DUMPSARENA

ISTQB Certified Tester Foundation Level (Syllabus 2018)

ISTQB CTFL_Syll2018

Version Demo

Total Demo Questions: 41

Total Premium Questions: 414

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Fundamentals of Testing	60
Topic 2, Testing Throughout the Software Development Lifecycle	49
Topic 3, Static Testing	36
Topic 4, Test Techniques	131
Topic 5, Test Management	103
Topic 6, Tool Support for Testing	35
Total	414

QUESTION NO: 1

A system is being enhanced to simplify screen navigation for users.

Which of the following does NOT reflect structural testing?

A. To test all paths that users could take through the screen menu system

B. To ensure that 100% decision testing is achieved for each system component

C. To test all branches of component calls within the application call graph

D. To ensure that users can navigate to all fields on the screen

ANSWER: D

Explanation:

“To ensure that users can navigate to all fields on the screen” does not reflect structural testing because it assesses whether the delivered user interface provides the required navigability from a user’s perspective. The test focus is on externally observable behavior: whether a person can reach and use every field by means such as keyboard navigation, tab order, controls, or other supported interaction mechanisms. This is a functional and potentially usability-oriented concern, rather than a test derived from the internal implementation structure.

Structural testing designs tests from the way the software is built, such as its control flow, decisions, paths, calls, or other internal structures, and commonly evaluates coverage against those structures. In contrast, confirming access to all screen fields can be specified and assessed without knowing how the application’s code, components, or control-flow logic is organized. It therefore fits black-box testing of a navigation requirement and user interaction behavior. ISTQB describes structural testing as a white-box approach based on the internal structure of a component or system; see the ISTQB Certified Tester Foundation Level certification information and the ISTQB Glossary entry for structural testing.

QUESTION NO: 2

Which of the following is a typical objective of system testing?

A. To evaluate the behaviour of the complete and integrated system against specified requirements

B. To test individual functions in isolation from all dependencies

C. To verify that a repaired defect has been removed

D. To determine whether users will accept the system for operational use

ANSWER: A

Explanation:

To evaluate the behaviour of the complete and integrated system against specified requirements is correct. System testing focuses on the behaviour and capabilities of the complete system. It commonly includes both functional and non-functional testing and can examine end-to-end tasks, interfaces, data handling, security, performance, and other specified quality characteristics.

Component testing focuses on individual components, while integration testing focuses on interfaces and interactions between integrated components or systems. Acceptance testing focuses on whether the system is suitable for acceptance and intended use. See the ISTQB Certified Tester Foundation Level resources.

QUESTION NO: 3

Which of the following is a correct statement about informal reviews?

A. The usefulness of an informal review is highly variable and strongly dependent on the reviewers

B. Informal review must be done by the author of the reviewed material, before it is submitted for formal review

C. The moderator of an informal review should not be the author of the reviewed material

D. An informal review has no impact on the quality of the product, just on the level of knowledge of the reviewers while Structural tests can be performed at Unit test level

ANSWER: A

Explanation:

The usefulness of an informal review is highly variable and strongly dependent on the reviewers is correct. Informal reviews are the least structured type of static review: they do not require a defined process, formally assigned roles, documented entry or exit criteria, or prescribed metrics. They may range from a quick discussion between colleagues to an author asking a peer to read a work product and provide feedback. Because the process is flexible, its effectiveness depends substantially on the people involved. Reviewers need sufficient technical, business, and testing knowledge to identify meaningful defects, ambiguities, omissions, or maintainability concerns. Their diligence, independence of perspective, available time, and willingness to provide constructive feedback also influence the value obtained. Informal reviews can therefore be very effective when knowledgeable reviewers examine material carefully, but their results are less predictable than those of more formally managed review types. The CTFL syllabus describes informal reviews as having no formal process and notes that their usefulness varies depending on the reviewers. This aligns with the syllabus's broader treatment of reviews as static testing activities that can detect defects early in work products. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB CTFL syllabus resources.

QUESTION NO: 4

What is a peer review?

A. An informal review without moderator

B. A review done by developers and testers together

C. A review done by developers only

D. A review done by colleagues at the same organizational level

ANSWER: D

Explanation:

A peer review is a review of a work product performed by the producer's peers: colleagues who have a comparable organizational position and sufficient relevant knowledge to examine the work product. Therefore, "A review done by colleagues at the same organizational level" is correct. The term describes the relationship of the reviewers to the author rather than restricting the review to a particular job role, document type, or review formality. Peer reviews can be applied to many work products, such as requirements, test plans, test cases, code, and user documentation. Their purpose is to identify defects, omissions, ambiguities, and opportunities for improvement early, when correction is usually less costly than after implementation or test execution. ISTQB's glossary defines a peer review as a review performed by the work-product producer's peers, which directly supports this interpretation. The CTFL syllabus also presents reviews as a static testing activity and emphasizes that appropriate reviewer selection, including relevant expertise and independence, supports effective defect detection. See the official ISTQB Glossary definition of peer review and the ISTQB Certified Tester Foundation Level resources.

QUESTION NO: 5

Which of the following apply to System Testing?

a) May satisfy legal requirements.

b) Can use system specifications as a test basis.

c) Often the responsibility of business users.

d) Main goal is to establish confidence.

e) Should focus on the communication between systems.

A. a and c.

B. b and d.

C. a and b.

D. c and e.

ANSWER: B

Explanation:

b and d. System testing evaluates the behavior and capabilities of the complete, integrated system. A central test basis at this level is the system requirements specification, often referred to in the question as system specifications. Testers derive conditions and test cases from these specifications to check whether the delivered system provides the required functional and non-functional behavior, including complete end-to-end workflows where relevant.

Establishing confidence is also an important objective of system testing. By exercising the complete system against its specified requirements, system testing provides stakeholders with evidence about the system's quality and its readiness to proceed to subsequent testing or release activities. This confidence is not a guarantee that no defects remain; rather, it is confidence based on the test evidence obtained, the coverage achieved, and the assessed residual risk.

The ISTQB Foundation Level syllabus describes system testing as testing of the behavior and capabilities of a complete system and identifies system requirements specifications as a typical test basis. It also states that testing can have objectives such as building confidence in the quality of the system. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB downloads page for the official syllabus.

QUESTION NO: 6

Choose below the best characterization of the difference between black-box testing and white-box testing

A. For black-box testing, the testers do not need to know the test object in advance

B. For white-box testing, the testers have to consult specifications and developers before running any tests

C. Black box testing uses static analysis. White box testing uses dynamic analysis

D. Black-box testing is based on the test basis documentation. White-box testing is based on an analysis of the code itself.

E. Black-box testing is based on an analysis of the code itself White-box testing is based on the test basis.

ANSWER: D

Explanation:

Black-box testing is based on the test basis documentation. White-box testing is based on an analysis of the code itself. This accurately captures the defining distinction in the ISTQB Foundation Level syllabus. Black-box test techniques derive test conditions, test cases, and test data from externally observable information about the test object. The test basis can include requirements specifications, user stories, acceptance criteria, business-process models, use cases, and interface documentation. The tester focuses on what the system is expected to do, without needing knowledge of its internal implementation.

White-box testing, in contrast, derives tests from the internal structure of the component or system. For software, this commonly means analysing source code, control flow, decisions, conditions, paths, and statement structure, then designing tests to achieve an appropriate structural coverage target. Thus, code analysis is central to white-box test design, whereas documented intended behaviour is central to black-box test design. Both approaches are useful and complementary: externally derived tests provide confidence in specified behaviour, while structurally derived tests provide evidence about

exercised implementation elements. The terminology and technique classifications are defined in the ISTQB Certified Tester Foundation Level materials and the ISTQB Glossary definition of black-box testing.

QUESTION NO: 7

What of the following statements regarding walkthrough are NOT true?

