ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

**"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."**

**www.atmicnetworks .com**

Warning: Keep connected with our support team for latest updates

## Question: 1

Which option is part of the 'implementation and execution' area of the fundamental test process?

A. Developing the tests.
B. Comparing actual and expected results.
C. Writing a test summary.
D. Analyzing lessons learnt for future releases.

**Answer: B**

Explanation:

The implementation and execution phase of the fundamental test process involves the execution of the tests according to the test plan, where the main task is to compare the actual results of the test execution against the expected results. This phase ensures that the tests are carried out and the discrepancies between the actual and expected outcomes are identified and documented. This information is critical to determining whether the software behaves as intended and meets the specified requirements.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.4.1, Test Activities and Tasks

## Question: 2

The five parts of the fundamental test process have a broad chronological order. Which of the options gives three different parts in the correct order?

A. Implementation and execution, planning and control, analysis and design.
B. Analysis and design, evaluating exit criteria and reporting, test closure activities.
C. Evaluating exit criteria and reporting, implementation and execution, analysis and design.
D. Evaluating exit criteria and reporting, test closure activities, analysis and design.

**Answer: B**

Explanation:
The fundamental test process consists of the following broad chronological steps:
Planning and control
Analysis and design
Implementation and execution
Evaluating exit criteria and reporting
Test closure activities
Option B correctly places three of these steps in the correct order: Analysis and design, followed by Evaluating exit criteria and reporting, and finally Test closure activities. This order reflects the logical sequence of performing detailed test analysis and design before executing tests and then evaluating the results to determine if exit criteria have been met, followed by wrapping up the testing activities. Reference: ISTQB CTFL Syllabus V4.0 - Section 1.4, The Test Process

## Question: 3

Which statement is most true?

A. Different testing is needed depending upon the application.
B. All software is tested in the same way.
C. A technique that finds defects will always find defects.
D. A technique that has found no defects is not useful.

**Answer: A**

Explanation:
Different types of applications require different testing approaches based on their specific characteristics, risk profiles, and operational environments. For example, testing a financial application would focus heavily on security and accuracy, while testing a mobile game might focus more on performance and usability. The context-driven nature of software testing recognizes that there is no one-size-fits-all approach, and effective testing must be tailored to the particular application under test.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.1, What is Testing?

## Question: 4

A bug or defect is:

A. A mistake made by a person;
B. A run-time problem experienced by a user;
C. The result of an error or mistake;
D. The result of a failure, which may lead to an error?

**Answer: C**

Explanation:

A defect, also known as a bug, is a flaw in a component or system that can cause it to fail to perform its required function. Defects arise from errors made by people, such as mistakes in code or design. When a defect in the software is executed, it can cause the system to behave unexpectedly, leading to failures. The relationship between errors, defects, and failures is crucial to understanding the importance of early detection and correction in the software development lifecycle.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.2.3, Human Error, Defects, Failures, and Root Causes

## Question: 5

The effect of testing is to:

A. Increase software quality;
B. Give an indication of the software quality;
C. Enable those responsible for software failures to be identified;

D. Show there are no problems remaining?

**Answer: B**

Explanation:
Testing provides valuable information about the quality of the software by identifying defects and issues before the software is released. It helps stakeholders make informed decisions about the release and deployment of the software. While testing can identify defects and provide confidence in the quality of the software, it cannot guarantee the absence of defects or ensure that the software quality will inherently increase. The primary goal of testing is to reveal problems and provide an indication of the software's current state of quality.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.2.1, The Contribution of Testing to Success

## Question: 6

What is retesting?

A. Running the same test again in the same circumstances to reproduce the problem.
B. A cursory run through a test pack to see if any new errors have been introduced.
C. Checking that the predetermined exit criteria for the test phase have been met.
D. Running a previously failed test against new software/data/documents to see if the problem is solved.

**Answer: D**

Explanation:
Retesting, also known as confirmation testing, involves executing a previously failed test case after the defect has been fixed. The objective is to ensure that the original defect has been successfully removed. This type of testing is critical in verifying that the defect fixes are effective and that the

software now operates as expected under the same conditions that previously caused the failure. Reference:
ISTQB CTFL Syllabus V4.0 - Section 1.4.2, Retesting

## Question: 7

Which of the following is correct?

Debugging is:

A. Testing/checking whether the software performs correctly.
B. Checking that a previously reported defect has been corrected.
C. Identifying the cause of a defect, repairing the code and checking the fix is correct.
D. Checking that no unintended consequences have occurred as a result of a fix.

