100% test coverage (e.g., requirements coverage) guarantees:**

a) The software is completely defect-free.

b) All specified functionalities have been tested according to the defined coverage criteria, but it doesn't guarantee the absence of all defects.

c) No further testing is needed.

d) The testing process was perfect.

**Answer:** b

**Section 24: Testing Specific Application Types**

**119. When testing a web application for responsiveness, a manual tester would primarily check:**

a) The speed of database queries.

b) How the application layout and functionality adapt to different screen sizes and devices (desktop, tablet, mobile).

c) The number of concurrent users the application can handle.

d) The security of data transmission over HTTPS.

**Answer:** b

**120. Cross-Browser Testing for a web application is essential to ensure:**

a) The application loads quickly on all browsers.

b) Consistent functionality, look, and feel across different web browsers (e.g., Chrome, Firefox, Safari, Edge).

c) The application is secure from browser-specific exploits.

d) All users are using the latest browser versions.

**Answer:** b

**121. Key considerations for manual testing of Mobile Applications (compared to web applications) include:**

a) Only testing on emulators.

b) Network conditions (WiFi, cellular data, offline mode), device fragmentation (OS versions, screen sizes), gestures, and app lifecycle (backgrounding, interruptions).

c) Ignoring battery consumption.

d) Assuming all users have the latest flagship devices.

**Answer:** b

**122. Session Management testing for web applications primarily aims to verify:**

a) How quickly web pages’ load.

b) That user sessions are handled securely, including proper creation, maintenance, timeout, and invalidation.

c) The visual design of session-related messages.

d) That users can have multiple active sessions simultaneously without issue (if intended).

**Answer:** b

**123. Cookie testing for web applications involves verifying:**

a) That cookies are delicious.

b) That cookies are correctly set, read, deleted, and that sensitive information is not stored insecurely in them.

c) The number of cookies a browser can store.

d) The colour and shape of cookie consent banners.

**Answer:** b

**Section 25: Introduction to Performance Concepts (Manual Perspective)**

**124. From a manual tester's perspective, a simple way to get an initial feel for an application's "Response Time" is to:**

a) Analyze server logs for CPU utilization.

b) Use a stopwatch or browser developer tools to measure the time taken for a page to load or an action to complete.

c) Ask developers how fast it should be.

d) Assume it's acceptable if no users complain.

**Answer:** b

**125. "Throughput" as a performance metric generally refers to:**

a) The speed of an individual user's connection.

b) The number of transactions or requests a system can process per unit of time (e.g., transactions per second).

c) The amount of data stored in the database.

d) The time it takes for the system to start up.

**Answer:** b

**126. If a manual tester observes that an application becomes significantly slower when multiple users access it simultaneously, this could be an indication of poor:**

a) Usability

b) Scalability or load handling capability

c) Security

d) Data integrity

**Answer:** b

**Section 26: Risk in Testing (More Details)**

**127. A "Risk Mitigation" strategy in testing involves:**

a) Ignoring all identified risks.

b) Taking actions to reduce the likelihood or impact of a risk (e.g., performing more thorough testing on high-risk modules).

c) Transferring the risk to the customer.

d) Accepting the risk without any action.

**Answer:** b

**128. Which of these is an example of a "Project Risk" that can impact testing?**

a) A complex algorithm in the software that might have calculation errors (this is a product risk).

b) Lack of skilled testing resources or unrealistic deadlines.

c) Potential security vulnerabilities in the login feature (product risk).

d) The software not meeting user needs (product risk).

**Answer:** b

**129. "Risk Appetite" refers to:**

a) The tester's eagerness to find high-risk defects.

b) The amount and type of risk an organization is willing to take to meet its strategic objectives.

c) The process of identifying all possible risks.

d) The tools used for risk management.

**Answer:** b

**Section 27: Ethical Considerations & Professional Conduct**

**130. If a tester discovers a critical security vulnerability just before a major release, what is the MOST ethical first step?**

a) Post about it on social media to warn users.

b) Immediately report it to the relevant stakeholders (e.g., test lead, project manager, development lead) following internal procedures.

c) Try to exploit the vulnerability to see its full impact without authorization.

d) Ignore it to avoid delaying the release.

**Answer:** b

**131. Which behavior is generally considered UNPROFESSIONAL for a tester?**

a) Asking clarifying questions about requirements.

b) Using a blaming tone when reporting defects or interacting with developers.

c) Providing objective and factual information in defect reports.

d) Collaborating with developers to understand and reproduce issues.