A. A walkthrough is lead by a moderator and requires pre-meeting preparation and post meeting follow up

B. A walkthrough may take the form of scenarios, dry runs and peer group review

C. The purpose of a walkthrough is to establish common understanding of a document and to find possible defects in it

D. Pre-meeting preparation, documentation and distribution of results are optional for a walkthrough

ANSWER: A

Explanation:

A walkthrough is lead by a moderator and requires pre-meeting preparation and post meeting follow up is the statement that is not true. In the ISTQB Foundation Level syllabus, the defining characteristic of a walkthrough is that it is led by the author of the work product, rather than by an independent moderator. The author presents the document or other work product to participants, typically to build a shared understanding, gather feedback, and identify potential defects. Walkthroughs may use practical approaches such as scenarios or dry runs, which help participants follow the work product in a realistic way and clarify intended behavior.

The formality of a walkthrough is flexible. Individual preparation is optional, and recording findings, actions, decisions, or recommendations is also optional. Consequently, walkthrough activities can be tailored to the review objective and the team's needs; they are not defined by mandatory pre-meeting preparation and follow-up administration. The key point is the author-led nature of the review and its support for shared understanding and defect discovery. This is described in the review-types material in the ISTQB Certified Tester Foundation Level syllabus resources and in ISTQB's walkthrough glossary entry.

QUESTION NO: 8

Which of the following test cases will ensure that the statement â€˜Print â€˜Holdâ€™â€™ is exercised? [K3]

Refer to the exhibit

A. X=2, Y=2, Z=2

B. X=2, Y=3, Z=4

C. X=2, Y=4, Z=3

D. X=4, Y=3, Z=2

ANSWER: D

Explanation:

X=4, Y=3, Z=2 exercises the *Print 'Hold'* statement because it satisfies the decision conditions needed to reach that statement: X is greater than Y, and Z is less than Y. Substituting the test data gives 4 > 3 and 2 < 3. Since both conditions evaluate to true, execution follows the path containing the required print statement, ensuring that the statement is executed at least once.

This is a statement-coverage objective: a test case must cause the executable statement of interest to run. The test data should therefore be evaluated against the predicates controlling the relevant path, rather than merely checking that the input values are distinct or near a boundary. Selecting values that make every required condition true provides direct evidence that the path leading to *Print 'Hold'* is traversed. This application of control-flow reasoning is consistent with the ISTQB Foundation Level treatment of white-box testing and statement coverage, where coverage is measured by executed

statements relative to all executable statements. See the ISTQB Certified Tester Foundation Level certification page and the CTFL syllabus.

QUESTION NO: 9

Which two of the following statements are true?

A. Software testing can be necessary to fulfill legal or contractual obligations.

B. Software testing is done to essentially increase the quality of developers.

C. Intensive testing can help reduce risks and problems during operation

D. Intensive testing is used to prove correctness.

E. C and D are true. A and B are wrong.

F. A and C are true. B and D are wrong

G. B and D are true. A and C are wrong

H. B and C are true, A and D are wrong

ANSWER: F

Explanation:

The correct selection is the one pairing “Software testing can be necessary to fulfill legal or contractual obligations” with “Intensive testing can help reduce risks and problems during operation.” The ISTQB Foundation Level syllabus identifies several general testing objectives, including fulfilling contractual, legal, or regulatory requirements. For example, testing may be required to demonstrate that a product complies with regulated-domain obligations, safety expectations, data-handling requirements, or explicit acceptance criteria agreed with a customer.

The syllabus also explains that testing provides information that helps reduce the risk of inadequate software quality. Testing cannot eliminate every possible failure, but sufficiently thorough and well-targeted testing can reveal defects and assess important product risks before release. Addressing the resulting information reduces the likelihood and impact of failures in production, thereby helping avoid operational problems. The amount and focus of testing should be selected according to the risks associated with the product and its intended use. These purposes reflect testing as a risk-management and assurance activity, rather than an activity that can establish absolute correctness.

See the ISTQB Certified Tester Foundation Level certification page and the ISTQB Glossary for ISTQB terminology and syllabus resources.

QUESTION NO: 10

Which of the following are triggers for Maintenance testing?

a) System migration from one platform to another.

b) Retirement of a system.

c) Preparation for an audit of a system.

d) Modifications to a system.

e) Development of a whole new system.

A. a, c and d.

B. b, c and e.

C. a, d and e.

D. a, b and d.

ANSWER: D

Explanation:

a, b and d. is correct because maintenance testing is performed on an existing operational system when a change or an end-of-life activity creates a need to assess continued quality or to manage associated risks. System migration from one platform to another is a maintenance trigger because the existing application is being moved into a changed technical environment. Testing is needed to confirm that the system and its data continue to operate correctly after the migration. Retirement of a system is also a maintenance-testing trigger: testing may be required for data migration, archiving, retention, accessibility, or controlled decommissioning activities. Modifications to a system directly trigger maintenance testing, as the changed functionality and potentially affected unchanged areas need appropriate testing, including regression testing where relevant.

The ISTQB Foundation Level syllabus identifies modification, migration, and retirement as typical triggers for maintenance testing. This classification focuses on changes to, movement of, or withdrawal of an existing system rather than the initial development lifecycle of a new system. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB syllabi and standards library for the Foundation Level syllabus material on maintenance testing.

QUESTION NO: 11

Which of the following is correct?

A. Intrusive test tools are tools that do not exhibit the probe-effect

B. Testing tools can be used by both developers and testers

C. Use of testing tools is effective only when done as part of a test automation system

D. Testing tools allow developers do testing Use of such tools changes the role of the test team

ANSWER: B

Explanation:

Testing tools can be used by both developers and testers. The ISTQB Foundation Level syllabus describes a broad range of tool support for testing activities, including static testing, test design and implementation, test execution and logging, performance measurement, and test management. These activities involve both development and testing roles. For example, developers may use static analysis tools, unit-test frameworks, code-coverage tools, debuggers, and continuous-integration systems, while testers may use test-management, defect-management, test-data-preparation, and test-execution tools. In collaborative delivery approaches, the distinction is even less rigid: developers and testers can both contribute to automated checks, investigate failures, review requirements, and use shared tooling to provide rapid feedback on product quality. The important principle is that tools support people and processes; their successful use requires suitable objectives, skills, training, processes, and maintenance. Tool usage is therefore not confined to a dedicated test team or to a complete test automation system. This aligns with ISTQB's treatment of tool support as applicable across the test process and to different project participants. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB CTFL syllabus resources.

QUESTION NO: 12

Which of the following test execution outcomes are likely to increase the overall quality of the software:

I) A test case passes

II) A test case fails, defect gets logged which subsequently gets fixed

III) Some tests are deferred because test environment is not available

IV) A bug is found but there is no corresponding test case

A. I, II and III

B. I and IV

C. I and II

D. I, II and IV

ANSWER: C

Explanation:

The correct combination is *I and II*. A test case passes provides objective evidence that the implemented software behaviour conforms to the specified expected result for the conditions covered by that test. This contributes to confidence in the product's current quality, because the test has verified that the relevant requirement or behaviour works as intended. The confidence is necessarily limited to the tested scope, but successful execution is still useful quality information.

