Last name, first name: _____

Company address: _____

_____

_____

Phone: _____

Fax: _____

E-mail-address: _____

Invoice address: _____

_____

_____

Training company: _____

Trainer: _____

Foundation Level Sample Exam

SET C (v2.4.0) – GTB Edition – ANSWERS

CTFL Syllabus Version v4.0.2

ISTQB® Certified Tester Foundation Level

Legal

Copyright © 2023 International Software Testing Qualifications Board (hereinafter called ISTQB®). All rights reserved.

The authors transfer the copyright to the International Software Testing Qualifications Board (hereinafter called ISTQB®). The authors (as current copyright holders) and ISTQB® (as the future copyright holder) have agreed to the following condition of use:

Any ISTQB® Member Board may translate this document.
The ISTQB® Examination Working Group is responsible for this document.
ISTQB® Working Group EXAM 2023

Acknowledgements

This document was created by a core team of the ISTQB®: Laura Albert, Wim de Coutere, Arnika Hryszko, Gary Mogyorodi (technical reviewer), Meile Posthuma, Gandhinee Rajkomar, Stuart Reid, Jean-François Riverin, Adam Roman, Lucjan Stapp, Stephanie Ulrich, Yaron Tsubery and Eshraka Zakaria.

The core team thanks the review team: Amanda Alderman, Alexander Alexandrov, Jürgen Beniermann, Rex Black, Young jae Choi, Nicola De Rosa, Klaudia Dussa-Zieger, Klaus Erlenbach, Joëlle Genois, Tamás Gergely, Dot Graham, Matthew Gregg, Gabriele Haller, Chinthaka Indikadahena, John Kurowski, Ine Lutterman, Isabelle Martin, Patricia McQuaid, Dénes Medzihradszky, Blair Mo, Gary Mogyorodi, Jörn Münzel, Markus Niehammer, Ingvar Nordström, Fran O'Hara, Raul Onisor, Dénes Orosz, Arnd Pehl, Horst Pohlmann, Nishan Portoyan, Ale Rebon Portillo, Stuart Reid, Ralf Reissing, Liang Ren, Jean-Francois Riverin, Lloyd Roden, Tomas Rosenqvist, Murian Song, Szilard Szell, Giancarlo Tomasig, Joanne Tremblay, François Vaillancourt, Daniel van der Zwan, André Verschelling and Paul Weymouth for their suggestions and ideas.

Revision History

Version	Date	Remarks
2.0	24.11.2024	Localized version and internal reviews
2.1	16.02.2025	Final after editing and changes for CTFL 4.0.2 errata compliance
2.2.1	06.03.2025	Q17 justification corrected
2.3.0	08.06.2025	Wording of the German-language questions improved
2.4.0	30.04.2026	Question 31 revised and wording improved in many German-language questions and justifications.

Introduction

This is a sample exam. It helps candidates to prepare for the actual certification exam. Questions are included whose structure, layout and format are like a regular ISTQB®/ GTB Certified Tester Foundation Level exam. It is strictly forbidden to use the exam questions as content of a certification exam.

1) Any individual or training provider may use this sample exam in a training course if ISTQB® is acknowledged as the source and copyright holder of the sample exam.
2) Any individual or group of individuals may use this sample exam as the basis for articles, books or other derivative writings if ISTQB® is acknowledged as the source and copyright holder of the sample exam.
3) Any national board recognized by ISTQB® may translate this sample exam and make it publicly available if ISTQB® is acknowledged as the source and copyright holder of the sample exam.
4) Exactly one correct solution is expected for almost every question. The exceptions explicitly mention the possibility of multiple answers.

Exam notes

Number of questions: 40

Duration of the exam: 60 minutes

Total score: 40 (one point per question)

Score to pass the exam: 26 (or more)

Percentage of passing the exam: 65% (or more)

Question 1	FL-1.1.1	K1	Score 1.0

Which of the following is a typical test objective?

Please select ONE option! (1 out of 4)

a)	Validating that documented requirements are met.	☐
b)	Causing failures and identifying defects.	☒
c)	Initiating errors and identifying root causes.	☐
d)	Verifying the test object meets user expectations.	☐

FL-1.1.1 (K1) Identify typical test objectives.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.1.1)

a) FALSE – Validating that documented requirements are met is incorrect as validation is concerned with meeting user requirements and expectations, while verification is concerned with meeting specified requirements, so this would be correct if we replaced ‘validating’ with ‘verifying’ (see [CTFL 4.0.2], section 1.1.1, 5th bullet point under ‘Typical test objectives’).

b) CORRECT – Causing failures and identifying defects is explicitly stated as a typical test objective in the CTFL Syllabus V4.0.2, Section 1.1.1.

c) FALSE – Initiating errors and identifying root causes is incorrect because testers do not initiate errors, they try to cause failures. Errors are typically made by developers (and cannot really be initiated) and result in defects, which testers attempt to identify either directly through static testing or indirectly through failures with dynamic testing. Identifying root causes is useful but is part of debugging, which is a separate activity to testing (see [CTFL 4.0.2], section 1.1.1, 2nd bullet point partial and section 1.1.2, last paragraph).

d) FALSE – Verifying the test object meets user expectations is incorrect as verification is concerned with checking specified (documented) requirements are met, while validation is concerned with meeting user requirements and expectations, so this would be correct if we replaced ‘verifying’ with ‘validating’ (see [CTFL 4.0.2], section 1.1.1, last bullet point under ‘Typical test objectives’).

Question 2	FL-1.1.2	K2	Score 1.0

Which of the following statements BEST describes the difference between testing and debugging?

Please select ONE option! (1 out of 4)

a)	Testing identifies the causes of defects, while debugging analyzes and fixes the defects.	☐
b)	Testing is a negative activity while debugging is a positive activity.	☐
c)	Testing can find defects, while debugging analyzes and fixes defects.	☒
d)	Testing finds the cause of defects while debugging fixes the cause of defects.	☐

FL-1.1.2 (K2) Differentiate testing from debugging.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.1.2)

a) FALSE – Testing does not identify the causes of defects. The purpose of debugging is to analyze and fix defects (see [CTFL 4.0.2], section 1.1.2, 1st and 2nd paragraphs).

b) FALSE – Both testing and debugging contribute to improving the quality of the test object, so should really both be considered positively. Debugging is generally considered to be a positive activity as it is fixing something. Dynamic testing does involve intentionally causing the test object to fail, which is why some people consider it a negative activity, but that is a very narrow view (and not one typically held by testers). Both positive and negative test cases are possible. Positive test cases check that the test object correctly performs what it is supposed to do, while negative testing checks that the test object does not do what it is not supposed to do (see [CTFL 4.0.2], section 1.1.2, 1st and 2nd paragraph).

c) CORRECT – Testing and debugging are separate activities. Dynamic testing can cause failures; static testing can directly find defects. Debugging typically includes reproducing a failure, diagnosing or finding the defect that caused it, and fixing the defect (see [CTFL 4.0.2], section 1.1.2).

d) FALSE –The causes of defects are typically human errors. Testing finds defects either directly through static testing, or indirectly by causing failures in dynamic testing, and debugging fixes defects. So, testing does not find the cause of defects and debugging does not fix the causes of defects (see [CTFL 4.0.2], section 1.1.2, 3rd paragraph).

Question 3	FL-1.3.1	K2	Score 1.0

One of the principles of testing states:

“The absence-of-defects fallacy must be avoided.”

Which of the following examples shows how this principle is addressed in practice?

Please select ONE option! (1 out of 4)

a)	Explaining that it is not possible for testing to show the absence of defects.	☐
b)	Supporting the end users to perform acceptance testing.	☒
c)	Ensuring that no implementation defects remain in the delivered system.	☐
d)	Modifying tests that cause no failures to ensure few defects remain.	☐

FL-1.3.1 (K2) Explain the seven testing principles.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.3.1)

The ‘absence-of-defects fallacy’ is concerned with the idea that ensuring correctness in accordance with the requirements (i.e., verifying the absence of implementation defects) does not guarantee user satisfaction with the system. To address this it is also necessary to validate that the system meets users' needs and expectations, fulfills business objectives, and outperforms competing systems.

a) FALSE – The ‘testing shows the presence, not the absence of defects’ principle explains that while testing can detect the existence of defects in the test object, it is not possible to demonstrate that there are no defects and, therefore, guarantee its correctness. Therefore, explaining that it is not possible for testing to show the absence of defects would partially address this principle, not the ‘absence-of-defects’ fallacy (see [CTFL 4.0.2], section 1.3, 1st principle).

b) CORRECT – By supporting the end users to perform acceptance testing it should be possible to validate that the system meets users' needs and expectations (see [CTFL 4.0.2], section 1.3, 7th principle, last paragraph).

c) FALSE – It is not possible to ensure that no implementation defects remain in the delivered system as the ‘testing shows the presence, not the absence of defects’ principle explains that while testing can detect the existence of defects in the test object, it is not possible to demonstrate that there are no defects and, therefore, guarantee its correctness (see [CTFL 4.0.2], section 1.3, 1st principle).

d) FALSE – Modifying tests that cause no failures to ensure few defects remain is one way to address the ‘tests wear out’ principle. This principle is concerned with the idea that repeating identical tests on unaltered code is unlikely to uncover novel defects and therefore, modifying tests may be essential. This will not validate that the system meets users' needs and expectations (see [CTFL 4.0.2], section 1.3, 5th principle).