**Answer:** b

**132. Maintaining confidentiality of project information and test data is an example of:**

a) A good test design technique.

b) An ethical responsibility for a software tester.

c) A type of performance testing.

d) A defect management practice.

**Answer:** b

**Section 28: Communication & Collaboration for Testers**

**133. When reporting a defect, a tester should strive to be:**

a) Vague to allow developers flexibility in fixing.

b) Subjective and opinionated.

c) Clear, concise, objective, and provide enough information to reproduce the issue.

d) Overly technical, assuming developers will understand complex jargon.

**Answer:** c

**134. "Active Listening" as a communication skill for testers involves:**

a) Constantly interrupting to share one's own ideas.

b) Fully concentrating on what is being said, understanding the message, responding thoughtfully, and retaining the information.

c) Preparing a rebuttal while the other person is speaking.

d) Only hearing parts of the conversation.

**Answer:** b

**135. Why is collaboration between testers and developers important in an effective software development process?**

a) It helps testers assign blame for defects more easily.

b) It fosters a shared understanding of quality, helps in quicker defect resolution, and improves overall product quality.

c) It eliminates the need for a separate testing phase.

d) It is only important in Waterfall models.

**Answer:** b

**Section 29: Understanding Different Test Artifacts**

**136. A "Test Basis" refers to:**

a) The test environment setup.

b) The document or set of documents from which test cases are derived (e.g., requirements, design specifications, user stories).

c) The team of testers assigned to a project.

d) The automation framework used for testing.

**Answer:** b

**137. What is the primary difference between a Test Plan and a Test Case?**

a) A Test Plan details individual test steps, while a Test Case outlines the overall strategy.

b) A Test Plan outlines the strategy, scope, resources, and schedule, while a Test Case specifies inputs, actions, and expected results for a specific test.

c) Test Plans are only for managers, and Test Cases are only for junior testers.

d) There is no significant difference; the terms are interchangeable.

**Answer:** b

**Section 30: Software Development Lifecycle (SDLC) in Testing Context**

**138. The "Requirements Analysis" phase of the SDLC is crucial for testers because:**

a) It's when all coding is completed.

b) It provides the foundation for defining test conditions and designing test cases to ensure the system meets user needs.

c) It's when the test environment is set up.

d) It's primarily the responsibility of the database administrator.

**Answer:** b

**139. During which SDLC phase are most Unit Tests typically designed and executed?**

a) Requirements Gathering

b) Design

c) Implementation/Coding

d) Maintenance

**Answer:** c

**140. The "Deployment" phase in SDLC directly leads to which type of testing activity if issues are reported from production?**

a) Early static reviews

b) Production incident verification, potentially triggering hotfix testing or regression testing.

c) User story grooming

d) Initial component testing

**Answer:** b

**Section 31: Nuanced Scenarios & Specific Test Types (Continued)**

**141. You are testing an e-commerce "checkout" process. Which of the following represents the MOST critical end-to-end test scenario?**

a) Verifying the colour of the "Proceed to Payment" button.

b) Testing if a user can add an item to the cart, enter shipping/billing info, select a payment method, confirm the order, and receive an order confirmation.

c) Checking if the "Terms and Conditions" link opens correctly.

d) Testing the input validation for the zip code field in isolation.

**Answer:** b

**142. During an ad-hoc testing session on a social media application, you notice that uploading a very large image file causes the app to freeze indefinitely. This is likely an issue related to:**

a) User interface design.

b) Robustness or error handling under stress conditions.

c) Spell-checking functionality.

d) User registration process.

**Answer:** b

**143. When performing "Gorilla Testing" on a specific module, the tester:**

a) Lightly tests various parts of the module.

b) Focuses on intensively testing that one module with a variety of valid and invalid inputs, often pushing its limits.

c) Tests the module's interaction with other modules.

d) Only tests the happy path scenarios for that module.

**Answer:** b

**144. "Soak Testing" (Endurance Testing) is particularly important for applications that are expected to:**

a) Handle sudden spikes in user traffic.

b) Run continuously for long periods without degradation or failure.

c) Process a small number of complex transactions.

d) Have a highly intuitive user interface.

**Answer:** b

**145. You are testing a financial application's interest calculation feature. A good negative test case would be:**

a) Calculating interest for a valid principal amount and interest rate.

b) Attempting to calculate interest with a negative principal amount or an impossibly high interest rate.

c) Verifying the display format of the calculated interest.

d) Checking if the calculation is fast enough.