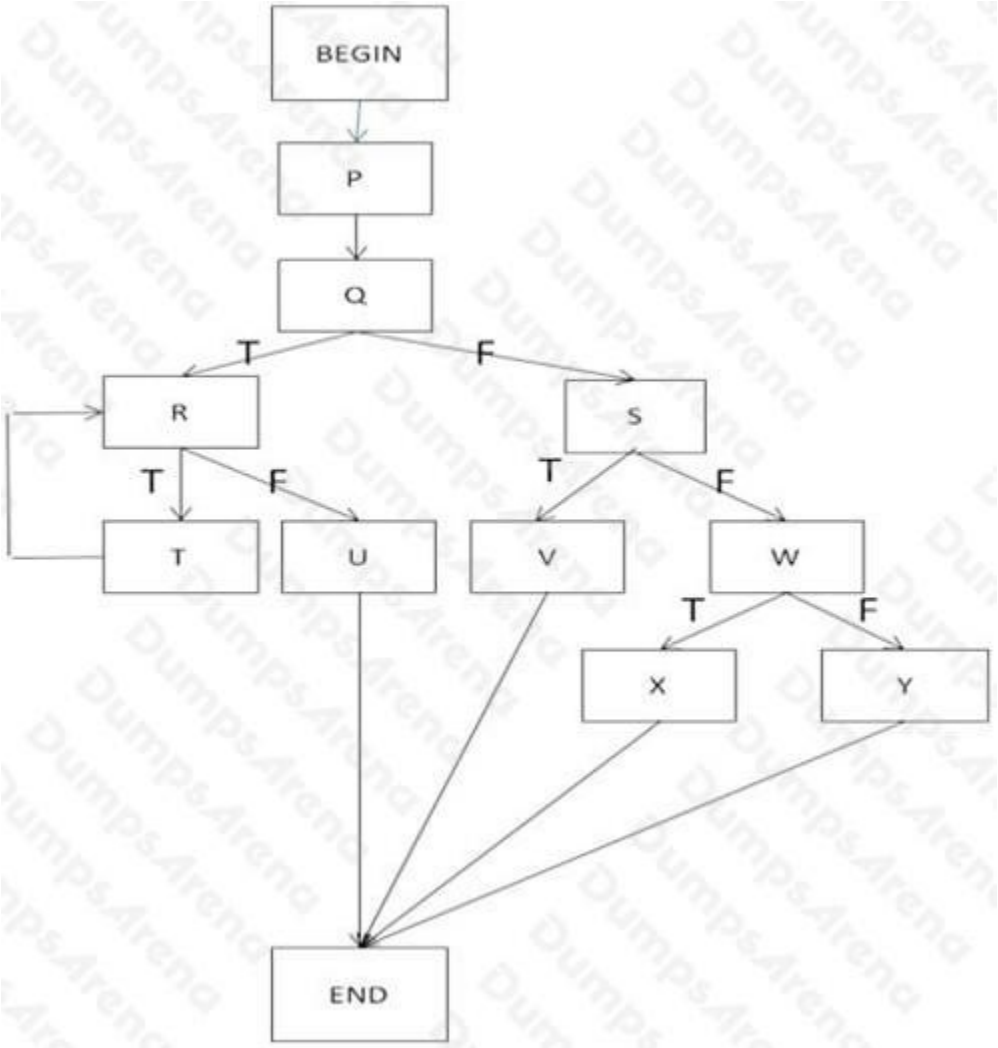
A test case fails, defect gets logged which subsequently gets fixed contributes more directly to improving quality. The failed test reveals a discrepancy between actual and expected behaviour. Recording the defect enables it to be managed, analysed, corrected, and subsequently confirmed through re-testing. Once the correction is verified, the software contains one fewer known defect and therefore has improved quality. The test case should also be retained as a regression test, helping detect recurrence of the same defect in later changes.

This reflects the CTFL view that testing provides information about product quality and can help reduce the risk of defects remaining in the delivered software. See the official ISTQB Certified Tester Foundation Level certification page and the ISTQB glossary definition of defect.

QUESTION NO: 13

The flow graph below shows the logic of a program for which 100% statement coverage and 100% decision coverage is required on exit from component testing. [K4]

The following test cases have been run: Test Case 1 covering path P,Q,R,U Test Case 2 covering path P,Q,S,V Test Case 3 covering path P,Q,S,W,X Test case 4 covering path P,Q,S,W,Y Refer to the exhibit



A. Statement coverage is 100%; decision coverage is 100%

B. Statement coverage is less than 100%; decision coverage is 100%.

C. Statement coverage is 100%; decision coverage is less than 100%

D. Statement coverage and decision coverage are both less than 100%

ANSWER: A

Explanation:

Statement coverage is 100%; decision coverage is 100% is correct because the combined set of executed paths visits every statement node represented in the flow graph: P, Q, R, U, S, V, W, X, and Y. Statement coverage measures the proportion of executable statements that have been exercised by at least one test. Since every node is included in one or more of the four listed paths, all statements have been executed.

The test set also exercises every available outcome from each decision. At Q, execution continues through both the R route and the S route. At S, both the V route and the W route are exercised. At W, execution reaches both X and Y. Thus, each decision has had all of its possible outcomes taken at least once. This satisfies 100% decision coverage, which ISTQB defines in terms of exercising the decision outcomes. The paths collectively therefore meet both stated component-test exit criteria. ISTQB describes coverage measures and decision testing in its Certified Tester Foundation Level materials; its terminology is also available through the ISTQB Glossary.

QUESTION NO: 14

The following test cases for a Library Management System are available to test changes made to the functions and data structures associated with borrowers

1. Add a new borrower to the system

2. Update a borrower's data

3. Remove a borrower from the system

4. Loan a book to a borrower

5. Return a book from a borrower

6. Reserve a book for a borrower

7. Send "reservation ready" message to a borrower

Which of the following test sequences represents a possible use case? (a test sequence always start with test #1)

A. 1-4-2-7-5-6-3

B. 1-6-2-5-7-4-3

C. 1-6-4-7-5-3-2

D. 1-2-6-7-4-5-3

ANSWER: D

Explanation:

The sequence **1-2-6-7-4-5-3** represents a coherent borrower lifecycle and therefore a possible use case. It begins by creating the borrower record, which establishes the necessary system entity for all later borrower-related actions. The borrower's details can then be updated while the account is active. Next, the borrower reserves a book; once that reservation can be fulfilled, the system sends the reservation-ready notification. The borrower can subsequently loan the available reserved book and later return it. Finally, the borrower is removed only after the borrowing activity has been completed, leaving no active loan associated with the account.

This follows the use-case perspective described in the ISTQB Foundation Level syllabus: a use case models interactions that achieve a user goal through a meaningful sequence of actions, including valid preconditions and state changes. Here, each operation has a sensible predecessor and produces a system state that supports the next operation. In particular, a reservation exists before a reservation-ready message is sent, and the loan is completed through a return before the borrower record is deleted. See the ISTQB Certified Tester Foundation Level overview and the ISTQB Glossary entry for "use case".

QUESTION NO: 15

Which of the following is a task of the Test Analysis and Design activity of the test process?

A. Measuring the percentage of prepared test cases with what was actually prepared

B. Identifying necessary test data to support the test conditions and test cases

C. Verifying that the test environment has been set up correctly

D. Checking test logs against the exit criteria specified in test planning

ANSWER: B

Explanation:

Identifying necessary test data to support the test conditions and test cases is a Test Analysis and Design task. In the ISTQB Foundation Level test process, test analysis transforms the test basis into testable conditions, while test design develops test

cases and related testware that will exercise those conditions. Test data is an essential part of this work because a test case must specify or make available the input values, preconditions, database records, files, user accounts, and other data needed to perform the test and obtain meaningful actual results. The required data is selected so that it supports coverage of the identified test conditions and allows the test procedure to be executed reliably and repeatably. This activity also considers any constraints on data availability, such as the need for representative production-like data, masked personal data, boundary values, invalid values, or data combinations needed by the chosen test techniques. Identifying test data during analysis and design ensures that execution is not delayed later by missing inputs or unsuitable records. The ISTQB Foundation Level syllabus describes test analysis and test design as activities that derive test conditions and test cases, including the test data needed for them. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB Glossary definition of test data.

QUESTION NO: 16

A tester thinks of a likely cause for a specific bug Should the tester make a comment about this in the bug report?

A. No. A bug r^&ort must only include factual information, and not unsupported hypothesis

B. No. Such addition may bias the developers' attitude when they attempt to fix the bug

C. Yes It will reduce the risk that the bug fix will cause a regression

D. Yes Observations that may help correct a bug should be included in the bug report

ANSWER: D

Explanation:

“Yes Observations that may help correct a bug should be included in the bug report” is correct because a defect report is a communication tool intended to give the people investigating and resolving a defect all useful information available to the tester. Alongside the observed result, expected result, reproduction details, test environment, severity, and supporting evidence, relevant observations can help developers understand the circumstances of the failure and investigate it more efficiently.

