ISTQB Test Analyst Exam Prep

Free Practice Test — 30 Real Exam-Style Questions

with full answer key & explanations

Practice Questions

Try all 30 first, then check the answer key at the back.

1. Which white-box test technique focuses on ensuring that each executable statement in the code is executed at least once during testing?

A. Modified condition/decision coverage
B. Decision testing
C. Multiple condition testing
D. Statement testing

2. Which of the following coverage levels provides the strongest logical test coverage?

A. Decision coverage
B. Statement coverage
C. Modified condition/decision coverage (MC/DC)
D. Branch coverage

3. In the context of white-box testing, what is the main advantage of decision testing over statement testing?

A. It executes fewer test cases
B. It ensures all branches or decision outcomes are tested
C. It can be completely automated
D. It does not require code access

4. A team is using multiple condition testing on a complex financial application. What is this technique primarily designed to evaluate?

A. All possible combinations of condition outcomes within a decision
B. The performance impact of each function
C. The memory usage of conditional statements
D. The time complexity of algorithms

5. Which metric would a technical test analyst use to measure decision coverage in a codebase?

A. Average execution time for each statement
B. Number of functions called during execution
C. Total count of conditions in the code
D. Percentage of decision outcomes exercised by tests

6. Which white-box technique would be MOST appropriate for testing critical aerospace software where safety is the primary concern?

A. Path coverage
B. Statement coverage
C. Modified condition/decision coverage (MC/DC)
D. Linear code sequence and jump (LCSAJ) coverage

7. When designing API tests using white-box techniques, which of the following would be a primary focus area?

A. User interface aesthetics
B. Parameter combinations and boundary values
C. Database optimization
D. Network latency measurement

8. If a technical test analyst achieves 100% statement coverage but only 70% decision coverage, what does this indicate about the test suite?

A. Some decision outcomes have not been tested
B. The code contains unreachable statements
C. The test suite is optimally designed
D. More test cases are needed than theoretically required

9. When using condition coverage for testing, what key aspect of the code is being verified?

A. The code executes within time constraints
B. Each statement executes at least once
C. All possible paths through the code are exercised
D. Each Boolean sub-expression evaluates to both true and false

10. What is a key consideration when designing test cases for multiple condition coverage?

A. The technique requires specialized testing tools
B. The test cases must be executed in a specific order
C. The number of test cases grows exponentially with the number of conditions
D. Each test case must cover exactly one condition

11. For which of the following software components would statement coverage likely be INSUFFICIENT as the only white-box testing technique?

A. Simple getter and setter methods
B. Complex error handling routines
C. Linear data transformation algorithms
D. Basic initialization sequences

12. What is the primary difference between decision testing and condition testing?

A. Decision testing focuses on the overall decision result, while condition testing examines individual Boolean sub-expressions
B. Decision testing requires code access while condition testing does not
C. Condition testing is only used for loops, while decision testing is for if-statements
D. Decision testing can be automated but condition testing must be done manually

13. Which statement about API testing using white-box techniques is correct?

A. It cannot be used for RESTful web services
B. It only requires black-box access to the API endpoints
C. It focuses exclusively on performance aspects of the API
D. It involves testing internal structures, data flow, and exception handling

14. A developer has written a complex function with nested if-statements and logical operators. Which white-box technique would BEST help detect logical defects in the conditions?

A. Path coverage
B. Statement coverage
C. Modified condition/decision coverage (MC/DC)
D. Loop coverage

15. What is the key principle of Modified Condition/Decision Coverage (MC/DC)?

A. Every possible combination of conditions must be tested
B. Each condition must independently affect the decision outcome
C. All loops must be executed exactly once
D. Each statement must be executed with different input values

16. When designing white-box tests for an API that processes financial transactions, which coverage criterion would be MOST effective at identifying logic defects in the transaction validation code?

A. Multiple condition coverage
B. Statement coverage
C. Loop coverage
D. Path coverage

17. What is a key limitation of statement coverage as a white-box testing technique?

A. It generates too many test cases to be practical
B. It requires special compiler support
C. It can only be applied to object-oriented languages
D. It cannot detect missing paths in the code

18. A code review reveals a complex decision with four conditions connected by AND and OR operators. What is the MINIMUM number of test cases needed for MC/DC coverage of this decision?