**Answer:** b

**Section 32: Non-Functional Testing Aspects (Reliability, Maintainability)**

**146. Reliability of software refers to its ability to:**

a) Perform its specified functions correctly under stated conditions for a specified period.

b) Be easily modified or corrected.

c) Be easily understood and used by end-users.

d) Protect information from unauthorized access.

**Answer:** a

**147. Mean Time Between Failures (MTBF) is a metric often associated with assessing:**

a) Software Usability

b) Software Maintainability

c) Software Reliability

d) Software Security

**Answer:** c

**148. Maintainability of software is concerned with the ease with which:**

a) Users can learn to operate the software.

b) The software can be modified to correct defects, improve performance, or adapt to a changed environment.

c) The software recovers from hardware failures.

d) The software protects data.

**Answer:** b

**149. Which factor contributes positively to software maintainability from a testing perspective?**

a) Highly complex and undocumented code.

b) Well-documented test cases and clear defect reports that help developers understand and fix issues.

c) Lack of version control for test assets.

d) Frequent changes to the test environment without notification.

**Answer:** b

**150. From a tester's viewpoint, "Portability" as a non-functional characteristic refers to the ease with which software can be:**

a) Understood by new testers.

b) Transferred from one hardware or software environment to another.

c) Accessed by users with disabilities.

d) Restored after a system crash.

**Answer:** b

**Section 33: Psychology of Testing & Cognitive Biases**

**151. The "Availability Heuristic" might lead a tester to:**

a) Underestimate the likelihood of a defect type they haven't encountered recently.

b) Overestimate the likelihood of a defect type they recently found or that was particularly memorable.

c) Always follow the test script without deviation.

d) Prefer testing features they are most familiar with.

**Answer:** b

**152. "Anchoring Bias" in test estimation could occur when:**

a) A tester bases their estimate heavily on the first piece of information received (e.g., an initial, perhaps inaccurate, suggestion), even if later information contradicts it.

b) A tester meticulously breaks down the work before estimating.

c) Multiple testers provide estimates independently.

d) Estimates are frequently revised based on new data.

**Answer:** a

**153. To counteract cognitive biases in testing, a good practice is to:**

a) Always trust one's first instinct without question.

b) Work in complete isolation to avoid external influence.

c) Encourage diverse perspectives, seek peer reviews of test cases, and be aware of common biases.

d) Only use automated testing tools.

**Answer:** c

**Section 34: Test Process Improvement Models (Beyond Basics)**

**154. CMMI (Capability Maturity Model Integration) provides a framework that can be used to improve an organization's processes, including those related to:**

a) Only marketing and sales.

b) Product development, service delivery, and acquisition, which encompasses testing activities as part of engineering.

c) Only financial accounting.

d) Human resource management exclusively.

**Answer:** b

**155. A key principle underlying most process improvement models like TMMi or CMMI is:**

a) Rapid, unstructured change is always best.

b) Continuous improvement through defined processes, measurement, and feedback.

c) Adherence to the initial plan regardless of changing circumstances.

d) Minimizing all documentation.

**Answer:** b

**Section 35: Formal Review Roles & Responsibilities**

**156. In a formal inspection process, the "Moderator" is primarily responsible for:**

a) Writing the document being reviewed.

b) Finding all the defects in the document.

c) Planning the inspection, leading the meeting, ensuring the process is followed, and mediating discussions.

d) Fixing the defects identified during the inspection.

**Answer:** c

**157. The "Author" in a formal review process is the person who:**

a) Leads the review meeting.

b) Created the work product (document, code) being reviewed and is responsible for fixing defects found in it.

c) Takes notes and records defects during the review meeting.

d) Decides whether the work product passes or fails the review.

**Answer:** b

**158. A "Reviewer" in a formal inspection has the responsibility to:**

a) Only attend the review meeting without prior preparation.

b) Prepare for the review by examining the work product, identify potential defects, and actively participate in the review meeting.

c) Manage the schedule and logistics for all reviews.

d) Approve the budget for the review process.

**Answer:** b

**Section 36: Advanced Test Metrics & Analysis**

**159. "Defect Detection Percentage (DDP)" or "Defect Removal Efficiency (DRE)" measures:**

a) The total number of defects found in the project.

b) The percentage of defects found by the test team out of the total defects present (including those found later by users).

c) The average time taken to fix a defect.

d) The cost of finding each defect.