A tester’s suspected cause should be clearly presented as an observation or hypothesis, rather than as an established fact. For example, the report can state that the problem appears to occur after a particular configuration change, only with a certain input pattern, or after a specific sequence of actions. This gives the development team potentially valuable diagnostic context while preserving the distinction between the tester’s evidence and the eventual root-cause analysis.

The CTFL syllabus describes defect reports as containing information that supports investigation and resolution, including details that help reproduce and analyze the failure. This supports recording relevant observations that may assist correction. See the ISTQB Certified Tester Foundation Level documentation and the ISTQB Glossary entry for defect report.

QUESTION NO: 17

Consider the following excerpt from a test report 'We tested the module AX45T and we got only 1 failure. To recreate the failure, we used test file PR624_DR78_TCB0022 pkr, available under the PR624_DR78 repository in the CK01 server".

Which incident report objective does this excerpt satisfy?

A. Provides developers with information to isolate the failure.

B. Does not belong to an incident report

C. Provides Project Managers with information on the project risks.

D. Provides test managers with information to report test progress.

ANSWER: A

Explanation:

The excerpt satisfies the objective “Provides developers with information to isolate the failure.” It identifies the affected module, states that a failure occurred, and, most importantly, supplies reproducibility information: the specific test file, repository, and server location used to recreate the observed failure. This information enables a developer or other technical investigator to establish the same conditions, observe the failure consistently, collect diagnostic evidence, and narrow the potential source of the defect. A useful incident report supports efficient investigation and resolution by recording sufficient details about the circumstances in which an anomaly was observed. Reproducible test data and an identifiable execution environment are especially valuable because they reduce ambiguity and avoid relying on the tester’s memory or an informal verbal description. The CTFL syllabus describes incident reports as a means of communicating anomalies found during testing to stakeholders who need to investigate and resolve them; the report should contain information that supports analysis, such as the observed result and relevant test context. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB glossary entry for incident report.

QUESTION NO: 18

Which of the following is a valid collection of equivalence classes for the following problem: "An Integer numeric field shall contain values from 1 to 80 both values inclusive"

A. Less than 0. 1 to 79, 80 and more than 80

B. Less than 0. 1 to 80, more than 80

C. Less than 1. 1 to 80, more than 80

D. Less than 1. 1 to 79, more than 80

ANSWER: C

Explanation:

“Less than 1. 1 to 80, more than 80” is the valid collection of equivalence classes because it partitions the complete integer input domain into groups that are expected to receive the same treatment. The requirement defines every integer from 1 through 80, including both endpoints, as valid. Therefore, the valid equivalence class is 1 to 80. Inputs below the lower boundary form one invalid class: less than 1. Inputs above the upper boundary form the other invalid class: more than 80.

The three classes are mutually exclusive: no integer can belong to more than one of them. They are also collectively exhaustive for integer input: every possible integer is either below 1, within the permitted inclusive range, or above 80. This makes the set suitable for equivalence partitioning, since a representative value can be selected from each class to test the expected acceptance or rejection behavior. The boundaries 1 and 80 are correctly included in the valid class, reflecting the phrase “both values inclusive.” ISTQB describes equivalence partitioning as dividing data into partitions that are assumed to be processed in the same way; boundary values are then especially relevant when applying boundary value analysis. See the ISTQB Certified Tester Foundation Level materials and the ISTQB glossary entry for equivalence partitioning.

QUESTION NO: 19

Which of the following would be appropriate test objectives for user acceptance testing of the first release of a new software product aimed at a general market and built using Agile methods? [K2]

a. To identify as many defects as possible

b. To maximise code coverage

c. To ensure the product works as expected

d. To assess the overall quality of the product

e. To determine the reliability of the product

A. b and c

B. a and d

C. b and e

D. c and d

ANSWER: D

Explanation:

c and d is correct because user acceptance testing focuses on gaining confidence that a product is suitable for its intended users and purposes before it is released to the market. For a first release aimed at a general market, representative users or customer representatives need to confirm that the product supports expected real-world use and delivers the value users expect. Therefore, ensuring that the product works as expected is a core acceptance-testing objective.

User acceptance testing also provides a broad, user-oriented assessment of whether the product is ready for use. This includes considering the product's overall quality from the user perspective, such as whether important workflows are usable, appropriate, and sufficiently complete for release. In an Agile context, acceptance criteria are evaluated throughout development, while user acceptance testing can provide final confidence that the integrated product meets user needs and is acceptable for deployment.

This aligns with the ISTQB definition of acceptance testing as testing performed to determine whether a system satisfies acceptance criteria and to enable a user or other authorized entity to decide whether to accept it. See the ISTQB Glossary: acceptance testing and the ISTQB Certified Tester Foundation Level syllabus resources.

QUESTION NO: 20

A program is used to control a manufacturing line (turn machines on and off. start and stop conveyor belts,

add raw materials to the flow, etc.)

Not all actions are possible at all times For example, there are certain manufacturing stages that cannot be stopped - unless there is an emergency. A tester attempts to evaluate if all such cases (where a specific action is not allowed) are covered by the tests. Which coverage metric will provide the needed information for this analysis?

A. Branch Coverage

B. Statement coverage

C. Data flow coverage

D. Code coverage

E. Invalid transition coverage

ANSWER: E

Explanation:

Invalid transition coverage provides the needed information because the manufacturing controller's permitted actions depend on its current state. For example, when a manufacturing stage is in a non-stoppable state, an ordinary stop command represents an invalid transition; an emergency stop may represent a separately permitted transition. The test objective is specifically to determine whether tests exercise and check every action that the system must reject in each relevant state.

State transition testing models the system as states and transitions caused by events or actions. Its coverage measures can address valid transitions, states, and invalid transitions. Invalid transition coverage measures the extent to which tests have attempted transitions that are not allowed from a given state, which directly matches the need to assess prohibited actions in the manufacturing-line workflow. Such tests should verify that the controller prevents the prohibited action and maintains a safe, specified condition.

The ISTQB Foundation Level syllabus identifies state transition testing as useful where system behavior depends on prior events or conditions, and it includes invalid transitions among the relevant test-design considerations. See the ISTQB Certified Tester Foundation Level resources and the ISTQB glossary entry for invalid transition.

QUESTION NO: 21

For a mandatory input field 'ZIP code" the following rules are given:

1 - The valid ZIP code format is 5 numeric digits

2 - The code has to exist in the pest office's official ZIP code list

Using equivalence classes partitioning, how many test cases are required to test this field?

A. 6

B. 4

C. 8

D. 3

ANSWER: A

Explanation:

6 is correct because equivalence partitioning identifies groups of inputs that should be handled in the same way, so one representative test is selected from each relevant valid or invalid class. Since the field is mandatory, an empty value is one invalid equivalence class. The format rule creates separate invalid classes for a value that is too short, a value that is too long, and a value that does not consist of numeric digits in the required five-digit form. A five-digit numeric value is then subject to the official-list rule, which creates two further classes: a code that exists in the ZIP-code list and is accepted, and a syntactically valid code that does not exist in the list and is rejected. This gives six classes and therefore six representative test cases. The aim is not to test every possible ZIP code, but to test a representative value from each class for which the system is expected to behave equivalently. This follows the CTFL definition of equivalence partitioning as dividing data into partitions expected to be processed in the same way. See the ISTQB Certified Tester Foundation Level information and the ISTQB glossary definition of equivalence partitioning.

QUESTION NO: 22

Which of the following represents good testing practice for testers, irrespective of the software lifecycle model used?