**Answer: C**

Explanation:

Debugging is the process that developers follow to find, analyze, and fix the root causes of defects in the software. This includes identifying the source of the problem, correcting the code, and then testing the changes to ensure the defect has been properly fixed and that the fix does not introduce new issues.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.1.2, Testing and Debugging

## Question: 8

When is testing complete?

A. When time and budget are exhausted.
B. When there is enough information for sponsors to make an informed decision about release.
C. When there are no remaining high priority defects outstanding.
D. When every data combination has been exercised successfully.

**Answer: B**

Explanation:

Sometimes time/money does signify the end of testing, but it is really complete when everything that was set out in advance has been achieved.
Testing is considered complete when sufficient information has been gathered to assess the quality of the software and to make an informed decision about its release. This includes understanding the risks of any remaining defects, the test coverage achieved, and the test results. It is not practical to wait until all possible test cases have been executed or until no defects remain, as these conditions are often impossible to meet within typical project constraints.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.1, What is Testing?

## Question: 9

Which list of levels of tester independence is in the correct order, starting with the most independent first?

A. Tests designed by the author; tests designed by another member of the development team; tests designed by someone from a different company.
B. Tests designed by someone from a different department within the company; tests designed by the author; tests designed by someone from a different company.
C. Tests designed by someone from a different company; tests designed by someone from a different department within the company; tests designed by another member of the development team.
D. Tests designed by someone from a different department within the company; tests designed by someone from a different company; tests designed by the author.

**Answer: C**

Explanation:

This option has someone nearer to the written code in each statement. All other options are not in this order.
Tester independence refers to the degree to which the tester is separate from the development team to avoid bias and increase the objectivity of the tests. The highest level of independence is achieved when tests are designed by

someone from an entirely different company, followed by someone from a different department within the same company, and then by another member of the development team. The lowest level of independence is when tests are designed by the author of the code being tested.

Reference: ISTQB CTFL Syllabus V4.0 - Section 1.4.3, Degrees of Independence

## Question: 10

Which of the following is in the correct order (typically)?

A. Unit testing, system testing, acceptance testing, maintenance testing. B. System testing, unit testing, acceptance testing, maintenance testing. C. Acceptance testing, system testing, maintenance testing, unit testing. D. Unit testing, maintenance testing, system testing, acceptance testing.

**Answer: A**

Explanation:

The typical order of testing phases starts with unit testing, where individual components are tested in isolation. This is followed by system testing, where the integrated system is tested as a whole. Acceptance testing is then conducted to ensure the system meets the business requirements and is ready for deployment. Finally, maintenance testing is performed after the system is in production to ensure it continues to function correctly and to address any issues that arise.

Reference: ISTQB CTFL Syllabus V4.0 - Section 2.2.1, Test Levels

## Question: 11

Which TWO of the review types below are the BEST fitted (most adequate) options to choose for reviewing safety critical components in a software project? Select 2 options.

A. Informal review.

B. Management review.

C. Inspection.

D. Walkthrough

E. Technical Review

**Answer: C, E**

Explanation:

For reviewing safety-critical components in a software project, the most adequate review types are Inspections and Technical Reviews. Inspections are a formal review technique with a well-defined process that includes thorough preparation, examination, and documentation, making them suitable for critical components. Technical Reviews involve a detailed examination of the technical aspects of the software and are conducted by experts, which ensures that any potential issues are identified and addressed early.

Reference: ISTQB CTFL Syllabus V4.0 - Section 3.2, Review Types

## Question: 12

Incidents would not be raised against

A. Requirements
B. Documentation
C. Test cases
D. Improvements suggested by users

**Answer: D**

Explanation:

Incidents, or defects, are typically raised against artifacts such as requirements, documentation, and test cases when they do not meet their expected outcomes or contain errors. However, improvements suggested by users are not considered incidents, as they are enhancements or changes that users would like to see, rather than defects in the current functionality or documentation.
Reference: ISTQB CTFL Syllabus V4.0 - Section 5.5, Incident Management

## Question: 13

The cost of fixing a fault:

A. Is not important
B. Increases as we move the product towards live use
C. Decreases as we move the product towards live use
D. Is more expensive if found in requirements than functional design
E. Can never be determined

**Answer: B**

Explanation:
The cost of fixing a fault increases as the software progresses through the development lifecycle. Fixing defects identified early in the requirements or design phases is less expensive compared to fixing those found later in testing or after the software has been deployed. This is due to the larger impact on the system, more extensive rework required, and potential disruptions to the user experience in later stages.
Reference: ISTQB CTFL Syllabus V4.0 - Section 1.2.1, The Cost of Defects

## Question: 14

Which of the following statements are TRUE?
A. Regression testing and acceptance testing are the same. B. Regression tests show if all defects have been resolved. C. Regression tests are typically well-suited for test automation. D. Regression tests are performed to find out if code changes have introduced or uncovered defects. E. Regression tests should be performed in integration testing.