**Answer:** b (A higher DDP/DRE for internal testing phases is generally desirable)

**160. "Test Effectiveness" can be qualitatively assessed by considering:**

a) Only the number of test cases executed.

b) The ability of the test set to find critical defects and cover important functionalities, relative to the effort expended.

c) The speed at which testers can write test cases.

d) The number of testers in the team.

**Answer:** b

**161. If the number of defects found in later stages of testing (e.g., System Test, UAT) is consistently high, it might indicate issues in:**

a) The user documentation.

b) The effectiveness of earlier testing phases (e.g., Unit Test, Integration Test) or review processes.

c) The performance of the test environment.

d) The tools used for defect tracking.

**Answer:** b

**Section 37: Context-Specific Testing (Conceptual)**

**162. When testing an Embedded System (e.g., firmware in a medical device), a unique challenge often faced by manual testers is:**

a) The user interface is always a standard web browser.

b) Interaction with hardware components, real-time constraints, and potentially limited observability or direct input mechanisms.

c) Requirements are always perfectly clear and unchanging.

d) Performance is never a concern.

**Answer:** b

**163.Testing a Data Warehouse (DW) or Business Intelligence (BI) system often involves verifying:**

a) Only the visual appeal of the reports.

b) Data transformations (ETL processes), data accuracy, report correctness, and performance of queries.

c) The security of the operating system hosting the DW.

d) The transaction speed of the source operational systems.

**Answer:** b

**164. Game testing, from a manual perspective, often heavily relies on:**

a) Strict adherence to pre-written, step-by-step test cases for all scenarios.

b) Exploratory testing, playability testing, checking game mechanics, balance, and user experience.

c) Only testing the installation process.

d) Verifying the game's source code line by line.

**Answer:** b

**Section 38: Future Trends & Evolution in Manual Testing**

**165. Even with the rise of test automation, manual testing remains crucial for:**

a) Running repetitive regression tests.

b) Exploratory testing, usability testing, ad-hoc testing, and scenarios requiring human intuition and subjective evaluation.

c) Generating large volumes of test data.

d) Performing load testing on APIs.

**Answer:** b

**166. A potential future trend for manual testers might involve increased focus on:**

a) Writing complex automation scripts in multiple programming languages.

b) Deeper domain expertise, AI-assisted testing (e.g., interpreting results, suggesting test areas), and more sophisticated exploratory testing techniques.

c) Managing server infrastructure for test environments.

d) Performing database administration tasks.

**Answer:** b

**167. How can AI and Machine Learning potentially assist manual testers in the future?**

a) By completely replacing the need for human testers.

b) By helping to predict high-risk areas, optimize test case selection, identify anomalies in test results, or generate exploratory test ideas.

c) By writing all user stories and acceptance criteria.

d) By designing the application's user interface.

**Answer:** b

**Section 39: Test Closure Activities**

**168. Which of the following is a key activity during the "Test Closure" phase?**

a) Designing new test cases for the next release.

b) Ensuring all test deliverables (e.g., test summary report, test logs) are archived and lessons learned are documented.

c) Executing smoke tests for the first time.

d) Setting up the test environment.

**Answer:** b

**169. A "Lessons Learned" session held during test closure aims to:**

a) Assign blame for project delays.

b) Identify what went well, what went wrong, and what could be improved in future test efforts and projects.

c) Plan the test activities for the next project.

d) Only discuss the defects that were not fixed.

**Answer:** b

**170. Handing over testware (test cases, test data, environment details) to a maintenance team is often part of:**

a) Test Planning

b) Test Execution

c) Test Closure

d) Requirements Review

**Answer:** c

**Section 40: Understanding Test Levels & SDLC Relationships**

**171. Acceptance Test Driven Development (ATDD) involves:**

a) Developers writing acceptance tests after the code is complete.

b) Collaboration between developers, testers, and business stakeholders to define acceptance tests before development begins, guiding implementation.

c) Testers performing acceptance testing only at the very end of the project.

d) Automating all acceptance tests using UI-based tools.

**Answer:** b

**172. If a critical defect is found during User Acceptance Testing (UAT), what is the most likely immediate impact on the project**

a) The project is immediately cancelled.

b) The defect is ignored if the user is willing to work around it.

c) It may lead to a delay in release, require a hotfix, and further re-testing.

d) Testers are asked to write more unit tests.