A. They should start test analysis when the corresponding development level is complete

B. They should be involved in reviewing requirements or user stories as soon as drafts are available

C. They should ensure that the same test objectives apply to each test level

D. They should minimize the ratio of development levels to test levels to reduce project costs

ANSWER: B

Explanation:

They should be involved in reviewing requirements or user stories as soon as drafts are available. Early tester involvement is a fundamental good practice because it allows testability, completeness, consistency, ambiguity, and acceptance criteria to be assessed before implementation begins. Finding and resolving defects in requirements or user stories at this point prevents those defects from being propagated into design, code, and later test artefacts, where they are generally more expensive and disruptive to correct. This practice is applicable regardless of whether the team uses a sequential, iterative, incremental, or agile lifecycle model: the artefact may be called a requirement, specification, epic, or user story, but it should be reviewed as early as practical. Tester participation also helps establish a shared understanding of expected behaviour and supports the creation of effective test conditions and tests early in the lifecycle. The CTFL syllabus describes testing activities as being tailored to the chosen software development lifecycle and emphasizes early involvement in reviewing work products to detect defects as soon as possible. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB Glossary for official terminology and syllabus resources.

QUESTION NO: 23

Which of the following are product risks?

a) Failure prone software delivered.

b) Software does not perform its intended functions.

c) Insufficient staff available for testing.

d) Test environment not ready on time.

e) Poor data integrity and quality.

A. b, c and e.

B. b, d and e.

C. a, b and e.

D. b, c and d.

ANSWER: C

Explanation:

a, b and e. is correct because product risks are potential problems associated with the quality characteristics of the software product itself. They represent ways in which the delivered system could fail to meet the needs of users, customers, or other stakeholders. Failure-prone software delivered is a product risk because unreliable behaviour directly affects operational use and user confidence. Software that does not perform its intended functions is also a product risk: functional suitability is a fundamental aspect of product quality, and failure to provide required behaviour can prevent business objectives from being achieved. Poor data integrity and quality is a product risk because incorrect, incomplete, inconsistent, or corrupted data can produce incorrect outcomes and undermine the trustworthiness of the system.

The CTFL syllabus distinguishes these risks from project risks by focusing product risks on potential adverse events related to the product's quality. Identifying such risks supports risk-based testing: the team can prioritize test design, test execution, and test depth in the areas where product failure would have the greatest impact. See the ISTQB Certified Tester Foundation Level certification information and the ISTQB standards and syllabus resources.

QUESTION NO: 24

Decision table testing is being performed on transactions in a bank's ATM (Automated teller Machine) system. Two test cases have already been generated for rules 1 and 4, which are shown below:

Given the following additional test cases:

Which two of the additional test cases would achieve fill coverage of the full decision table (when combined with the test cases that have already been generated for rules a and 4?

A. DT1, DT2

B. DT2, DT3

C. DT3, DT4

D. DT1, DT4

ANSWER: D

Explanation:

DT1, DT4 is correct because decision table testing derives tests from the rules represented by combinations of conditions and their associated actions. Full decision-table coverage requires that every feasible rule in the table be exercised by at

least one test case. The two tests already generated for rules 1 and 4 provide coverage for their respective condition combinations. Adding DT1 supplies the missing combination for a checking account where no amount is entered, producing the error action. Adding DT4 supplies the missing combination for a savings account with the relevant higher entered amount, producing the corresponding error action. Together with the previously generated tests, these selections cover the remaining rule combinations required by the complete table. This is the intended application of decision table testing: identify condition combinations, represent each combination as a rule, and select tests so that the required set of rules is covered. ISTQB defines decision table testing as a black-box technique based on a decision table that records combinations of inputs and resulting actions. See the ISTQB glossary definition of decision table testing and the ISTQB Certified Tester Foundation Level syllabus information.

QUESTION NO: 25

In foundation level syllabus you will find the main basic principles of testing, Which of the following sentences describes one of these basic principles?

A. Complete testing of software is attainable if you have enough resources and test tools

B. For a software system, it is not possible under normal conditions, to test all input and output combinations.

C. A goal of testing is to show that the software is defect free

D. With automated testing you can make statements with more confidence about the quality of a product than with manual testing.

ANSWER: B

Explanation:

For a software system, it is not possible under normal conditions, to test all input and output combinations. This expresses the fundamental testing principle that exhaustive testing is impossible. Real software can have enormous numbers of possible inputs, states, paths, data values, environmental configurations, and sequences of user actions. Even a seemingly simple feature may behave differently depending on prior activity, integration points, timing, permissions, or platform details. Attempting to execute every possible combination would require impractical amounts of time, effort, and computing resources.

The Foundation Level syllabus therefore teaches testers to select tests based on risk, requirements, likely defect locations, equivalence partitions, boundary values, and other systematic test techniques. The aim is not to prove that every conceivable scenario has been exercised, but to provide useful evidence about quality and reduce the risks that matter to stakeholders. This principle supports rational test planning: testers prioritize the combinations most likely to reveal important defects and adapt coverage to the product's risks and context. The ISTQB glossary defines exhaustive testing as testing all combinations of inputs and preconditions, reinforcing why it is generally infeasible in practice. See the ISTQB Glossary entry for exhaustive testing and the ISTQB Certified Tester Foundation Level certification page.

QUESTION NO: 26

Which of the following BEST describes error guessing?

A. Error guessing involves designing tests based on experience, defect data, or common knowledge about why software fails

B. Error guessing involves building test cases based on the various experiences of developers, architects, users, and other stakeholders on project teams

C. Error guessing is a suitable test technique and can be used effectively in place of more formal techniques

D. Error guessing is a valid and useful white-box test technique due to its high degree of statement coverage

ANSWER: A

Explanation:

Error guessing involves designing tests based on experience, defect data, or common knowledge about why software fails. It is an experience-based test technique in the ISTQB Foundation Level syllabus. The tester uses knowledge gained from previous projects, familiarity with typical implementation mistakes, known defect patterns, and awareness of failure-prone areas to anticipate where defects may be present and to derive focused test conditions and test cases. For example, a tester may deliberately check boundary values, empty fields, unusual input formats, repeated actions, or error-handling paths because such situations have frequently revealed defects in similar systems.

This technique is valuable because it applies practical insight that may not be fully represented by specification-based or structure-based techniques. Error guessing is commonly used alongside more systematic techniques, helping testing address risks indicated by historical defects and tester expertise. The approach is inherently dependent on the tester's relevant knowledge, which is why retaining defect information and learning from prior testing are useful inputs. ISTQB classifies error guessing within experience-based testing; the CTFL 2018 syllabus describes it as designing tests based on experience, defect data, and common knowledge of software failures. See the ISTQB Certified Tester Foundation Level page and the ISTQB downloads repository for the official CTFL syllabus.

QUESTION NO: 27

Which of the following are part of ISTQB code of ethics?

I. Certified software testers shall advance the integrity and reputation of the profession consistent with the public interest

II. Certified software tester shall always sign a NDA (Non Disclosure Agreement) in presence of customer data

III. Certified software testers shall maintain integrity and independence in their professional judgment

IV Certified software testers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest

A. I, II, III

B. II, III, IV

C. I, II, V

D. I, III, IV

E. Certified software testers shall advance the integrity and reputation of the profession consistent with the public interest

II. Certified software tester shall always sign a NDA (Non Disclosure Agreement) in presence of customer data

III. Certified software testers shall maintain integrity and independence in their professional judgment

IV Certified software testers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest

ANSWER: D

Explanation:

The correct selection is **I, III, IV**. These statements directly reflect core principles of the ISTQB/ACM-IEEE Software Testing Code of Ethics. Test professionals are expected to act consistently with the public interest, including when advancing the integrity and reputation of the testing profession. This professional responsibility supports confidence in software testing and emphasizes that testers' conduct affects users, organizations, and society.

The code also requires testers to maintain integrity and independence in professional judgment. Independence here concerns making honest, evidence-based assessments rather than allowing inappropriate pressure, bias, or conflicting interests to determine testing conclusions. In addition, testers must act in the best interests of their client and employer while remaining consistent with the public interest. This balances legitimate organizational responsibilities with the wider duty to avoid practices that could harm users or undermine trust.