A. A, C and D and E are true; B is false.
B. A, C and E are true; B and D are false.

C. C and D are true; A, B and E are false.
D. B and E are true; A, C and D are false.

**Answer: C D**

Explanation:

Regression testing is designed to verify that recent code changes have not adversely affected existing functionality. It is particularly well-suited for test automation because automated tests can be rerun quickly and consistently. While regression tests are essential in many testing phases, they are distinct from acceptance testing, which focuses on validating that the system meets business requirements.

Reference: ISTQB CTFL Syllabus V4.0 - Section 4.2.3, Regression Testing

## Question: 15

Which is not the fundamental test process?

A. Planning and control
B. Test closure activities
C. Analysis and design
D. None

**Answer: D**

Explanation:

All the options listed (Planning and control, Test closure activities, Analysis and design) are indeed part of the fundamental test process. The fundamental test process includes activities such as planning and control, analysis and design, implementation and execution, evaluating exit criteria and reporting, and test closure activities. Thus, the correct answer is 'None,' as all options are fundamental parts of the test process.

Reference: ISTQB CTFL Syllabus V4.0 - Section 1.4, The Test Process

## Question: 16

What is the purpose of test completion criteria in a test plan:

A. To know when a specific test has finished its execution
B. To ensure that the test case specification is complete
C. To set the criteria used in generating test inputs
D. To know when test planning is complete
E. To plan when to stop testing

**Answer: E**

Explanation:

The purpose of test completion criteria in a test plan is to define when to stop testing. These criteria are used to determine if sufficient testing has been conducted to meet the test objectives and to decide whether the product is ready for release. They typically include coverage goals, risk levels, defect rates, and other quality indicators. According to the ISTQB CTFL Syllabus, defining test completion criteria helps ensure that testing is both thorough

and efficient, avoiding unnecessary testing while ensuring the product meets quality standards.

## Question: 17

Which of the following statements describes a key principle of software testing?

A. Automated tests allow better statements of confidence about the quality of software products.
B. For a software system, it is normally impossible to test all the input and output combinations.
C. Exhaustive software testing is, with enough effort and tool support, feasible for all software.
D. The purpose of software testing is demonstrating the absence of defects in software products.

**Answer: B**

Explanation:

A key principle of software testing is that it is usually impossible to test all possible input and output combinations for a software system. This principle, highlighted in the ISTQB CTFL Syllabus,

acknowledges the practical limitations of testing. Instead, risk-based testing and prioritization techniques are employed to focus on the most critical areas. This approach ensures effective and efficient testing without attempting the infeasible task of exhaustive testing.

## Question: 18

Reviewing the test Basis is a part of which phase

A. Test Analysis and Design
B. Test Implementation and execution
C. Test Closure Activities
D. Evaluating exit criteria and reporting

**Answer: A**

Explanation:

Reviewing the test basis is a key activity in the Test Analysis and Design phase. This phase involves analyzing the test basis documents (such as requirements, design specifications, and use cases) to identify test conditions and design test cases. The ISTQB CTFL Syllabus explains that thorough review of the test basis ensures that the tests are aligned with the specified requirements and design, improving the accuracy and relevance of the tests.

## Question: 19

Which of the following is a benefit of test independence?

A. It does not require familiarity with the code.
B. It is cheaper than using developers to test their own code.

C. It avoids author bias in defining effective tests.
D. Testers are better at finding defects than developers.

**Answer: C**

Explanation:

Test independence is beneficial because it helps to avoid author bias, ensuring that the tests are designed and executed objectively. When testers who are not involved in the development of the code test it, they bring a fresh perspective and are more likely to identify defects that the developers might have overlooked due to familiarity with their own work. This principle is supported by the ISTQB CTFL Syllabus, which highlights that independence can enhance the effectiveness of testing by eliminating the preconceived notions and biases of those who wrote the code.

## Question: 20

Failure is

A. Incorrect program behavior due to a fault in the program
B. Bug found before product Release
C. Bug found after product Release
D. Bug found during Design phase

**Answer: A**

Explanation:

A failure is defined as the incorrect program behavior caused by a fault (defect) in the program. This means that when a defect is executed, it leads to the system not performing as expected, which is observed as a failure. This concept is clearly defined in the ISTQB CTFL Syllabus, where it distinguishes between defects (faults) and the resulting failures, emphasizing that failures are the manifestation of defects during the execution of the software.