A. 8 test cases
B. 4 test cases
C. 5 test cases
D. 16 test cases

19. Which of the following statements about white-box testing techniques is TRUE?

A. They are generally more time-efficient than black-box techniques
B. They complement black-box techniques by focusing on code structure rather than specifications
C. They eliminate the need for black-box testing
D. They can only be performed by developers, not testers

20. What is the primary benefit of using decision testing for API validation?

A. Ensuring all logical branches in the API implementation are exercised
B. Improving API documentation quality
C. Reducing API response time
D. Eliminating the need for API security testing

21. Which of the following is the primary goal of conducting performance efficiency tests from a technical test analyst perspective?

A. To check if the system documentation is complete
B. To verify the functional correctness of the system
C. To ensure the system's user interface is intuitive
D. To determine if the system's response time, throughput, and resource utilization meet specified requirements

22. When conducting security testing, which approach is MOST appropriate for a technical test analyst to use when testing for SQL injection vulnerabilities?

A. Functional testing of database queries
B. Static code review of database schemas
C. Dynamic analysis with malicious SQL inputs
D. User acceptance testing with standard inputs

23. Which reliability metric would be MOST useful for a technical test analyst to evaluate when testing a system that must operate continuously without failure?

A. Code coverage percentage
B. Mean Time Between Failures (MTBF)
C. Number of test cases executed
D. User satisfaction score

24. A technical test analyst is asked to evaluate the maintainability of a system. Which metric would provide the MOST value for this assessment?

A. Cyclomatic complexity of the code
B. Number of function points
C. User interface design consistency
D. Database query performance

25. During portability testing, a technical test analyst discovers that a software application fails when deployed to a different operating system. This is an example of a defect in which quality characteristic?

A. Reliability
B. Functionality
C. Usability
D. Portability

26. Which of the following would be MOST appropriate for a technical test analyst to include in a security test plan?

A. Performance testing under normal load conditions
B. Usability testing with end users
C. Penetration testing to identify exploitable vulnerabilities
D. Functional testing of business requirements

27. When testing the performance efficiency of a web application, which of the following metrics would be LEAST relevant for a technical test analyst to measure?

A. Response time under various load conditions
B. Brand recognition
C. CPU utilization during peak usage
D. Memory consumption over time

28. A technical test analyst is evaluating the compatibility of a web application across different browsers. Which testing approach would be MOST effective?

A. Cross-browser testing with automated tools
B. Code reviews of HTML and CSS
C. Manual testing on a single browser
D. Performance testing of browser rendering speed

29. During reliability testing of a financial system, which of the following would be MOST important for a technical test analyst to test?

A. The responsiveness of the help documentation
B. The visual appearance of the user interface
C. The system's ability to support multiple languages
D. The system's ability to recover from failures without data loss

30. Which of the following would be the MOST appropriate approach for a technical test analyst to test the security of an authentication mechanism?

A. Evaluating the visual design of the login form
B. Measuring the response time of the login page
C. Testing for resistance to brute force attacks
D. Testing the browser compatibility of the login page

Answer Key & Explanations

1. D — Statement testing
Statement testing (also called statement coverage) aims to ensure each executable statement in the code is executed at least once during testing. It's the most basic form of white-box testing coverage.

2. C — Modified condition/decision coverage (MC/DC)
Modified condition/decision coverage (MC/DC) requires that each condition in a decision independently affects the outcome of the decision, making it stronger than the other options in detecting logical defects.

3. B — It ensures all branches or decision outcomes are tested
Decision testing ensures that all branches or decision outcomes are tested, which provides better test coverage than statement testing by ensuring that both true and false paths in conditions are executed.

4. A — All possible combinations of condition outcomes within a decision
Multiple condition testing evaluates all possible combinations of condition outcomes within a decision, making it especially useful for testing complex logical expressions where each combination could lead to different behavior.

5. D — Percentage of decision outcomes exercised by tests
Decision coverage is measured by calculating the percentage of decision outcomes that have been exercised by test cases, determining how many of the possible true/false paths have been tested.

6. C — Modified condition/decision coverage (MC/DC)
Modified condition/decision coverage (MC/DC) is often required for safety-critical systems like aerospace software because it ensures each condition in a decision independently affects the outcome, providing thorough testing of complex logical expressions.

7. B — Parameter combinations and boundary values
Parameter combinations and boundary values are critical for API testing because they help verify the API handles various input combinations correctly, including edge cases and invalid inputs.

8. A — Some decision outcomes have not been tested
This situation indicates that some decision outcomes (typically false conditions) have not been tested, despite executing all statements. This is possible because statement coverage does not ensure all decision paths are exercised.