These ethical expectations are part of the professional-foundation material associated with the Certified Tester Foundation Level syllabus and ISTQB's certification framework. See the ISTQB Certified Tester Foundation Level page and the ISTQB official website for certification and professional guidance.

QUESTION NO: 28

Which of the following would NOT be a typical target of testing support tools?

A. Automate activities that require significant resources when done manually

B. Automate activities that cannot be executed manually

C. Automate repetitive tasks

D. Automating repetitive inspections

ANSWER: B

Explanation:

“Automate activities that cannot be executed manually” is the correct answer because the normal purpose of testing support tools is to make testing activities that people can perform more efficient, repeatable, scalable, and less error-prone. Tool support is selected where it provides useful assistance to a testing process, such as reducing the manual effort needed to perform a task, increasing consistency, or enabling the same activity to be performed at a useful frequency or volume. The activity remains a testing activity with an understandable objective and outcome; the tool provides support for its execution, management, analysis, or reporting.

The CTFL syllabus describes the principal rationale for tool support in terms of improving testing efficiency through automation, particularly for repetitive work and work that would require substantial resources if done manually. Therefore, an activity being inherently impossible to execute manually is not the typical selection criterion expressed by the syllabus. Appropriate tool use also requires considering the tool’s fit with the organization, processes, skills, and intended testing objectives. See the official ISTQB Certified Tester Foundation Level certification page and the ISTQB glossary definition of test tool.

QUESTION NO: 29

Given the following types of testing and testing scenarios:

a. Regression testing

b. Confirmation testing

1. Testing to ensure the application of a new version of the operating system does not have any unintended side-effects on the system

2. Testing due to the application of a security patch

3. Testing due to the application of a new version of the database management system

4. Testing to ensure the fix to the payroll system truly works

Which of the following BEST matches the type of testing with the testing scenario?

A. 1-a, 2-a, 3-b, 4-b

B. 4-a, 3-a, 1-b, 2-b

C. 1-a, 2-a, 3-a, 4-b

D. 4-a, 3-a, 2-b, 1-b

ANSWER: C

Explanation:

The correct matching is **1-a, 2-a, 3-a, 4-b**. Regression testing checks that existing functionality continues to work as intended after a change. The new operating-system version, security patch, and new database-management-system version are

changes to the system's operating environment. Testing following each of these changes is therefore regression testing, because the objective is to identify unintended effects on already working software and its interfaces. This is important even where the application code itself has not changed, since environment changes can alter behavior, compatibility, performance, security-related interactions, or integrations.

Confirmation testing is performed after a defect has been corrected, with the specific objective of establishing that the reported failure has been successfully fixed. Testing to ensure that the payroll-system fix truly works is confirmation testing because it directly retests the corrected behavior against the defect's expected result. In practice, confirmation testing and regression testing are often both performed after a correction: confirmation testing verifies the particular fix, while regression testing checks for unintended consequences in related or unaffected areas.

This distinction is defined in the ISTQB Foundation Level syllabus section on maintenance testing and is also reflected in the ISTQB glossary definitions of regression and confirmation testing. See the ISTQB Certified Tester Foundation Level resources and the ISTQB Glossary: regression testing.

QUESTION NO: 30

A garden irrigation system allows the user to specify 2 inputs:

1. Frequency - The number of times the system should be automatically switched on per day; minimum once per day, maximum 5 times

2. Duration - The duration of operation, in whole minutes, each time it is switched on; ranging from 1 to 60

Applying 2-value boundary value analysis which of the following options has the correct test set of valid and invalid boundary values?

A. Frequency 1, 5; Duration 1, 60

B. Frequency 0, 1, 5, 6; Duration 59 seconds, 1 minute, 60 minutes, 60 minutes 1 second

C. Frequency 0, 1, 5, 6; Duration 0, 1, 30, 60, 61

D. Frequency 0, 1, 2, 5, 6; Duration 0, 1, 30, 60, 61

ANSWER: C

Explanation:

Frequency 0, 1, 5, 6; Duration 0, 1, 30, 60, 61 is correct because 2-value boundary value analysis selects two values for each boundary: the boundary value itself and one value immediately beyond it, on the adjacent side of the boundary. The Frequency input has an allowed range from 1 through 5. Its lower boundary is therefore tested with 0 and 1, while its upper boundary is tested with 5 and 6. This covers both valid boundary inputs and the immediately adjacent invalid inputs.

Duration is constrained to whole-minute values from 1 through 60. Accordingly, 0 and 1 exercise the lower boundary, and 60 and 61 exercise the upper boundary. The additional value 30 is a representative valid interior value, useful for confirming normal processing away from the boundaries. Importantly, the duration values must be expressed in whole minutes because that is the defined input domain; values in seconds do not conform to this specification. ISTQB describes boundary value analysis as deriving tests from boundary values of equivalence partitions, with two-value BVA using each boundary and its closest neighboring value. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB CTFL syllabus resources.

QUESTION NO: 31

Which of the following is an example of Static testing?

A. Calculating path coverage using tools

B. Performance testing

C. Requirements review

D. Usability testing

ANSWER: C

Explanation:

Requirements review is an example of static testing because it evaluates a work product without executing software code. Static testing includes activities such as reviews, walkthroughs, inspections, and static analysis. In a requirements review, relevant stakeholders examine the requirements specification to identify defects early, including ambiguity, inconsistency, incompleteness, lack of testability, or conflicts with business needs. The review may use defined roles, entry criteria, checklists, and documented findings, but it does not require the system or component described by the requirements to run.

Finding defects at this stage supports the fundamental testing principle of early testing: defects in requirements can otherwise propagate into design, implementation, and test artifacts, where they are typically more costly to correct. Static testing therefore contributes to product quality and improves the quality of later dynamic testing by making the test basis clearer and more testable. The ISTQB Foundation Level syllabus distinguishes static testing from dynamic testing precisely by whether the software is executed; reviews assess documents and other work products directly. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB glossary entry for static testing.

QUESTION NO: 32

Which of the following factors is LEAST likely to be used as a basis for estimating testing effort?

A. Requirements for documentation

B. Quality of the test basis Work

C. Breakdown Structure

D. Skills of the test team

ANSWER: C

Explanation:

Breakdown Structure is the least likely factor to be used as a direct basis for estimating testing effort. The ISTQB Foundation Level syllabus identifies factors that influence test-effort estimates as characteristics of the product, the development process, the people involved, and previous or anticipated test results. In practice, these include such matters as the quality of the test basis, the competence and experience of the test team, the scope and complexity of testing, product risk, and the required test documentation. These characteristics provide substantive evidence about how much testing work is likely to be needed.

A work breakdown structure is instead a project-planning and decomposition technique. It organizes project or testing work into manageable activities and work packages, which can help a team structure, allocate, and monitor an estimate after the relevant effort drivers have been considered. It is therefore useful in planning and presenting an estimate, but it is not itself an inherent factor that determines the amount of testing effort required. This distinction makes **Breakdown Structure** the best answer to a question asking for the factor least likely to form the basis of the estimate. The relevant ISTQB estimation guidance is covered in the Foundation Level syllabus and certification materials: ISTQB Certified Tester Foundation Level and ISTQB Certifications.

QUESTION NO: 33

A system sets new users' password to a temporary password

The temporary password is a random number based on the first sft: characters of the username. If the username is shorter than 6 characters, the system displays an error message.

Which of the following is a possible representation of equivalence classes for the username string?

A. {Random number}

{Error message}

B. {Error message displayed}

{Error message not displayed}

C. {Username shorter than 6 characters}

{Username equal to or longer than 6 characters}

D. {Username with a permanent password}

{Username with a temporary password}

ANSWER: C

Explanation:

{Username shorter than 6 characters} {Username equal to or longer than 6 characters} is the appropriate representation because equivalence partitioning divides an input domain into groups that are expected to receive the same processing. Here, the relevant input characteristic of the username is its length, and the stated business rule establishes a boundary at six characters. Every username whose length is below six belongs to one partition because it triggers the same specified outcome: an error message. Every username with a length of six or more belongs to the other partition because it satisfies the length precondition for the temporary-password processing.