Question 4	FL-1.4.1	K2	Score 1.0

Which of the following test activities are MOST likely to involve the application of boundary value analysis and equivalence partitioning?

Please select TWO options! (2 out of 5)

a)	Test implementation	☐
b)	Test design	☒
c)	Test execution	☐
d)	Test monitoring	☐
e)	Test analysis	☒

FL-1.4.1 (K2) Summarize the different test activities and tasks.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.4.1)

Given the following description of test analysis:

To identify the features that require testing, the test basis is analyzed and defined as test conditions, which are then prioritized along with related risks. The systematic identification of test conditions as coverage items often involves using test techniques both during test analysis and as part of the test design activity.

From the above description, it can be seen that test techniques are often used in the test analysis and test design activities. Boundary value analysis and equivalence partitioning are test techniques.

a) FALSE – Test implementation is not likely to involve the use of test techniques as it is mostly concerned with assembling test cases into test procedures, while test techniques create test cases (see [CTFL 4.0.2], Section 1.4.1, 5th paragraph: 'The test realisation includes...').

b) CORRECT – Test design is likely to involve the use of test techniques to create test cases from test conditions and coverage items (see [CTFL 4.0.2], Section 1.4.1, 4th paragraph: 'The test design includes...').

c) FALSE – Test execution is primarily concerned with running test procedures and comparing actual and expected results. Test techniques such as boundary value analysis and equivalence partitioning are mainly used during test analysis and test design (see [CTFL 4.0.2], section 1.4.1, 6th paragraph: 'The test procedure includes...').

d) FALSE – Test monitoring is concerned with ongoing checks of test progress against the test plan. It is not the activity in which test techniques are typically applied to derive test conditions or test cases (see [CTFL 4.0.2], section 1.4.1, 6th paragraph: 'The test procedure includes...').

e) CORRECT – Test analysis is likely to involve the use of test techniques to identify test conditions (see [CTFL 4.0.2], Section 1.4.1, 3rd paragraph: 'The test analysis includes...').

Question 5	FL-1.4.3	K2	Score 1.0

Given the following test activities:

A. Test analysis

B. Test design

C. Test implementation

D. Test completion

And the following testware:

1. Coverage items

2. Change requests

3. Test execution schedule

4. Prioritized test conditions

Which of the following BEST shows the testware produced by the activities?

Please select ONE option! (1 out of 4)

a)	1B, 2D, 3C, 4A	☒
b)	1B, 2D, 3A, 4C	☐
c)	1D, 2C, 3A, 4B	☐
d)	1D, 2C, 3B, 4A	☐

FL-1.4.3 (K2) Differentiate the testware that supports the test activities.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.4.3)

Considering each of the listed test activities and their output testware:

a) Test analysis - prioritized test conditions (4) (e.g., acceptance criteria), and defect reports for defects identified in the test basis (see [CTFL 4.0.2], Section 1.4.3, 4th paragraph: 'Test analysis... (prioritised) test conditions (...)').

b) Test design - prioritized test cases, test charters, coverage items (1), test data requirements, and test environment requirements (see [CTFL 4.0.2], Section 1.4.3, 5th paragraph: '[...]coverage items [...]').

c) Test implementation - test procedures, automated test scripts, test suites, test data, test execution schedule (3), and test environment elements such as stubs, drivers, simulators, and service virtualizations (see [CTFL 4.0.2], Section 1.4.3, 6th paragraph: 'Test execution plans...').

d) Test completion - test completion report, documented lessons learned, action items for improvement, and change requests (2) (as product backlog items) (see [CTFL 4.0.2], section 1.4.3, last paragraph: 'Amendments...').

Thus:

a) CORRECT – The correct match is: 1B, 2D, 3C, 4A

b) FALSE

c) FALSE

d) FALSE

Question 6	FL-1.4.5	K2	Score 1.0

Which of the following statements about testing roles is MOST likely to be CORRECT?

Please select ONE option! (1 out of 4)

a)	In Agile software development, the test management role is the primary responsibility of the team, while the testing role is primarily the responsibility of a single individual from outside the team.	☐
b)	The testing role is primarily responsible for test monitoring and control, while the test management role is primarily responsible for test planning and test execution.	☐
c)	In Agile software development, test management activities that span multiple teams are handled by a test manager outside the team, while some test management tasks are handled by the team itself.	☒
d)	The test management role is primarily responsible for test analysis and test design, while the testing role is primarily responsible for test implementation and execution.	☐

FL-1.4.5 (K2) Compare the different roles in testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.4.5)

a) FALSE – Although it is correct to say that in Agile software development, some of the test management tasks may be handled by the Agile team itself, the testing role is not primarily the responsibility of a single individual from outside the team. Instead the testing is more likely to be performed by various team members following the whole-team approach.

b) FALSE – The test management role primarily involves activities related to test planning, test monitoring and control, and test completion. So, although this statement is partially correct, it is wrong to say that the testing role is primarily responsible for test monitoring and control.

c) CORRECT – In Agile software development, some of the test management tasks may be handled by the Agile team itself. However, for test activities that span multiple teams within an organization, test managers outside of the development team may perform these tasks (see [CTFL 4.0.2], section 1.4.5, 2nd paragraph).

d) FALSE – The test management role primarily involves activities related to test planning, test monitoring and control, and test completion, while the testing role is primarily responsible for the technical and engineering aspects of testing, such as test analysis, test design, test implementation, and test execution. Thus the test management role is not normally responsible for test analysis and test design, although it is correct to say that the testing role is primarily responsible for test implementation and test execution.

Question 7	FL-1.5.2	K1	Score 1.0

What is a typical advantage of the whole-team approach in Agile projects?

Please select ONE option! (1 out of 4)

a)	The team works without dedicated testers.	☐
b)	Improved team dynamics.	☒
c)	Only business representatives are responsible for testing.	☐
d)	Teams are enlarged by adding additional roles.	☐

FL-1.5.2 (K1) Recall the advantages of the whole-team approach.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.5.2)

a) FALSE – In the whole-team approach, testers play a vital role by sharing their testing expertise with the team and guiding product development. They collaborate with other team members to achieve the desired quality levels and work with business representatives to create acceptance tests. Testers also partner with developers to determine the optimal test strategy and automation approaches.

b) CORRECT – By leveraging the diverse skill sets of each team member most effectively, the whole-team approach fosters superior team dynamics, promotes robust communication and collaboration, and generates a synergistic effect that benefits the entire project (see [CTFL 4.0.2], Section 1.5.2, ‘’The whole-team approach improves team dynamics, (...)").

c) FALSE – The whole-team approach does not mean that only business representatives are responsible for testing. Instead, testers, developers, and business representatives collaborate, and responsibility for quality is shared across the whole team (see [CTFL 4.0.2], section 1.5.2).

d) FALSE – There is no specific guidance on the optimum size of teams using the whole-team approach, and there is no suggestion that larger teams are better.

Question 8	FL-1.5.3	K2	Score 1.0

Which of the following statements about the independence of testing is correct?