## Question: 21

During which test activity could faults be found most cost effectively?

A. Execution
B. Design
C. Planning
D. Check Exit criteria completion

**Answer: B**

Explanation:

Identifying faults during the design phase is most cost-effective because defects found early in the software development lifecycle are cheaper to fix. During the design phase, faults can be detected through reviews and

inspections before any code is written, thus avoiding the costs associated with later stages of development. According to the ISTQB CTFL Syllabus, early testing and defect detection are key principles, which emphasize that the cost of fixing defects increases the later they are found in the lifecycle.

## Question: 22

Tests are prioritized so that:

A. You shorten the time required for testing
B. You do the best testing in the time available
C. You do more effective testing
D. You find more faults

**Answer: B**

Explanation:

Tests are prioritized to ensure that the most critical and risk-prone areas are tested first, thereby achieving the best possible coverage within the available time frame. This approach helps manage resources effectively and ensures that significant defects are found early. As per the ISTQB CTFL Syllabus, prioritization helps in maximizing the effectiveness of testing by focusing on the most important aspects and making the best use of limited time and resources.

## Question: 23

Which of the following comparisons of component testing and system testing are TRUE?

A. Component testing verifies the functioning of software modules, program objects, and classes that are separately testable, whereas system testing verifies interfaces between components and interactions with different parts of the system.
B. Test cases for component testing are usually derived from component specifications, design specifications, or data models, whereas test cases for system testing are usually derived from requirement specifications, functional specifications or use cases.
C. Component testing focuses on functional characteristics, whereas system testing focuses on functional and non-functional characteristics.
D. Component testing is the responsibility of the technical testers, whereas system testing typically is the responsibility of the users of the system.

**Answer: B**

Explanation:

Component testing focuses on the individual parts of the software, often using design specifications, component specifications, or data models to derive test cases. In contrast, system testing evaluates the entire system's functionality and performance, using requirements and functional specifications or use cases to develop test cases. The ISTQB CTFL Syllabus clarifies these distinctions, emphasizing the different sources for test case

derivation and the scopes of each testing level.

## Question: 24

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

A. Testing pinpoints (identifies the source of) the defects. Debugging analyzes the faults and proposes prevention activities.
B. Dynamic testing shows failures caused by defects. Debugging finds, analyzes, and removes the causes of failures in the software.
C. Testing removes faults. Debugging identifies the causes of failures.
D. Dynamic testing prevents causes of failures. Debugging removes the failures.

**Answer: B**

Explanation:
Testing and debugging are two distinct activities in the software development process. Testing, particularly dynamic testing, involves executing the software to identify failures caused by defects. It aims to find and document defects. Debugging, on the other hand, is the process of diagnosing and

fixing the root causes of the failures identified during testing. According to the ISTQB CTFL Syllabus, testing demonstrates the presence of defects, while debugging addresses and removes these defects.

## Question: 25

Which of the following statements BEST describes one of the seven key principles of software testing?

A. Automated tests are better than manual tests for avoiding the Exhaustive Testing.
B. Exhaustive testing is, with sufficient effort and tool support, feasible for all software.
C. It is normally impossible to test all input / output combinations for a software system.
D. The purpose of testing is to demonstrate the absence of defects. The purpose of testing is to demonstrate the absence of defects.

**Answer: C**

Explanation:

One of the seven key principles of software testing is that exhaustive testing is impossible. This principle highlights the impracticality of testing all possible input and output combinations due to the vast number of permutations. Instead, risk-based and prioritized testing approaches should be used to focus on the most critical areas. This principle is fundamental in the ISTQB CTFL Syllabus, emphasizing the importance of efficient test strategy over attempting exhaustive testing.

## Question: 26

Which of the following, if observed in reviews and tests, would lead to problems (or conflict) within teams?

A. Testers and reviewers are not curious enough to find defects.
B. Testers and reviewers are not qualified enough to find failures and faults.
C. Testers and reviewers communicate defects as criticism against persons and not against the software product.
D. Testers and reviewers expect that defects in the software product have already been found and fixed by the developers.

**Answer: C**

Explanation:
Effective communication is crucial during reviews and testing. When testers and reviewers communicate defects as personal criticism rather than focusing on the software product, it can lead to conflicts and a toxic team environment. The ISTQB CTFL Syllabus stresses the importance of maintaining a professional and objective approach to defect reporting to avoid interpersonal conflicts and promote a collaborative atmosphere.