9. D — Each Boolean sub-expression evaluates to both true and false
Condition coverage focuses on ensuring each Boolean sub-expression (condition) evaluates to both true and false, which helps identify defects in individual conditions that might be masked when testing only the overall decision outcome.

10. C — The number of test cases grows exponentially with the number of conditions
Multiple condition coverage requires testing all possible combinations of conditions, so the number of test

cases grows exponentially with the number of conditions, making test case design more complex for decisions with many conditions.

11. B — Complex error handling routines
Complex error handling routines contain multiple decision points and exception paths that statement coverage alone would not adequately test, as it doesn't ensure all decision outcomes are exercised.

12. A — Decision testing focuses on the overall decision result, while condition testing examines individual Boolean sub-expressions
Decision testing focuses on the overall decision result (true/false), while condition testing focuses on the individual Boolean sub-expressions (conditions) within a decision, providing more granular coverage.

13. D — It involves testing internal structures, data flow, and exception handling
API testing using white-box techniques typically focuses on testing internal structures, data flow, and exception handling by accessing and analyzing the underlying code implementation rather than just the interface behavior.

14. C — Modified condition/decision coverage (MC/DC)
Modified condition/decision coverage (MC/DC) is specifically designed to detect logical defects by ensuring each condition independently affects the decision outcome, which is ideal for complex nested conditions with logical operators.

15. B — Each condition must independently affect the decision outcome
The key principle of MC/DC is that each condition must independently affect the decision outcome, proving that each condition has a distinct influence on the result when all other conditions remain constant.

16. A — Multiple condition coverage
Multiple condition coverage would be most effective as it tests all combinations of conditions in the validation logic, ensuring that all possible validation scenarios are verified, which is critical for financial transaction processing.

17. D — It cannot detect missing paths in the code
Statement coverage cannot detect missing paths in the code, such as error handling paths that should exist but don't, since it only measures execution of existing code statements.

18. C — 5 test cases
For MC/DC coverage of a decision with n conditions, the minimum number of test cases required is n+1. With four conditions, at least 5 test cases are needed to ensure each condition independently affects the outcome.

19. B — They complement black-box techniques by focusing on code structure rather than specifications
White-box testing techniques complement black-box techniques by focusing on code structure rather than specifications, providing different perspectives and often finding different types of defects, resulting in more comprehensive testing when used together.

20. A — Ensuring all logical branches in the API implementation are exercised
Decision testing for API validation ensures that all logical branches in the API implementation are exercised, including alternate paths and error handling, which helps verify the API behaves correctly under various conditions and edge cases.

21. D — To determine if the system's response time, throughput, and resource utilization meet specified requirements
Performance efficiency testing is conducted to evaluate how the system performs under various conditions, focusing on response times, throughput, and resource utilization as key metrics rather than functional behavior.

22. C — Dynamic analysis with malicious SQL inputs
Dynamic analysis involves executing the application with malicious SQL inputs to see if they are executed by the database, which is the most effective way to detect actual SQL injection vulnerabilities.

23. B — Mean Time Between Failures (MTBF)
Mean Time Between Failures (MTBF) specifically measures the average time between system failures, which is crucial for systems requiring continuous operation.

24. A — Cyclomatic complexity of the code
Cyclomatic complexity measures the number of independent paths through code, providing insight into how difficult the code is to understand, test, and maintain. Higher complexity generally indicates lower maintainability.

25. D — Portability
Portability refers to the ability of software to be transferred from one environment to another. Failure when deployed to a different operating system directly impacts this quality characteristic.

26. C — Penetration testing to identify exploitable vulnerabilities
Penetration testing is specifically designed to identify security vulnerabilities by simulating attacks, making it a critical component of security test planning.

27. B — Brand recognition
Brand recognition is a marketing metric unrelated to the technical performance efficiency of a web application, unlike the other options which measure technical aspects of performance.

28. A — Cross-browser testing with automated tools
Cross-browser testing with automated tools is most effective for compatibility testing across browsers as it can systematically verify application behavior and appearance across multiple browser environments efficiently.

29. D — The system's ability to recover from failures without data loss
For financial systems, recovery from failures without data loss is critical to ensure transaction integrity and prevent financial discrepancies.

30. C — Testing for resistance to brute force attacks
Authentication mechanisms need to be tested for resistance to brute force attacks, which directly tests the security strength of the password protection and account lockout features.

Ready to pass?

Unlock the full ISTQB Test Analyst Exam Prep bank, every explanation, and unlimited timed mock exams.

Scan to start practising
https://certs.theorypractice.app/istqbta

Watch the full video walkthrough on YouTube @CertsQuizPrep