Please select ONE option! (1 out of 4)

a)	Independent testers are more likely to find defects due to their different perspective from developers, but their independence may lead to tensions in the team.	☒
b)	Developers' familiarity with their own code means they only find a few defects in it, however their shared software background with testers means these defects would also be found by the testers.	☐
c)	Independent testing requires testers who are outside the developer's team and ideally from outside the organization, however these testers find it difficult to understand the application domain.	☐
d)	Testers from outside the developer's team are more independent than testers from within the team, but the testers from within the team are more likely to be blamed for delays in product release.	☐

FL-1.5.3 (K2) Distinguish the benefits and drawbacks of independence of testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 1.5.3)

a) CORRECT – The primary benefit of independence in testing is that testers are more likely to identify different types of failures and defects compared to developers, due to their varied backgrounds, technical viewpoints, and potential biases, including cognitive bias. However, the main disadvantage of independence in testing is that testers may become isolated from the development team, leading to communication problems, a lack of collaboration, and potentially an adversarial relationship, with testers being blamed for delays and bottlenecks in the release process (see [CTFL 4.0.2], section 1.5.3; 1st and 4th paragraph, 1st sentence in each case).

b) FALSE – A developer's familiarity with the code does not mean that they rarely find defects in it, instead this familiarity means they can efficiently find many defects in their own code. And, rather than developers and testers having a shared background, developers having a different background to testers is normally cited as the reason that testers and developers find different kinds of defects.

c) FALSE – Testing can be conducted at different levels of independence, ranging from no independence for the author to very high independence for testers from outside the organization. In most projects, multiple levels of independence are utilized, with developers performing component and component integration testing, the test team performing system and system integration testing, and business representatives performing acceptance testing. So, testers can be in the developer's team and do not need to come from outside the organization. Knowledge of the application domain will change from case-to-case and is not dependent on the level of independence.

d) FALSE – Testing can be conducted at different levels of independence, ranging from no independence for the author to very high independence for testers from outside the organization, with testers from outside the developer's team generally more independent than testers from within the team. However, there is more reason to believe that testers from outside the team are likely to be more isolated from the developers and so are more likely to be blamed for delays in product release.

Question 9	FL-2.1.2	K1	Score 1.0

Which of the following is a good testing practice that applies to all software development lifecycles?

Please select ONE option! (1 out of 4)

a)	For each test level, there is a corresponding development level.	☐
b)	For each test objective, there is a corresponding development objective.	☐
c)	For every software test activity, there is a corresponding user activity.	☐
d)	For every software development activity, there is a corresponding test activity.	☒

FL-2.1.2 (K1) Recall good testing practices that apply to all software development lifecycles.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 2.1.2)

a) FALSE – Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity. However, here we are attempting to equate test levels with development levels, and, although we know what is meant by ‘test levels’, there is no common understanding of the term ‘development level’ (see [CTFL 4.0.2], Section 2.1.2, 1st paragraph: ... For every software development activity, there is a corresponding test activity’).

b) FALSE – Every software development activity has a corresponding test activity; however test objectives are quite different. For instance, there might be a test objective of ensuring that a test object adheres to a contractual requirement that a certain type of testing must be performed before delivery. In this case there is no reason for there to be a corresponding development objective (see [CTFL 4.0.2], section 2.1.2, ‘Development Objective’).

c) FALSE – Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity. However, the same symmetry does not apply to testing and user activities. For instance, for some systems it is difficult to even identify the end users. Also, some test activities are focused on developers (e.g., testing for ease of maintainability), which has no user aspect to it.

d) CORRECT – Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity (see [CTFL 4.0.2], section 2.1.2, 1st paragraph, 1st bullet point).

Question 10	FL-2.1.3	K1	Score 1.0

Which of the following is an example of a test-first approach to development?

Please select ONE option! (1 out of 4)

a)	Feature-Driven Development	☐
b)	Integration Test-Driven Development	☐
c)	System Test-Driven Development	☐
d)	Acceptance Test-Driven Development	☒

FL-2.1.3 (K1) Recall the examples of test-first approaches to development.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 2.1.3)

a) FALSE – Feature-Driven Development is mentioned in the CTFL syllabus as a software development method or Agile practice, but it is not an example of a test-first approach. Examples of test-first approaches are TDD, ATDD, and BDD (see [CTFL 4.0.2], section 2.1.3).

b) FALSE – Integration Test-Driven Development is not a correct example of a test-first approach to development.

c) FALSE – System Test-Driven Development is not a correct example of a test-first approach to development.

d) CORRECT – Acceptance Test-Driven Development (ATDD) is a well-known example of a test-first approach to development (see [CTFL 4.0.2], Section 2.1.3, 3rd paragraph, 2nd sentence: 'Tests are written before the part of the application that fulfils the tests is developed.').

Question 11	FL-2.1.5	K2	Score 1.0

Which of the following statements BEST describes shift-left in software testing?

Please select ONE option! (1 out of 4)

a)	When agreed by the developers, manual tests are automated to save time and costs.	☐
b)	Where cost-effective, test activities are moved to be performed earlier in the software development lifecycle (SDLC) to reduce the total cost of quality by reducing the number of defects found later in the SDLC.	☒
c)	When they have spare time available, testers are required to automate tests for regression testing, starting with component tests and component integration tests.	☐
d)	When available, testers are trained to perform tasks early in the SDLC to allow more test activities to be automated later in the SDLC.	☐

FL-2.1.5 (K2) Explain the shift-left approach.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 2.1.5)

a) FALSE – Shift-left is not primarily about automating manual tests after approval by developers. It is about performing test activities earlier in the SDLC.

b) CORRECT – Shift-left emphasizes the importance of starting testing earlier in the software development lifecycle (SDLC). Implementing shift-left testing necessitates additional training, and increased effort and costs during the early stages of the SDLC, nevertheless, overall savings should be higher (see [CTFL 4.0.2], section 2.1.5. 1. and last paragraph).

c) FALSE – Although automated component tests and component integration tests for regression testing are generally valuable, the creation of these tests is normally the responsibility of the developers, and if a continuous integration/continuous delivery (CI/CD) approach is followed, then these tests will have been submitted with the code. In some situations the tester may automate tests for regression testing, and sometimes even for component tests and component integration tests, however this is not part of a 'shift-left' approach which moves testing earlier in the SDLC.

d) FALSE – Training testers to perform tasks early in the SDLC would support a shift-left approach by emphasizing the importance of starting testing earlier in the SDLC. However, automating more test activities to be performed later in the SDLC is not part of a 'shift-left' approach.

Question 12	FL-2.1.6	K2	Score 1.0

Which of the following is LEAST likely to occur as a result of a retrospective?

Please select ONE option! (1 out of 4)

a)	The quality of future test objects improves by identifying improvements in development practices.	☐
b)	Test efficiency improves by speeding up the configuration of test environments through automation.	☐
c)	End users' understanding of the development and test processes is improved.	☒
d)	Automated test scripts are enhanced through feedback from developers.	☐

FL-2.1.6 (K2) Explain how retrospectives can be used as a mechanism for process improvement.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 2.1.6)

a) FALSE – One of the purposes of retrospectives is to identify potential process improvements, which, if put into practice, should result in the quality of future outputs of the development process (test objects) being higher. So, this is likely to occur as a result of a retrospective.

b) FALSE – A benefit of retrospectives for testing includes increased test efficiency through process improvements. So, this is likely to occur as a result of a retrospective.

c) CORRECT – Participants at retrospectives typically include testers, developers, architects, product owners, and business analysts, but end users are rarely invited or attend these meetings – and they are also unlikely to receive any reports from these meetings. So, it is very unlikely that they will learn and understand more about the development and test processes through retrospectives (see [CTFL 4.0.2], section 2.1.6, 1st paragraph).

d) FALSE – A benefit of retrospectives for testing includes improved quality of testware (including automated test scripts) through joint reviews with developers. So, this is likely to occur as a result of a retrospective.

Question 13	FL-2.2.1	K2	Score 1.0

At which test level are validation and demonstrating readiness for deployment the main focus, and the tests should ideally be performed by the intended users?

Please select ONE option! (1 out of 4)

a)	Component testing	☐
b)	Component integration testing	☐
c)	System integration testing	☐
d)	Acceptance testing	☒

FL-2.2.1 (K2) Distinguish the different test levels.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 2.2.1)

a) FALSE – Component testing (also called unit testing) involves testing individual components in isolation and is mostly verification against a specification, rather than validation against user needs. However, this testing is not normally performed by testers, as developers usually carry out this testing in their development environment.

b) FALSE – Component integration testing involves testing the interfaces and interactions between components and is mostly verification against a specification, rather than validation against user needs. However, this testing is not normally performed by testers, as developers usually carry out this testing.

c) FALSE – System integration testing examines the interfaces with other systems and external services and is mostly verification against a specification, rather than validation against user needs. This type of testing is also most often performed by testers.

d) CORRECT – Acceptance testing focuses on validation and demonstrating readiness for deployment, i.e., on determining whether the system satisfies the user's business needs. Ideally, acceptance tests should be performed by the intended users (see [CTFL 4.0.2], section 2.2.1).