**Answer:** c

**173. The output of the System Testing phase often serves as an input criterion for starting:**

a) Unit Testing

b) Integration Testing

c) User Acceptance Testing (UAT)

d) Requirements Gathering

**Answer:** c

**174. Sanity Testing is typically performed:**

a) Before starting any other type of testing.

b) After a build is received, to quickly check if major functionalities are working and if the build is stable enough for further, more detailed testing.

c) Only on the final release candidate.

d) By the end-users in their environment.

**Answer:** b

**175. What is the relationship between Quality Assurance (QA), Quality Control (QC), and Testing?**

a) QA and QC are synonyms for Testing.

b) Testing is a QC activity; QA is a broader set of processes to ensure overall quality throughout the lifecycle, including defining QC measures.

c) QC is a process that ensures QA is done correctly, and Testing is separate.

d) QA is about finding defects, QC is about preventing them, and Testing is about fixing them.

**Answer:** b

**Section 41: Tool-Related Concepts (Manual Testing Perspective)**

**176. A "Defect Tracking Tool" (e.g., Jira, Bugzilla) primarily helps manual testers to:**

a) Automatically execute test cases.

b) Write and design test cases.

c) Report, manage, track the status of, and communicate information about defects.

d) Generate test data.

**Answer:** c

**177. When a manual tester uses a "Test Management Tool" (e.g., TestRail, Zephyr, QMetry), a key benefit is:**

a) It writes the application code automatically.

b) Centralized storage for test cases, mapping to requirements, planning test execution cycles, and tracking results.

c) It performs security scans on the application.

d) It eliminates the need for any human testers.

**Answer:** b

**178. For manual exploratory testing, which type of tool might be most beneficial during a session?**

a) A load testing tool.

b) A screen capture and annotation tool (for documenting issues quickly).

c) A unit testing framework.

d) A code compiler.

**Answer:** b

**Section 42: Variations in Defect Reporting & Analysis**

**179. When is it appropriate for a tester to include a "Suggested Fix" or "Workaround" in a defect report?**

a) Never, as it's the developer's job.

b) Always, to show they have fully analyzed the issue.

c) Optionally, if they have a clear idea that might help, but it should be clearly marked as a suggestion and not a demand.

d) Only if the defect is of low severity.

**Answer:** c

**180. If a defect is intermittent (occurs sporadically), what is crucial for the tester to include in the defect report?**

a) A guarantee that it will happen every time.

b) As much detail as possible about the conditions under which it was observed, frequency, and any potential contributing factors or patterns.

c) A statement that it's probably not a real defect.

d) Only the expected result.

**Answer:** b

**181. "Defect Clustering" implies that if a tester finds several defects in one module, they should:**

a) Stop testing that module as it's clearly unstable.

b) Potentially focus more testing effort on that module, as there might be more underlying issues.

c) Assume other modules are defect-free.

d) Immediately report the module to management for a complete rewrite.

**Answer:** b

**Section 43: Advanced Non-Functional Aspects & Usability Heuristics**

**182. "Recoverability" as a non-functional characteristic refers to the system's ability to:**

a) Be easily learned by new users.

b) Re-establish a specified level of performance and recover data directly affected in case of a failure.

c) Be easily modified by developers.

d) Run on different operating systems.

**Answer:** b

**183. Nielsen's Usability Heuristic "Visibility of system status" suggests that the system should:**

a) Hide all internal workings from the user.

b) Always keep users informed about what is going on, through appropriate feedback within reasonable time.

c) Only show error messages.

d) Require users to guess the current state.

**Answer:** b

**184. Nielsen's Usability Heuristic "Error prevention" is generally considered better than:**

a) Good error messages.

b) Allowing users to undo actions.

c) Fast system performance.

d) Extensive user documentation.

**Answer:** a (Preventing errors in the first place is better than just reporting them well after they occur)

**Section 44: Test Data Management Nuances**

**185. The challenge of "Test Data Staleness" refers to:**

a) Test data that is too new and doesn't reflect historical scenarios.

b) Test data that is no longer representative of the current production data environment or application state, leading to irrelevant test results.

c) Test data that is perfectly organized.

d) Using the same test data for every test cycle without variation.

**Answer:** b

**186. When testing an application that interacts with third-party services, a common test data challenge is:**

a) The third-party service is always perfectly stable.

b) Ensuring appropriate and consistent test data or mocked responses from the third-party service for reliable testing.

c) The third-party service provides all necessary test data.

d) Third-party services never have defects.

