DUMPSARENA

ISTQB Certified Tester Advanced Level - Technical Test Analyst

iSQI CTAL-TTA

Version Demo

Total Demo Questions: 6

Total Premium Questions: 65

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, The Technical Test Analyst's Tasks in the Risk-Based Testing Approach	8
Topic 2, White-Box Test Techniques	20
Topic 3, Static and Dynamic Analysis	7
Topic 4, Quality Characteristics for Technical Testing	8
Topic 5, Reviews	9
Topic 6, Test Tools and Automation	10
Topic 7, Mix Questions	3
Total	65

QUESTION NO: 1

When developing a checklist to support code reviews, which two questions should be included? Select exactly two answers.

A. Are any magic numbers (numeric literals) used, other than 0 or 1?

B. Can each item be implemented with the techniques, tools, and resources available?

C. Is it possible during acceptance testing to verify whether the item has been satisfied?

D. Is the item specified in an exact, unambiguous way?

E. Is there any logic containing no statements and no explanation as to why it is empty?

ANSWER: A E

Explanation:

“Are any magic numbers (numeric literals) used, other than 0 or 1?” is an appropriate code-review checklist question because unexplained literal values obscure the purpose of code and make later maintenance riskier. A reviewer should identify whether a value represents a meaningful business rule, limit, timeout, status, or conversion factor and, where appropriate, require it to be replaced with a named constant or otherwise clearly documented. This improves readability, traceability, and the ability to safely change the value in the future.

“Is there any logic containing no statements and no explanation as to why it is empty?” is also appropriate because an empty branch, loop body, exception handler, or conditional block can indicate omitted behavior, a defect, or code whose intent is not evident. Where deliberately empty logic is needed, an explicit explanation lets reviewers distinguish intentional design from an accidental omission. Technical reviews use such concrete, code-focused checks to detect maintainability and implementation-quality defects early. The ISTQB Technical Test Analyst syllabus addresses static testing and the technical perspective needed when assessing code quality; see ISTQB Certified Tester Advanced Level – Technical Test Analyst. Guidance on maintainable, readable code is also reflected in Google Java Style Guide.

QUESTION NO: 2

To investigate the third identified risk, a performance test is proposed. Which measure should be used to quantify the performance situation and provide a basis for defining performance test exit criteria?

Number of correct answers: 1

A. Defect analysis time

B. Change parameter success ratio

C. Number of errors made by the user

D. Time to perform the function

ANSWER: D

Explanation:

Time to perform the function is the appropriate measure because it directly quantifies the elapsed time required for a user-facing or system function to complete under defined test conditions. This is a performance characteristic commonly expressed as response time, transaction time, or completion time. It provides an objective, repeatable value that can be collected during execution, compared with a required service level, and used to determine whether the performance risk has been adequately addressed.

For example, an exit criterion can state that a specified function must complete within an agreed maximum duration for a defined percentage of transactions, workload level, and environment. To make the criterion meaningful, the test team should define the start and end points of the timing measurement, the relevant workload profile, any percentile target, and the observation period. This turns a general concern about performance into measurable evidence for stakeholders and supports a clear test completion decision. The ISTQB Technical Test Analyst certification includes performance testing as a

technical testing area; see the ISTQB Technical Test Analyst certification page. General definitions of performance efficiency and time behaviour are also provided by the ISO/IEC 25010 quality model.

QUESTION NO: 3

A development manager asks you to define suitable test-coverage-related entry criteria for the component testing phase. Which two of the following are appropriate entry criteria for beginning component testing?

Number of correct answers: 2

A. 100% statement coverage

B. No critical outstanding defects

C. Test log available

D. Code review completed

E. Static analysis shows no major violations

ANSWER: D E

Explanation:

Code review completed is an appropriate entry criterion because component testing is most effective when the code has already passed a basic static quality gate. A completed review provides evidence that peers have examined the implementation for defects, maintainability concerns, and conformance to agreed coding practices before dynamic tests consume time investigating easily preventable issues. It also helps ensure that the component is sufficiently stable and understandable to test effectively.