Question 14	FL-2.2.3	K2	Score 1.0

After a software update, the navigation system suggests routes such as driving the wrong way down one-way streets - a clear violation of traffic laws.

After the underlying defect has been fixed, which of the following actions is MOST appropriate in this case?

Please select ONE option! (1 out of 4)

a)	Only confirmation testing.	☐
b)	Confirmation testing then regression testing.	☒
c)	Only regression testing.	☐
d)	Regression testing then confirmation testing.	☐

FL-2.2.3 (K2) Distinguish confirmation testing from regression testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 2.2.3)

a) FALSE – Confirmation testing to check that the updates have resulted in a correct implementation is necessary, however, it would then be sensible to perform regression testing to ensure that no defects have been introduced or uncovered in unchanged areas of the system.

b) CORRECT – Confirmation testing will check that the updates have resulted in a correct implementation, and then regression testing will be used to ensure that no defects have been introduced or uncovered in unchanged areas of the system (see [CTFL 4.0.2], section 2.2.3, 2nd and 3rd paragraphs).

c) FALSE – Regression testing should be used to ensure that no defects have been introduced or uncovered in unchanged areas of the system when the update was made, however it is also necessary to perform confirmation testing that will check that the updates have resulted in a correct implementation.

d) FALSE – Confirmation testing will check that the updates have resulted in a correct implementation, and regression testing will be used to ensure that no defects have been introduced or uncovered in unchanged areas of the system. However, when performed (i.e., when an update needs to be tested), confirmation testing precedes regression testing.

Question 15	FL-3.1.3	K2	Score 1.0

Which of the following example defects BEST identifies example defects that could be found by static testing (rather than dynamic testing)?

Please select TWO options! (2 out of 5)

a)	Two different parts of the design specification disagree due to the complexity of the design.	☒
b)	A response time is too long and so makes users lose patience.	☐
c)	An failure occurs when the system attempts to write a file while running out of disk space.	☐
d)	A variable is declared but never subsequently used in the program.	☒
e)	The amount of memory needed by the program to generate a report is too high.	☐

FL-3.1.3 (K2) Compare and contrast static and dynamic testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 3.1.3)

Considering each of the listed example defects:

a) Two different parts of the design specification disagree due to the complexity of the design – this is an example of a specification defect, which includes inconsistencies, ambiguities, contradictions, omissions, inaccuracies, and duplications, which can most easily be found by static testing (see [CTFL 4.0.2], section 3.1.3, 2nd paragraph, 2nd bullet point).

b) A response time is too long and so makes users lose patience – this is an example of a response time defect, which can only be detected in practice by executing the program and measuring the response time, which can most easily be found by dynamic testing.

c) Memory and file-writing issues can only be discovered through dynamic testing, as they involve the system's runtime behavior (see [CTFL 4.0.2], Section 3.1.3, 4th paragraph: "Dynamic testing investigates runtime issues affecting file processing and resource constraints").

d) A variable is declared but never subsequently used in the program - this is an example of a coding defect, which includes variables with undefined values, undeclared variables, duplicated or unreachable code, and excessive code complexity, which can most easily be found by static testing (see [CTFL 4.0.2], section 6.1.1), (see [CTFL 4.0.2], section 3.1.3, 2nd paragraph, 3rd bullet point).

e) The amount of memory needed by the program to generate a report is too high – this is an example of a performance defect, which can only be detected in practice by executing the program and measuring the memory used, which can most easily be found by dynamic testing.

Thus: The correct answers for static testing are a) and d).

Question 16	FL-3.2.1	K1	Score 1.0

Which of the following is a benefit of early and frequent stakeholder feedback?

Please select ONE option! (1 out of 4)

a)	Changes to requirements are understood and implemented earlier.	☒
b)	Business stakeholders receive training on user requirements.	☐
c)	It allows product owners to change their requirements as often as they want.	☐
d)	End users are told which requirements will not be implemented prior to release.	☐

FL-3.2.1 (K1) Identify the benefits of early and frequent stakeholder feedback.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 3.2.1)

a) CORRECT – Obtaining feedback from stakeholders early and often in the software development process can be highly beneficial. It facilitates early communication of potential quality issues, can prevent misunderstandings about requirements, and ensures that any changes in stakeholder requirements are understood and implemented sooner (see [CTFL 4.0.2], section 3.2.1, 2nd paragraph, 1st sentence).

b) FALSE – The feedback is from stakeholders, and them providing feedback is unlikely to improve their understanding of their own user requirements.

c) FALSE – Obtaining feedback from stakeholders early and often in the software development process can be highly beneficial. It facilitates early communication of potential quality issues, can prevent misunderstandings about requirements, and ensures that any changes in stakeholder requirements are understood and implemented sooner. However, because changes in requirements can be understood and implemented sooner, it does not mean that unlimited changes to requirements are encouraged.

d) FALSE – The feedback is from stakeholders and does not cover communication to them. Communications to end users could include them being told about which requirements will not be implemented prior to release.

Question 17	FL-3.2.4	K2	Score 1.0

Given the following review types:

1. **Technical review**
2. **Informal review**
3. **Inspection**
4. **Walkthrough**

And the following descriptions:

A) Led by the author, this review aims, among other things, at building confidence, generating ideas, and training reviewers.

B) Primarily aims at detecting potential defects and does not require formal documentation.

C) The main objective is detecting a maximum number of potential defects and collecting metrics to support process improvement.

D) Aims to reach consensus through technically qualified reviewers and to identify anomalies.

Which of the following assignments is correct?

Please select ONE option! (1 out of 4)

a)	1A, 2B, 3C, 4D	☐
b)	1D, 2B, 3C, 4A	☒
c)	1B, 2C, 3D, 4A	☐
d)	1C, 2D, 3A, 4B	☐

FL-3.2.4 (K2) Compare and contrast the different review types.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 3.2.4)

Considering each of the listed review types:

1. Technical review - This type of review is performed by technically qualified reviewers and led by a moderator. The objectives are to gain consensus and make decisions on technical problems while also evaluating quality and building confidence in the work product, generating new ideas, motivating and enabling authors to improve, and detecting anomalies. Assignment: 1D.

2. Informal review - The main objective is to detect anomalies. The process is not defined and does not require formal documented output. Assignment 2B.

3. Inspection - This is the most formal review type, and it follows the complete generic review process. The primary objective is to find the most anomalies, and other objectives include evaluating quality and building confidence in the work product, motivating and enabling authors to improve, and collecting metrics that can be used to enhance the software development life cycle (SDLC), including the inspection process. The author cannot act as the review leader or scribe. Assignment: 3C

4. Walkthrough - Led by the author, this type of review serves various objectives such as evaluating quality and building confidence in the work product, educating reviewers, gaining consensus, generating new ideas, motivating and enabling authors to improve, and detecting anomalies. Reviewers might perform an individual review before the walkthrough, but this is not mandatory. Assignment: 4A.

Thus:

a) Is not correct

b) Is correct.

c) Is not correct

d) Is not correct

Question 18	FL-3.2.5	K1	Score 1.0

Which of the following is a factor that contributes to a successful review?

Please select ONE option! (1 out of 4)

a)	Ensure management participate as reviewers.	☐
b)	Split large work products into smaller parts.	☒
c)	Set reviewer evaluation as an objective.	☐
d)	Plan to cover one document per review.	☐

FL-3.2.5 (K1) Recall the factors that contribute to a successful review.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 3.2.5)

a) FALSE – To ensure successful reviews, it's important to secure management's support for the review process, however that does not mean that they should participate as reviewers.

b) CORRECT – To ensure successful reviews, it's important to break the work product into parts that are small enough to be reviewed in a reasonable timescale to prevent reviewers from losing focus during individual review or review meetings (see [CTFL 4.0.2], section 3.2.5, 3rd bullet point).

c) FALSE – To ensure successful reviews, it's important to clearly define objectives and measurable exit criteria, without evaluating participants.

d) FALSE – To ensure successful reviews, it's important to break down the review into smaller chunks to prevent reviewers from losing focus during individual review or review meetings. So you should not plan to cover one document per review.

Question 19	FL-4.1.1	K2	Score 1.0

What is the MAIN difference between black-box test techniques and experience-based test techniques?