**Answer:** b

**Section 45: Reviewing Different Types of Requirements**

**187. When reviewing "Functional Requirements," a tester primarily looks for:**

a) The speed and efficiency of the system.

b) Clarity, completeness, testability, and unambiguity of what the system should \*do\*.

c) The aesthetic appeal of the user interface.

d) The programming language to be used.

**Answer:** b

**188. When reviewing "Non-Functional Requirements" (e.g., performance, security), a tester would check if they are:**

a) Vague and open to interpretation.

b) Specific, measurable, achievable, relevant, and time-bound (SMART), making them testable.

c) Identical to functional requirements.

d) Only focused on the look and feel.

**Answer:** b

**189. A common issue found when reviewing User Stories as a form of requirements is:**

a) They are always too detailed.

b) Lack of clear, testable Acceptance Criteria.

c) They focus too much on technical implementation.

d) They are only written by testers.

**Answer:** b

**Section 46: Core Testing Principles - Final Check**

**190. "Early Testing" principle suggests that:**

a) Testing should only start after all code is written.

b) Integrating testing activities as early as possible in the lifecycle (e.g., reviewing requirements, static analysis) is more cost-effective.

c) Only senior testers should perform early tests.

d) Early testing is less important than end-stage testing.

**Answer:** b

191. The "Absence of Errors Fallacy" means:

a) If no errors are found, the software is perfect.

b) Finding and fixing defects does not guarantee the system will be useful or meet user needs if it's fundamentally flawed or built for the wrong requirements.

c) Testers should strive to find every single error.

d) Automated testing finds all errors.

**Answer:** b

**Section 47: Test Levels & Their Focus (Reiteration & Nuance)**

**192. While Unit Testing focuses on individual components, Integration Testing primarily verifies:**

a) The usability of the entire system.

b) The interfaces and interactions between integrated components or systems.

c) The system's performance under load.

d) Compliance with business requirements by end-users.

**Answer:** b

**193. The scope of System Testing is broader than Integration Testing because it:**

a) Only tests individual modules.

b) Tests the fully integrated system against overall system requirements, including functional and non-functional aspects, often in a more production-like environment.

c) Is only performed by developers.

d) Focuses solely on the user interface.

**Answer:** b

**Section 48: Test Execution & Reporting Fundamentals**

**194. If a test case is "Blocked," it means:**

a) The test case has passed successfully.

b) The test case has failed.

c) The test case cannot be executed, usually due to an external dependency or a preceding defect preventing further progress.

d) The test case is not applicable to the current build.

**Answer:** c

**195. A daily test execution status report might include:**

a) Only the names of the testers.

b) Number of test cases planned, executed, passed, failed, blocked, and any critical issues found.

c) Detailed root cause analysis for every failed test.

d) The complete source code of the application.

**Answer:** b

**Section 49: Independent Testing Considerations**

**196. A key advantage of having an independent test team is:**

a) They are always cheaper than developers testing their own code.

b) They can bring an objective viewpoint, avoid developer bias (confirmation bias), and may have specialized testing skills.

c) They understand the code better than the developers.

d) They eliminate the need for developers to do any testing.

**Answer:** b

**197. A potential disadvantage of a highly independent test team if communication is poor can be:**

a) They find too many defects.

b) Isolation from the development team, leading to misunderstandings or delays in information flow.

c) They always agree with the developers.

d) They are less skilled than developers.

**Answer:** b

**Section 50: Final Thoughts on Manual Testing's Role**

**198. Even in highly automated environments, manual testing often plays a vital role in:**

a) Initial feature validation, usability checks, exploratory testing, and verifying complex end-to-end scenarios that are difficult or too costly to automate comprehensively.

b) Executing all unit tests.

c) Managing the CI/CD pipeline.

d) Writing database schemas.

**Answer:** a

**199. A software tester's primary goal should be to:**

a) Prove the software works perfectly.

b) Find every single defect in the software.

c) Provide stakeholders with information about the quality of the product and the risks associated with its release, by finding and reporting defects.

d) Write as many test cases as possible, regardless of their effectiveness.

**Answer:** c

**200. Continuous learning and adapting to new technologies, methodologies, and testing techniques is crucial for a manual tester because:**

a) It guarantees a higher salary.

b) The software industry is constantly evolving, and staying relevant requires updating skills and knowledge to meet new challenges and needs.

c) It means they can stop manual testing and only do automation.

d) It's only important for junior testers.

**Answer:** b