These classes are mutually exclusive and collectively cover the relevant username-length domain: a given username cannot simultaneously be shorter than six and at least six characters long, while every possible username length falls into one of those groups. This is exactly the purpose of equivalence partitioning: select representative tests from partitions where the system is expected to behave equivalently, rather than testing every individual input value. The six-character threshold can subsequently be targeted with boundary value analysis using values immediately below and at the boundary. ISTQB describes equivalence partitioning and boundary value analysis as black-box test techniques in its Foundation Level materials: ISTQB Certified Tester Foundation Level.

QUESTION NO: 34

Equivalence Partitioning is best defined as:

A. An analysis technique that divides inputs into groups that are expected to exhibit similar behaviors.

B. Applying to time-related data classes only.

C. A form of white-box testing.

D. A method to reduce test coverage.

ANSWER: A

Explanation:

Equivalence partitioning is a black-box test technique in which the test basis is divided into partitions, or equivalence classes, that are expected to be processed in the same way by the system. The core idea is that values within the same partition should produce equivalent behavior, so testing one representative value from a valid partition and one from each relevant invalid partition can provide effective coverage without testing every possible input. For example, if a field accepts whole-number quantities from 1 through 100, values within that valid range form a partition, while values below 1 and above 100 represent invalid partitions. The technique is based on identifying these groups from requirements, business rules, input domains, output values, or other applicable conditions. It supports efficient test design while retaining meaningful coverage of the specified behavior. ISTQB describes equivalence partitioning as dividing data into groups that are expected to be processed in the same way, and it is commonly used together with boundary value analysis to select focused test cases. See the ISTQB Certified Tester Foundation Level syllabus and the ISTQB glossary definition of equivalence partitioning.

QUESTION NO: 35

An iPhone application identifies and counts all purchases of a particular product from a shopping website. The application incorrectly counts purchase attempts by including both failed attempts, and also those where the purchase was terminated by the user before completion. Testing has identified that the problem was located in the 'purchase identification' module, where the first stage in the purchasing process was counted, rather than a successful confirmed purchase.

Which of the following statements correctly identifies what has happened? [K2]

A. The application failed because of a defect in the purchase identification module caused by a programmer mistake or an error in the specification.

B. An error by the programmer led to a mistake in the purchase identification module and this caused a defect in the application

C. A defect in the purchase identification module caused by a mistake in the module specification led to a defect in the overall application

D. A bug in the purchase identification module caused a fault in the application

ANSWER: A

Explanation:

The application failed because of a defect in the purchase identification module caused by a programmer mistake or an error in the specification. The observed incorrect purchase total is a failure: the application does not perform the required function, because it counts attempts that do not result in a confirmed purchase. Testing has localized the underlying defect to the purchase-identification logic, which uses the first stage of the purchasing process as the condition for counting a purchase instead of using successful confirmation.

ISTQB terminology distinguishes the externally observable failure from the internal defect that produces it. A defect may be introduced through a human error, such as a programmer implementing an incorrect condition or a specification author defining the wrong business rule. The scenario does not provide enough evidence to decide which of those human-error sources occurred, so the statement correctly expresses both supported possibilities without claiming one specific origin. This causal chain—human error potentially introducing a defect, and the defect causing a failure when executed—is aligned with the CTFL glossary definitions. See the ISTQB definition of failure and the ISTQB definition of defect.

QUESTION NO: 36

Which of the following statements is true?

A. Regression testing is a technique for system testing and acceptance testing only.

B. Regression testing can be used in any level of testing, but must be done after white-box testing has been finished.

C. Regression testing is an alternative term for re-testing

D. Regression testing is intended to find side-effects after changes have been applied to the system under test.

ANSWER: D

Explanation:

Regression testing is intended to find side-effects after changes have been applied to the system under test. A change can include a defect fix, an enhancement, a configuration change, an environment update, or another modification that might unintentionally affect functionality that previously worked. Regression testing therefore involves re-executing relevant tests to provide confidence that unchanged parts of the product still behave as expected and that the change has not introduced new defects elsewhere.

This purpose is central to regression testing in the ISTQB Foundation Level syllabus. The scope of regression testing is selected according to the nature and risk of the change, the affected components and interfaces, and the potential impact on existing functionality. It is commonly supported by test automation because the same checks may need to be run repeatedly

after builds or changes. Regression testing is especially valuable in iterative, incremental, and maintenance work, where frequent changes can create unintended consequences in already-tested areas. The test basis and prior test results help testers identify the tests needed to detect those possible side-effects. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB glossary definition of regression testing.

QUESTION NO: 37

Which of the following would NOT be a common metric used for monitoring test preparation and execution?

A. Number of Test cases passed and failed

B. Percentage of planned test cases designed

C. Number of test plan review comments

D. Percentage of tasks complete in test environment preparation

ANSWER: C

Explanation:

Number of test plan review comments is the correct answer. In the CTFL 2018 syllabus, test monitoring involves collecting information to assess progress and to support test control actions during test preparation and execution. Commonly monitored measures include progress in designing and implementing test cases, test execution results such as passed and failed tests, completion of test-environment preparation tasks, coverage, defects, effort, and schedule-related task completion. These measures provide direct visibility into whether the planned testing work is ready, underway, and producing the expected evidence about product quality.

A count of comments raised while reviewing a test plan can be useful information for improving a particular test work product or tracking the progress of that review. However, it is not identified as a common operational metric for monitoring the preparation and execution of testing itself. It concerns review feedback on a planning document rather than the status of test-case design, environment readiness, execution, or results. This distinction is important: test monitoring metrics should help the test manager make timely control decisions about the ongoing test activities and their completion against the test plan.

See the ISTQB Certified Tester Foundation Level certification page and the ISTQB CTFL syllabus for test monitoring and control guidance.

QUESTION NO: 38

Which from the following list are typically found to enable the review process to be successful? [K2]

a. Each review has clear defined objectives

b. The lower the number of defects, the better the review process

c. The right people for the review objective are involved

d. There is an emphasis on learning and process improvement

e. Management are not involved in the process at all

f. Checklists should not be used, as these slow down the process

g. Defects found are welcomed and expressed objectively

A. a, f and g.

B. b, c and f.

C. a, c and d.

D. d, e and g.

E. a, c, d and g.

ANSWER: E

Explanation:

The correct combination is “a, c, d and g.” The ISTQB Foundation Level syllabus identifies several factors that support successful reviews. Reviews need clear, predefined objectives so that participants understand the purpose, scope, and expected outcome of the review. The people involved must be appropriate for the review objective, ensuring that the necessary technical, business, and quality perspectives are available.

A successful review culture also treats defects as valuable information. Defects should be welcomed and expressed objectively, with attention directed toward improving the work product rather than criticizing an individual. This supports constructive collaboration, trust, and effective communication among review participants. In addition, reviews should emphasize learning and process improvement. Review findings can reveal recurring defect patterns, weaknesses in development practices, or opportunities to improve review techniques and checklists. Applying these lessons improves both future work products and the review process itself.

These success factors are stated in the ISTQB Certified Tester Foundation Level Syllabus, section 3.2.1, “The Review Process.” See the ISTQB CTFL certification page and the ISTQB downloads page for the official syllabus.

QUESTION NO: 39

Which of the following statements about use-case testing are most accurate?

(i)In a use-case diagram an actor represents a type of user. (ii)Use-cases are the most common test basis for unit testing.

(iii) A use-case describes interactions between actors.

(iv) An actor is always a human user that interacts with the system. (v)Test cases can be based on use-case scenarios.

(vi) Use-case testing will often identify gaps not found by testing individual components.