Please select ONE option! (1 out of 4)

a)	The test object.	☐
b)	The test level at which the test technique is used.	☐
c)	The test basis.	☒
d)	The software development lifecycle (SDLC) in which the test technique can be used.	☐

FL-4.1.1 (K2) Distinguish black-box, white-box and experience-based test techniques.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.1.1)

a) FALSE – In most cases both black-box test techniques and experience-based test techniques can be used for the same test objects.

b) FALSE – Both black-box test techniques and experience-based test techniques can be used at all test levels.

c) CORRECT – Black-box test techniques (also known as specification-based techniques) are based on an analysis of the specified behavior of the test object without reference to its internal structure. So, the test basis is usually a specification. Experience-based test techniques effectively use the knowledge and experience of testers for the design and implementation of test cases. This means that the tester, when designing tests, may not use the specification at all (see [CTFL 4.0.2], section 4.1).

d) FALSE – Experience-based test techniques can detect defects that may be missed using black-box (and white-box) test techniques. Hence, experience-based test techniques are complementary to black-box test techniques and white-box test techniques and both black-box test techniques and experience-based test techniques can be used in all SDLCs.

Question 20	FL-4.2.1	K3	Score 1.0

A PIN validator accepts valid PINs and rejects invalid ones. Only inputs consisting of digits are considered.

A PIN is valid if it consists of exactly four digits and at least two of them are different.

Which of the following data sets covers all relevant equivalence partitions?

Please select ONE option! (1 out of 4)

a)	112, 1111, 1234, 123456	☒
b)	1, 123, 1111, 1234	☐
c)	12, 112, 1112, 11112	☐
d)	1, 111, 1111, 11111	☐

FL-4.2.1 (K3) Use equivalence partitioning to derive test cases.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.2.1)

The equivalence class for valid PIN and invalid PINs are:

1. Valid PINs:
   - A PIN with exactly four digits, of which at least two are different (e.g. 1234, 1123, 5678).
2. Invalid PINs:
   - PINs with fewer than four digits (e.g. 1, 12, 123).
   - PINs with more than four digits (e.g. 123456, 11111).
   - PINs with four identical digits (e.g. 1111, 2222).

a) CORRECT – 112, 1111, 1234, 123456

- 112 → Invalid (less than four digits).
- 1111 → Invalid (four identical digits).
- 1234 → Valid (four digits, at least two different).
- 123456 → Invalid (more than four digits).

This option covers all equivalence classes:

- Less than four digits (112)
- Four identical digits (1111)
- Valid PIN (four digits, at least two different) (1234)
- More than four digits (123456)

This option covers all equivalence classes and is correct.

b) FALSE – 1, 123, 1111, 1234

- 1 → Invalid (less than four digits).
- 123 → Invalid (less than four digits).
- 1111 → Invalid (four identical digits).
- 1234 → Valid.

The equivalence class "more than four digits" is missing.

c) FALSE – 12, 112, 1112, 11112

- 12 → Invalid (less than four digits).
- 112 → Invalid (less than four digits).
- 1112 → Valid (four digits, at least two different).
- 11112 → Invalid (more than four digits).

The equivalence class "four identical digits" (e.g., 1111) is missing.

→ Incomplete, therefore incorrect.

d) FALSE – 1, 111, 1111, 11111

- 1 → Invalid (less than four digits).
- 111 → Invalid (less than four digits).
- 1111 → Invalid (four identical digits).
- 11111 → Invalid (more than four digits).

This option does not include any valid PIN.
The equivalence class for valid PINs (e.g., 1234) is completely missing.

Question 21	FL-4.2.2	K3	Score 1.0

Given the following business rule:

INPUT: value (integer number)

IF (value ≤ 100 OR value ≥ 200) THEN write "value incorrect"

ELSE write "value OK"

You want to design test cases using boundary value analysis with two values per boundary.

Which of the following sets of test values completely covers the relevant boundary values?

Please select ONE option! (1 out of 4)

a)	100, 150, 200, 201	☐
b)	99, 100, 200, 201	☐
c)	98, 99, 100, 101	☐
d)	100, 101, 199, 200	☒

FL-4.2.2 (K3) Use boundary value analysis to derive test cases.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.2.2)

The equivalence partitions are: {…, 99, 100}, {101, 102, …, 198, 199}, and {200, 201, …}.

Thus, the relevant boundary values are 100, 101, 199, and 200.

With two-value boundary value analysis for the two relevant boundaries, the values immediately on both sides of each boundary are selected. Therefore, the four relevant coverage items are 100, 101, 199, and 200.

Thus:

a) FALSE – Only 100 and 200 are included from the four relevant boundary values; 101 and 199 are missing, so only 50% coverage is achieved.

b) FALSE – Only 100 and 200 are included from the four relevant boundary values; 101 and 199 are missing, so only 50% coverage is achieved.

c) FALSE – Only 100 and 101 are included from the four relevant boundary values; 199 and 200 are missing, so only 50% coverage is achieved.

d) CORRECT – 100, 101, 199, and 200 are exactly the four relevant boundary values, so full coverage is achieved.

Question 22	FL-4.2.3	K3	Score 1.0

You are working on a project to develop a system to analyze driving test results. You have been asked to design test cases based on the following decision table.

	R1	R2	R3
C1: First attempt at the exam?	-	-	F
C2: Theoretical exam passed?	T	F	-
C3: Practical exam passed?	T	-	F
Issue a driving license?	X		
Request additional driving lessons?			X
Request to take the exam again?		X	

What test data will show that there are contradictory rules in the decision table?

Please select ONE option! (1 out of 4)

a)	C1 = T, C2 = T, C3 = F	☐
b)	C1 = T, C2 = F, C3 = T	☐
c)	C1 = T, C2 = T, C3 = T and C1 = F, C2 = T, C3 = T	☐
d)	C1 = F, C2 = F, C3 = F	☒

FL-4.2.3 (K3) Use decision table testing to derive test cases.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.2.3)

a) FALSE – The combination (T, T, F) does not match any rule. This is an example of omission, not a contradiction.

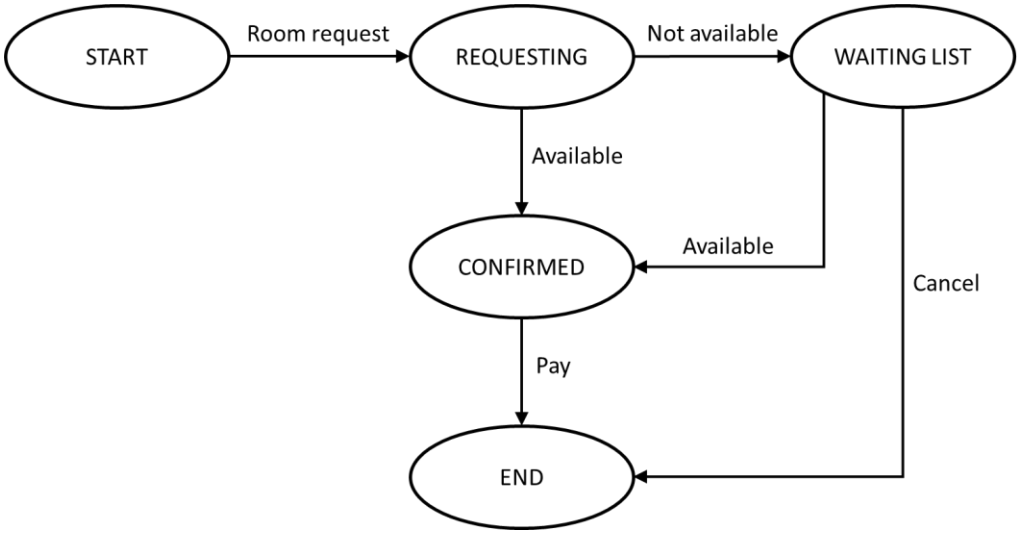
b) FALSE – The combination (T, F, T) matches only one column, R2, so there is no contradiction.

c) FALSE – Both combinations (T, T, T) and (F, T, T) match only one column, R1, so there is no contradiction.

d) CORRECT – The combination (F, F, F) matches both R2 and R3, but R2 and R3 have different actions, so this shows a contradiction between R2 and R3.

Question 23	FL-4.2.4	K3	Score 1.0

You are designing test cases based on the following state transition diagram:



What is the MINIMUM number of test cases required to achieve 100% valid transitions coverage?