Static analysis shows no major violations is also an appropriate entry criterion. Static-analysis tools can identify important code-quality and reliability risks without executing the software, such as serious complexity concerns, unsafe constructs, likely null dereferences, security-relevant patterns, and violations of defined coding rules. Requiring the absence of major violations establishes a practical readiness threshold: the component can enter dynamic testing once significant issues identified through static techniques have been resolved or formally accepted. This supports the ISTQB principle that static testing and dynamic testing are complementary and that entry criteria define conditions that should be fulfilled before a test activity starts. See the ISTQB Glossary entry for entry criteria and the ISTQB Advanced Level Technical Test Analyst certification page.

QUESTION NO: 4

Using the state diagram shown above, how many test cases are required to achieve 1-switch coverage?

Select one answer.

A. 5

B. 7

C. 8

D. 9

ANSWER: C

Explanation:

The correct answer is **8**. In state-transition testing, 1-switch coverage requires tests to cover every feasible pair of consecutive transitions. It is also commonly called transition-pair coverage: rather than checking each individual transition in isolation, the test suite must exercise a transition followed immediately by each valid subsequent transition from the reached state. To determine the number of required test cases, the feasible two-transition sequences in the supplied state diagram

are identified and then arranged into executable paths beginning from the applicable initial state. For this diagram, eight distinct test paths are required to ensure that every feasible transition pair is exercised at least once. A single test case may cover more than one transition pair when its path contains three or more transitions, but the paths cannot be combined sufficiently to reduce the complete set below eight while preserving the required coverage. This criterion provides stronger confidence in state-dependent behavior because it checks that transitions work correctly in sequence, including the effect of arriving in a state through a particular preceding transition. ISTQB describes state-transition testing and transition coverage within its test-technique guidance; see the ISTQB Technical Test Analyst certification page and the ISTQB glossary entry for transition coverage.

QUESTION NO: 5

Consider the following code fragment:

```
if (a > b) and (b > c) then
  b = (a + c) / 2
endif
```

In each input triple, the first value represents `a`, the second represents `b`, and the third represents `c`.

Which set of test inputs achieves 100% decision coverage for this code fragment using the minimum number of test cases?

A. (5, 3, 2)

B. (5, 3, 2); (5, 4, 0)

C. (5, 4, 0); (4, 5, 0)

D. (4, 5, 0); (5, 4, 5)

ANSWER: C

Explanation:

The input set **(5, 4, 0); (4, 5, 0)** achieves complete decision coverage with the minimum of two tests. Decision coverage requires that the decision formed by the complete condition `(a > b) and (b > c)` produces both possible outcomes: true and false. In the first test, `a = 5`, `b = 4`, and `c = 0`. Both comparisons are true: `5 > 4` and `4 > 0`. Therefore, the overall decision is true and the assignment statement is executed. In the second test, `a = 4`, `b = 5`, and `c = 0`. The comparison `4 > 5` is false, so the complete conjunction is false and the assignment is not executed. Thus, the two tests exercise both outcomes of the sole decision, while no single test can produce both outcomes during one execution. This follows the ISTQB definition of decision coverage, which measures whether each decision outcome has been exercised. See the ISTQB Glossary definition of decision coverage and the ISTQB Technical Test Analyst certification overview.

QUESTION NO: 6

You need to select tools to support performance testing. Which **two** types of tools should you recommend?

Number of correct answers: 2

A. Test management tool

B. Load testing tool

C. Static analysis tool

D. Monitoring tool

E. Test oracle tool

ANSWER: B D

Explanation:

Load testing tool and **Monitoring tool** are the appropriate recommendations because they support complementary, essential parts of performance testing. A load testing tool generates a defined workload, such as concurrent virtual users, requests, transactions, and data volumes. This enables the test team to exercise the system under expected or challenging load profiles and to collect timing, throughput, error-rate, and responsiveness measures. It is therefore central to executing performance tests in a controlled and repeatable way.

A monitoring tool observes the system and its environment while the workload is running. It can collect resource and operational measurements such as processor utilization, memory consumption, disk and network activity, database behavior, application-server metrics, and service availability. Correlating these measurements with the workload and response-time results helps the Technical Test Analyst identify bottlenecks, determine likely causes of degraded performance, and provide meaningful diagnostic information to stakeholders and development teams.

Together, these tools allow the team both to create the demand placed on the system and to understand how the system behaves internally under that demand. This aligns with the Technical Test Analyst focus on performance efficiency testing and on using suitable tools for workload generation and system observation. See the ISTQB Technical Test Analyst certification page and the ISTQB Glossary.