A. ii, iii, iv, v

B. i, iii, v, vi

C. i, ii, iv, v

D. iii, iv, v, vi

ANSWER: B

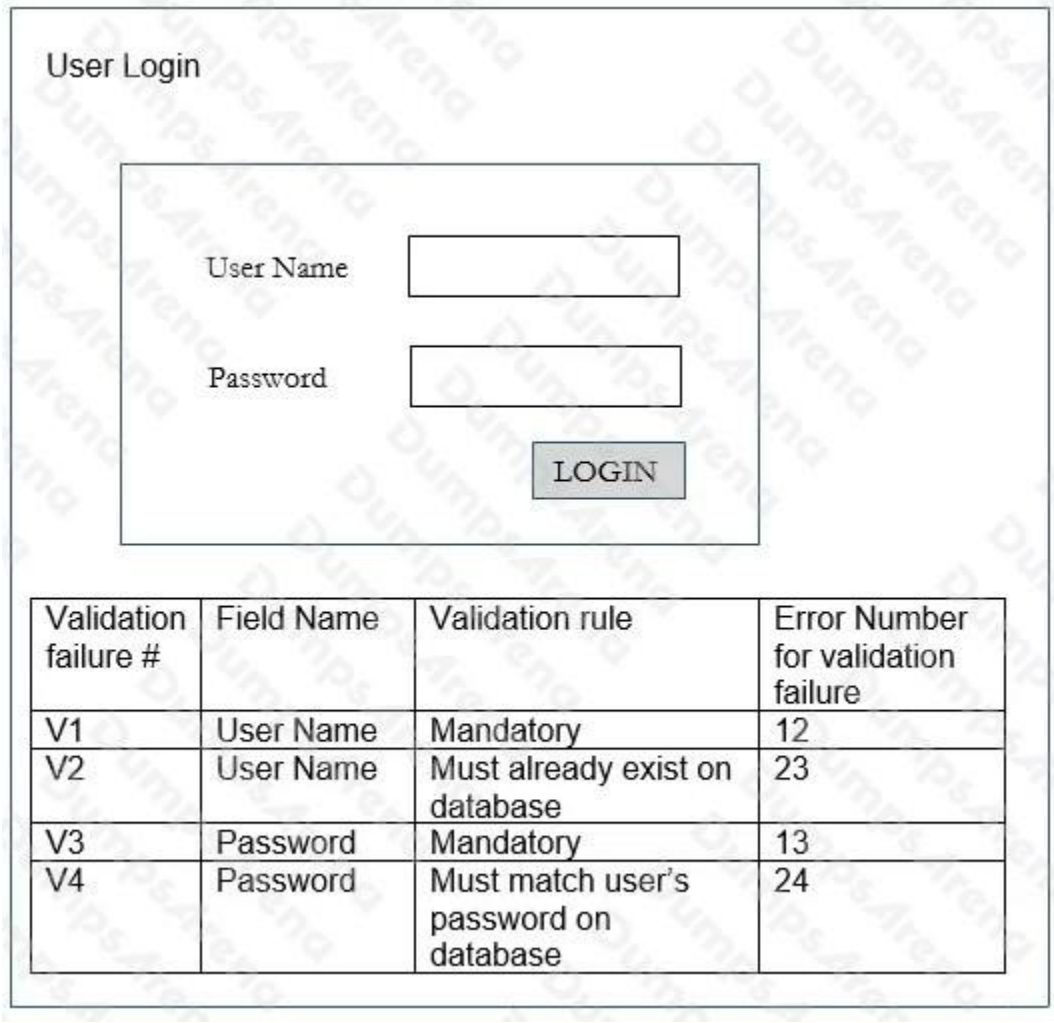
Explanation:

The correct selection is **i, iii, v, vi**. Use-case testing is a black-box test technique in which test conditions and test cases are derived from use cases and their scenarios. A use case captures how an actor participates in achieving a goal through interaction with the system; consequently, actors represent roles external to the system, such as a category of user or another interacting entity. The interactions represented by a use case provide a valuable basis for end-to-end test scenarios, including normal flows, alternative flows, and exception flows.

Because scenarios describe behaviour across interfaces and collaborating parts of the system, tests based on them can reveal omissions, incorrect handoffs, and integration gaps that may not be visible when components are tested independently. This makes use-case testing particularly useful for validating business processes and user-oriented system behaviour. The ISTQB Foundation Level syllabus identifies use cases as a source for deriving tests and describes the value of testing interactions across the system. See the ISTQB Certified Tester Foundation Level certification page and the ISTQB Glossary entry for “use case”.

QUESTION NO: 40

You are performing a review of your colleague’s test cases based on the following test basis document:



Validation failure #	Field Name	Validation rule	Error Number for validation failure
V1	User Name	Mandatory	12
V2	User Name	Must already exist on database	23
V3	Password	Mandatory	13
V4	Password	Must match user's password on database	24

The Test Cases are as follows:

TC1. Success – valid 'User Name' and 'Password'; Customer Menu displayed TC2. Failure – 'User Name' field has blank entry; Error Number 12 displayed

TC3. Failure – 'User Id' entered does not exist on database (i.e. unregistered user); Error Number 23 displayed TC4. Failure – 'Password' entered does not match user's password on database; Error Number 24 displayed You are guided by the following checklist in your review:

C1. There must be one test case to cover success

C2. There must be one test case for each error path (e.g. validation failure)

C3. Each test case must use terminology consistent with the test basis document (field names, error numbering, etc.)

Record a separate defect for each missing test case (checklist items C1 and C2) and for each test case that does not meet checklist item C3.

How many defects should you record?

A. 1.

B. 2.

C. 3.

D. 4.

ANSWER: C

Explanation:

The correct total is 3 defects. The successful-login behaviour is covered, so the review can focus on error-path coverage and consistency with the test basis. The documented behaviour includes a validation error path for a blank Password, but no test case exercises that condition. This is one missing-test-case defect under the checklist's error-path coverage requirement. In addition, the test basis uses the field name "User Id". The success test and the blank-field test instead use "User Name". Each of those test cases is independently non-compliant with the terminology-consistency checklist item, so each requires its own recorded defect. The remaining stated error paths use the specified error numbers and the corresponding terminology, and therefore support traceability from the test basis to the tests. Counting the one omitted error-path test plus the two separate terminology inconsistencies gives three recorded defects. This application of a checklist is consistent with ISTQB's definition of a review as an evaluation of a work product to identify defects and recommend improvements; see the ISTQB Glossary entry for "review". The Foundation Level syllabus also identifies reviews as a static testing activity used to find defects in work products before execution; see ISTQB Certified Tester Foundation Level.

QUESTION NO: 41

Which of the following would be a key difference between a peer review of code and static analysis of code using a tool?

A. A peer reviews finds defects while static analysis finds failures

B. Peer reviews cannot find missing requirements whereas static analysis can

C. A peer reviews find failures while static analysis finds defects

D. Static analysis targets the code technically, whereas peer review is applicable to further aspects.

ANSWER: D

Explanation:

Static analysis targets the code technically, whereas peer review is applicable to further aspects. Static analysis uses tools to examine code without executing it. It is particularly suited to applying predefined technical rules consistently, such as identifying suspicious control flow, data-flow anomalies, metrics, dependencies, coding-standard violations, and certain security-related patterns. The analysis is automated and based on the rules, models, and configuration supplied to the tool.

A peer review is a human examination activity. Even when the review subject is source code, reviewers can apply context, domain knowledge, architectural understanding, maintainability concerns, and their interpretation of intended behavior. This means a review can address broader qualities and concerns than those detectable through a technical rule set alone. The key distinction is therefore not whether one technique finds defects and the other finds failures: both activities are static test techniques concerned with identifying defects rather than observing failures during execution. The distinction is the automated, technically focused examination performed by static-analysis tools compared with the broader human assessment possible in peer review. ISTQB describes reviews and static analysis as complementary static testing techniques; see the ISTQB Certified Tester Foundation Level resources and the ISTQB glossary entry for static analysis.