Please select ONE option! (1 out of 4)

a)	3	☒
b)	2	☐
c)	5	☐
d)	6	☐

FL-4.2.4 (K3) Use state transition testing to derive test cases.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.2.4)

The following three transitions:

> “REQUESTING -> CONFIRMED”
>
> “WAITING LIST -> CONFIRMED”
>
> “WAITING LIST -> END”

cannot appear in the same test case, which suggests that at least three test cases are required. All the other transitions can appear in combination with one or more of these three transitions, so we need a minimum of three test cases. In fact, only three sequences are possible:

> TC1: START (Room request) -> REQUESTING (Available) -> CONFIRMED (Pay) -> END
>
> TC2: START (Room request) -> REQUESTING (Not available) -> WAITING LIST (Available) □ CONFIRMED (Pay) -> END
>
> TC3: START (Room request) -> REQUESTING (Not available) -> WAITING LIST (Cancel) -> END

Thus:

a) CORRECT

b) FALSE

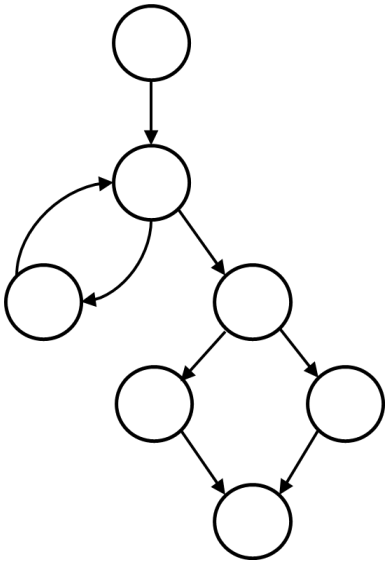
c) FALSE

d) FALSE

The correct option is a) because three test cases are sufficient to cover all valid transitions in the state transition diagram. The prerequisite is that the transitions shown in the diagram are clearly identifiable as valid transitions.

Question 24	FL-4.3.2	K2	Score 1.0

You want to apply branch testing to the code represented by the following control flow graph.



How many coverage items do you need to test?

Please select ONE option! (1 out of 4)

a)	2	☐
b)	4	☐
c)	8	☒
d)	7	☐

FL-4.3.2 (K2) Explain branch testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.3.2)

In branch testing the coverage items are branches, which are represented by the edges of a control flow graph. There are 8 edges in the control flow graph.

Thus:

a) FALSE

b) FALSE

c) CORRECT

d) FALSE

Question 25	FL-4.3.3	K2	Score 1.0

How can white-box testing be useful in support of black-box testing?

Please select ONE option! (1 out of 4)

a)	White-box coverage measures can help testers evaluate black-box tests in terms of the code coverage achieved by these black-box tests.	☒
b)	White-box coverage analysis can help testers identify unreachable fragments of the source code.	☐
c)	Branch testing subsumes black-box test techniques, so achieving full branch coverage guarantees achieving full coverage of any black-box technique.	☐
d)	White-box test techniques can provide coverage items for black-box techniques.	☐

FL-4.3.3 (K2) Explain the value of white-box testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.3.3)

a) CORRECT – Performing only black-box testing does not provide a measure of actual code coverage. White-box coverage measures provide an objective measurement of coverage and provide the necessary information to allow additional tests to be generated to increase this coverage, and subsequently increase confidence in the code.

b) FALSE – This statement is correct, but it has nothing to do with black-box testing.

c) FALSE – In general there are no subsumes relationships between white-box and black-box techniques.

d) FALSE – White-box techniques are used to design tests based on the test object itself, while black-box techniques are used to design tests based on the specification. Therefore, there is no relation between coverage items derived from these two types of techniques.

Question 26	FL-4.4.1	K2	Score 1.0

Consider the following list:

- **Correct input not accepted**
- **Incorrect input accepted**
- **Wrong output format**
- **Division by zero**

What test technique is MOST PROBABLY used by the tester who uses this list when performing testing?

Please select ONE option! (1 out of 4)

a)	Exploratory testing	☐
b)	Fault attack	☒
c)	Checklist-based testing	☐
d)	Boundary value analysis	☐

FL-4.4.1 (K2) Explain error guessing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.4.1)

a) FALSE – Exploratory testing uses test charters, not a list of possible defects/failures. Although exploratory testing can incorporate the use of other test techniques, in this case fault attack is the most likely option.

b) CORRECT – This is a list of possible failures. Fault attacks are a methodical approach to the implementation of error guessing and require the tester to create or acquire a list of possible errors, defects and failures, and to design tests that will identify defects associated with the errors, expose the defects, or cause the failures.

c) FALSE – The tester is using a checklist of items to support their testing. Both error guessing and checklist-based testing use such lists, however, the list here is of possible failures, not test conditions, and so the MOST PROBABLE test technique is fault attack, which focuses on errors, defects and failures.

d) FALSE – BVA is based on an analysis of boundary values of equivalence partitions. The above list does not mention equivalence partitions or their boundaries.

Question 27	FL-4.4.3	K2	Score 1.0

Which of the following BEST describes how using checklist-based testing can result in increased coverage?

Please select ONE option! (1 out of 4)

a)	Checklist items can be defined at a sufficiently low level of detail, so the tester can implement and execute detailed test cases based on these items.	☐
b)	Checklists can be automated, so each time an automated test execution covers the checklist items, it results in additional coverage.	☐
c)	Each checklist item should be tested separately and independently, so the elements cover different areas of the software.	☐
d)	Two testers designing and executing tests based on the same high-level checklist items will typically perform the testing in slightly different ways.	☒

FL-4.4.3 (K2) Explain checklist-based testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.4.3)

a) FALSE – Although it is true that the tester can implement and execute detailed test cases based on the checklist, it does not explain how this would result in increased coverage.

b) FALSE – Checklist items should not be automated. But even if they are, the automated test scripts always execute the tests in the same way, which usually does not result in increased coverage.

c) FALSE – It is true that each checklist item should be tested separately and independently. But this impacts the test execution order and does not impact the achieved coverage, and so does not result in increased coverage.

d) CORRECT – If the checklists are high-level, some variability in the actual testing is likely to occur, resulting in potentially greater coverage but less repeatability. If two testers follow a checklist of high-level items, each of them may use different test data, test steps, etc. This way, one tester will probably cover some areas not covered by the other tester and this will result in increased coverage (see [CTFL 4.0.2], section 4.4.3, last sentence).

Question 28	FL-4.5.2	K2	Score 1.0

Which of the following acceptance criteria is a typical example of a scenario-oriented criterion?

Please select ONE option! (1 out of 4)

a)	The application must allow users to delete their account and all associated data upon request.	☐
b)	When a customer adds an item to their cart and proceeds to checkout, they should be prompted to log in or create an account if they haven't already done so.	☒
c)	IF (contain(product(23).Name, cart.products())) THEN return FALSE.	☐
d)	The website must comply with the ICT Accessibility 508 Standards and ensure that all content is accessible to users with disabilities.	☐

FL-4.5.2 (K2) Classify the different options for writing acceptance criteria.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.5.2)

a) FALSE – This acceptance criterion describes what rules or regulations the system must adhere to (in this case, the right to be forgotten). This is an example of a rule-oriented acceptance criterion.

b) CORRECT – This acceptance criterion describes an example scenario that must be realizable by the system. This is an example of a scenario-oriented acceptance criterion.

c) FALSE – This sentence looks more like a line of code that implements some business rule. Acceptance criteria should be written in collaboration with business representatives, and therefore should be written in language they understand. This sentence will most likely be unintelligible to these stakeholders.

d) FALSE – This acceptance criterion describes what rules or regulations the system must adhere to and how compliance will be ensured. Therefore, this is an example of a rule-oriented acceptance criterion, not a scenario-based acceptance criterion.

Question 29	FL-4.5.3	K3	Score 1.0

You are using acceptance test-driven development and designing test cases based on the following user story:

As a Regular or Special user, I want to be able to use my electronic floor card, to access specific floors.

Acceptance Criteria:

AC1: Regular users have access to floors 1 to 3

AC2: Floor 4 is only accessible to Special users

AC3: Special users have all the access rights of Regular users

Which test case is the MOST reasonable one to test AC3?

Please select ONE option! (1 out of 4)

a)	Check that a Regular user can access floors 1 and 3.	☐
b)	Check that a Regular user cannot access floor 4.	☐
c)	Check that a Special user can access floor 5.	☐
d)	Check that a Special user can access floors 1, 2 and 3.	☒

FL-4.5.3 (K3) Use acceptance test-driven development (ATDD) to derive test cases.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 4.5.3)

a) FALSE – We want to check that Special users have the rights of Regular users, so we need to test access rights for a Special user, not for a Regular user.

b) FALSE – We want to check that Special users have the rights of Regular users, so we need to test access rights for a Special user, not for a Regular user.

c) FALSE – There is no floor 5 described in the acceptance criteria. The test cases should not extend the scope of the user story. But even if we would like to perform negative testing, this test is not directly related to AC3.

d) CORRECT – This way we can check if a Special user can access floors which are accessible to a Regular user.

Question 30	FL-5.1.1	K2	Score 1.0

Which of the following is NOT a purpose or typical content of a test plan?

Please select ONE option! (1 out of 4)

a)	To define detailed concrete test data and expected results for individual component and component integration test cases.	☒
b)	To define as exit criteria from the component test level that “100% statement coverage and 100% branch coverage must be achieved”.	☐
c)	To define the forms and frequency of communication, as well as documentation templates for test reports.	☐
d)	To explain why a test level required by the test strategy, such as system integration testing, will not be performed.	☐

FL-5.1.1 (K2) Exemplify the purpose and content of a test plan.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.1.1)

a) CORRECT – A test plan may include test data requirements as part of the test approach, but it typically does not define detailed concrete test data and expected results for individual test cases. These are specified during test design and test implementation.

b) FALSE – One of the purposes of a test plan is to help ensure that the performed test activities will meet the established criteria, by including entry criteria and exit criteria. The code coverage criteria are an example of such criteria for the component test level.

c) FALSE – Documentation templates are typical content of a test plan. This helps to facilitate communication between the stakeholders by defining a standard way of communicating or reporting.

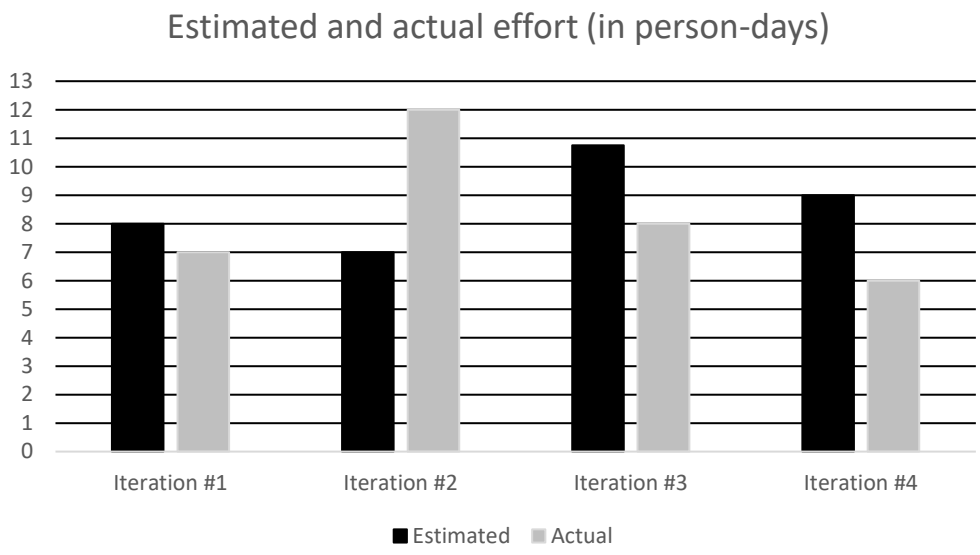
d) FALSE – One of the purposes of a test plan is to demonstrate that testing will adhere to the existing test policy and test strategy, or to explain why the testing will deviate from them. This is an example of explaining the deviation, regarding the test levels that will be (or will not be) followed.

Question 31	FL-5.1.4	K3	Score 1.0

At the beginning of each iteration, the team estimates the test effort (in person-days) it will perform during that iteration. E(n) denotes the estimated effort and A(n) the actual effort performed in iteration n. From the third iteration onward, the team uses the following extrapolation model for effort estimation:

$$E(n) = \frac{3 * A(n-1) + A(n-2)}{4}$$

A graph shows the estimated and actual values for the first four iterations.



What is the estimated test effort for iteration #5?

Please select ONE option! (1 out of 4)

a)	10.5 person-days	☐
b)	8.25 person-days	☐
c)	6.5 person-days	☒
d)	9.4 person-days	☐

FL-5.1.4 (K3) Use estimation techniques to calculate the required test effort.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.1.4)

This question is based on extrapolation, a technique for estimating the test effort. According to the syllabus, with extrapolation the effort required for the remaining work is approximated by applying a mathematical model to historical data.

From the graph we have:

A(4)=6 and A(3)=8 (the last two gray bars/columns).

From the formula we obtain:

E(5) = (3*A(4) + A(3)) / 4 = (3*6+8) / 4 = 26 / 4 = 6.5 person-days.

Thus:

a) FALSE

b) FALSE

c) CORRECT

d) FALSE

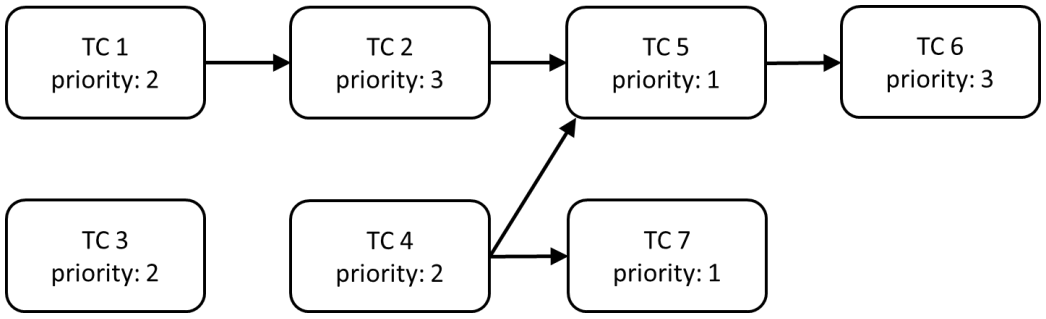
The answer c) is correct because the estimate of 6.5 person-days is consistent with the formula and the historical data.

Question 32	FL-5.1.5	K3	Score 1.0

You are preparing a test execution schedule for seven test cases (TC 1 to TC 7).

The figure shows: - the priority of each test case (1 = highest, 3 = lowest) - dependencies between the test cases in the form of directed arrows.

A test case may only be executed if all predecessor test cases on which it depends have already been executed.



The test execution schedule is to be created so that priority 1 test cases are executed as early as possible. For this purpose, predecessor test cases with lower priority may be scheduled before other already executable test cases with higher priority, if this is required to execute priority 1 test cases earlier.

Which test case should be executed sixth according to these constraints?

Please select ONE option! (1 out of 4)

a)	TC 3	☒
b)	TC 5	☐
c)	TC 6	☐
d)	TC 2	☐

FL-5.1.5 (K3) Apply test case prioritization.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.1.5)

We want to run test cases according to their priorities, but we also need to consider the dependencies.

If we only consider priorities, we want to first run TC 5 and TC 7 (highest priority), then TC 1, TC 3, and TC 4, and finally TC 2 and TC 6 (lowest priority).

However, in order to run TC 7, we need to first run TC 4.

In order to run TC 5, TC 1 and TC 2 must be executed first. TC 2, in turn, depends on TC 1. In order to run TC 7, TC 4 must be executed first.

So, in order to run priority 1 test cases as early as possible, the first five test cases should be: TC 4 - TC 7 - TC 1 - TC 2 - TC 5.

Next, we need to run TC 3, because it has higher priority than TC 6.

Thus the full schedule will be TC 4 – TC 7 – TC 1 – TC 2 – TC 5 – TC 3 – TC 6.

So, the sixth test case will be TC 3.

Thus:

a) CORRECT

b) FALSE

c) FALSE

d) FALSE

Question 33	FL-5.1.6	K1	Score 1.0

What does the test pyramid model show?

Please select ONE option! (1 out of 4)

a)	That tests may have different priorities.	☐
b)	That tests may have different granularity.	☒
c)	That tests may require different coverage criteria.	☐
d)	That tests may depend on other tests.	☐

FL-5.1.6 (K1) Recall the concepts of the test pyramid.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.1.6)

a) FALSE – The test pyramid model does not describe test priorities. It focuses on the hierarchy and granularity of tests, not on their prioritization (see [CTFL 4.0.2], section 5.1.6).

b) CORRECT – The test pyramid is a model showing that different tests can have different levels of granularity. It supports the team in test automation and in distributing the test effort; the pyramid levels represent groups of tests with different granularity, isolation, and typical execution time (see [CTFL 4.0.2], section 5.1.6, 1st sentence).

c) FALSE – Coverage criteria are not the focus of the test pyramid model. This model addresses test levels and granularity, not the coverage criteria of tests (see [CTFL 4.0.2], section 5.1.6).

d) FALSE – The test pyramid model does not describe dependencies between tests. It focuses on the structure and hierarchy of test levels (see [CTFL 4.0.2], section 5.1.6).

Question 34	FL-5.1.7	K2	Score 1.0

What is the relationship between the testing quadrants, test levels and test types?

Please select ONE option! (1 out of 4)

a)	Testing quadrants represent particular combinations of test levels and test types, defining their location in the software development lifecycle.	☐
b)	Testing quadrants describe the degree of granularity of individual test types performed at each test level.	☐
c)	Testing quadrants assign the test types that can be performed to the test levels.	☐
d)	Testing quadrants group test levels and test types by several criteria such as targeting specific stakeholders.	☒

FL-5.1.7 (K2) Summarize the testing quadrants and their relationships with test levels and test types.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.1.7)

a) FALSE – Testing quadrants group examples of test levels, test types, activities, test techniques, and work products according to criteria such as whether tests are business-facing or technology-facing and whether they support the team or critique the product. They do not assign these tests to a fixed location within the software development lifecycle.

b) FALSE – Testing quadrants group test levels and test types according to several criteria. They do not describe the degree of granularity of individual test types performed at each test level. Such a model, regarding the test levels, is called the test pyramid.

c) FALSE – The statement is wrong, because in general any test type can be performed at any test level.

d) CORRECT – The testing quadrants group test levels, test types, activities, test techniques and work products in Agile software development. In this model, tests can be business facing or technology facing. Tests can support the team (i.e., guide the development) or critique the product (i.e., measure its behavior against expectations). The combination of these two viewpoints determines the four quadrants.

Question 35	FL-5.2.3	K2	Score 1.0

Which of the following is an example of how product risk analysis may influence the thoroughness and scope of testing?

Please select ONE option! (1 out of 4)

a)	Continuous risk monitoring allows us to identify emerging risk as soon as possible.	☐
b)	Risk identification allows us to implement risk mitigation activities and reduce the risk level.	☐
c)	The assessed risk level helps determine the required thoroughness and scope of testing.	☒
d)	Risk analysis provides the basis for selecting technical test objects.	☐

FL-5.2.3 (K2) Explain how product risk analysis may influence thoroughness and scope of testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.2.3)

a) FALSE – Risk monitoring is part of risk control, not risk analysis.

b) FALSE – Risk identification itself does not allow us to implement risk mitigation activities. The mitigating actions are defined during the risk control phase.

c) CORRECT – This is an example of how risk analysis influences the thoroughness and scope of testing. ([CTFL 4.0.2], last Paragraph, 1st sentence)

d) FALSE – Coverage items are derived using test techniques, not through risk analysis.

Question 36	FL-5.3.2	K2	Score 1.0

In which of the following activities are data from test progress reports typically used?

Please select ONE option! (1 out of 4)

a)	Test design	☐
b)	Test completion	☒
c)	Test analysis	☐
d)	Test planning	☐

FL-5.3.2 (K2) Summarize the purposes, content, and audiences for test reports.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.3.2)

a) FALSE – Test progress reports are mostly used during test monitoring and control, and test completion, not during test design.

b) CORRECT – A test completion report is prepared during test completion, when a project, test level, or test type is complete and when, ideally, its exit criteria have been met. This report uses information from test progress reports and other data.

c) FALSE – Test progress reports are mostly used during test monitoring and control, and test completion, not during test analysis.

d) FALSE – Test progress reports are most used during test monitoring and control, and test completion, not during test planning.

Question 37	FL-5.4.1	K2	Score 1.0

Which of the following is NOT an example of how configuration management supports testing?

Please select ONE option! (1 out of 4)

a)	All configuration items in the repository are uniquely identified, version controlled, and changes are tracked.	☐
b)	All changes in the test environment elements are tracked.	☐
c)	All requirement specifications are referenced unambiguously in test plans.	☐
d)	All identified defects have an assigned status.	☒

FL-5.4.1 (K2) Summarize how configuration management supports testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.4.1)

a) FALSE – When a user reports a software failure, thanks to the unique identification of commits, it is possible to reassemble the files from the software version which was used by the user (as well as the corresponding versions of the test scripts) and thus reproduce the failure and locate the defect faster.

b) FALSE – If a change to the test environment causes unexpected issues during testing, configuration management allows testers to roll back to a previous version of the environment. This ensures that testing can continue without being affected by the change.

c) FALSE – Configuration management ensures that all identified documentation (e.g., requirement specifications) and software items are referenced unambiguously in test documentation (e.g., test plans).

d) CORRECT – This is ensured by defect management, not by configuration management.

Question 38	FL-5.5.1	K3	Score 1.0

Consider the following defect report for a web-based shopping application:

Application: WebShop v0.99

Defect: Login button not working

Steps to Reproduce:

Launch the website

Click on the login button

Expected result: The user should be redirected to the login page.

Actual result: The login button does not respond when clicked.

Severity: High

Priority: Urgent

What is the MOST important missing information for reproducing this defect?

Please select ONE option! (1 out of 4)

a)	Name of the tester and date of the report.	☐
b)	Test environment elements and their version numbers.	☒
c)	Identification of the test object.	☐
d)	Impact on the interests of stakeholders.	☐

FL-5.5.1 (K3) Prepare a defect report.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 5.5.1)

a) FALSE – This is important, but not as important as test environment elements.

b) CORRECT – The important thing that is missing is the identification of the browser and device used for the testing. The browser and device information are important because such a defect can be browser- or device-specific. For example, a login button may work fine on one browser (or one version of a specific browser) but not on another. Therefore, the browser and device information can help the developers to reproduce the issue and find the root cause of the problem more quickly.

c) FALSE – The test object is identified (WebShop v0.99).

d) FALSE – A description of the impact on stakeholder interests may be useful for assessing and prioritizing a defect report, but for immediate reproduction of this defect, test environment information is more important.

Question 39	FL-6.1.1	K2	Score 1.0

Which of the following tool categories supports the management of requirements, tests, defects, and configuration?

Please select ONE option! (1 out of 4)

a)	Test execution and coverage tools	☐
b)	Test design and implementation tools	☐
c)	Defect management tools	☐
d)	Test management tools	☒

FL-6.1.1 (K2) Explain how different types of test tools support testing.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 6.1.1)

a) FALSE – Test execution and coverage tools facilitate the automated execution of test cases and the measurement of the coverage achieved by running those test cases. However, these tools do not help with the organization of defects and configuration management.

b) FALSE – Test design and implementation tools facilitate the generation of test cases, test data and test procedures, but they do not help with the organization of defects and configuration management.

c) FALSE – Defect management tools support the recording, tracking, and management of defect reports, but they do not provide the comprehensive management of requirements, tests, defects, and configuration described in the question.

d) CORRECT – Test management tools increase the test process efficiency by facilitating the management of the software development lifecycle (SDLC), requirements, tests, defects, and configuration management.

Question 40	FL-6.2.1	K1	Score 1.0

Which of the following is a typical benefit of test automation?

Please select ONE option! (1 out of 4)

a)	The capability of generating test cases without access to the test basis.	☐
b)	The achievement of increased coverage through more objective assessment.	☐
c)	The increase in test execution times available with higher processing power.	☐
d)	The prevention of simple human errors through consistent and repeatable test execution.	☒

FL-6.2.1 (K1) Recall the benefits and risks of test automation.

Justification: (see ISTQB® Foundation Level Syllabus v4.0.2; Section 6.2.1)

a) FALSE – The automated generation of test cases without access to the test basis is not a typical benefit of test automation. Meaningful test cases should be derived from, or be traceably linked to, the test basis.

b) FALSE – More objective assessment of results can be a benefit of test automation, but increased coverage is not achieved simply through objective assessment. Coverage can only increase when additional relevant tests are designed and executed.

c) FALSE – Longer execution times are not a benefit of test automation. Higher processing power would normally reduce execution time rather than increase it.

d) CORRECT – Preventing simple human errors through consistent and repeatable test execution is a typical benefit of test automation. Automated tests can be executed in the same way and with the same frequency, reducing variability in execution.

Space for your notes:
(are neither read nor valuated during correction)

Space for your notes:
(are neither read nor valuated during correction)