## Question: 27

The purpose of requirement phase is:

A. To freeze requirements
B. To understand user needs
C. To define the scope of testing
D. All of the above

**Answer: D**

Explanation:
The requirement phase serves multiple purposes. It aims to understand user needs, freeze the requirements to provide a stable basis for development, and define the scope of testing. These activities ensure that the project has a clear direction, and the testing efforts are aligned with the defined requirements. The ISTQB CTFL Syllabus outlines that a thorough requirement phase is essential for the success of subsequent phases in the software development lifecycle.

## Question: 28

Which of the following could be a disadvantage of independent testing?

A. Developer and independent testing will overlap and waste resources.
B. Communication is limited between independent testers and developers.
C. Independent testers are too slow and delay the project schedule.
D. Developers can lose a sense of responsibility for quality.

**Answer: D**

Explanation:
A disadvantage of independent testing is that developers might lose a sense of responsibility for the quality of their work. When developers rely too much on independent testers to find defects, they may become less diligent in their own testing and quality assurance efforts. The ISTQB CTFL Syllabus points out that while independent testing can provide an unbiased perspective and identify defects that might be missed by developers, it can also

lead to potential downsides such as diminished developer accountability.

## Question: 29

Which of the following best describes the purpose of non-functional testing?

A. To measure characteristics of a system which give an indication of how the system performs its functions
B. To ensure that a system complies with the quality standards set by ISO 9126
C. To ensure that the system deals appropriately with software malfunctions
D. To measure the extent to which a system has been tested by functional testing

**Answer: A**

Explanation:
A disadvantage of independent testing is that developers might lose a sense of responsibility for the quality of their work. When developers rely too much on independent testers to find defects, they may become less diligent in their own testing and quality assurance efforts. The ISTQB CTFL Syllabus points out that while independent testing can provide an unbiased perspective and identify defects that might be missed by developers, it can also lead to potential downsides such as diminished developer accountability.

## Question: 30

Which of the following is the task of a Tester?

i. Interaction with the Test Tool Vendor to identify best ways to leverage test tool on the project.
ii. Prepare and acquire Test Data
iii. Implement Tests on all test levels, execute and log the tests.
iv. Create the Test Specifications

A. i, ii, iii is true and iv is false
B. ii, iii, iv is true and i is false
C. i is true and ii, iii, iv are false
D. iii and iv is correct and i and ii are incorrect

**Answer: B**

Explanation:

The tasks of a tester typically include preparing and acquiring test data, implementing tests on all levels, executing and logging the tests, and creating test specifications. Interaction with the test tool vendor is generally not a primary task for testers but rather for test managers or tool specialists. The ISTQB CTFL Syllabus outlines these roles and responsibilities, emphasizing the technical tasks testers perform to ensure thorough testing and documentation.

## Question: 31

Which of the following is not a major task of Exit criteria?

A. Checking test logs against the exit criteria specified in test planning.
B. Logging the outcome of test execution.
C. Assessing if more tests are needed.
D. Writing a test summary report for stakeholders.

**Answer: B**

Explanation:

Logging the outcome of test execution is part of the test execution phase, not directly related to the exit criteria. Major tasks of exit criteria include checking test logs against specified criteria, assessing

if additional tests are needed, and writing test summary reports for stakeholders. The ISTQB CTFL Syllabus specifies these tasks as part of the exit criteria to ensure that testing meets the planned objectives and that sufficient testing has been conducted before concluding the testing phase.

## Question: 32

Which is not a major task of test implementation and execution?

A. Develop and prioritizing test cases, creating test data, writing test procedures and optionally, preparing test harness and writing automated test scripts.
B. Logging the outcome of test execution and recording the identities and versions of the software under test, test tools and test ware.
C. Checking test logs against the exit criteria specified in test planning.
D. Verifying that the test environment has been set up correctly.

**Answer: C**

Explanation:

Checking test logs against the exit criteria specified in test planning is part of the exit criteria evaluation phase, not the test implementation and execution phase. Test implementation and execution tasks include developing and prioritizing test cases, creating test data, writing test procedures, preparing test harnesses, writing automated test scripts, logging the outcome of test execution, and verifying the test environment setup. The ISTQB CTFL Syllabus delineates these tasks to ensure proper test preparation and execution.

## Question: 33

The difference between re-testing and regression testing is:

A. Re-testing is running a test again; regression testing looks for unexpected side effects
B. Re-testing looks for unexpected side effects; regression testing is repeating those tests
C. Re-testing is done after faults are fixed; regression testing is done earlier
D. Re-testing uses different environments, regression testing uses the same environment
E. Re-testing is done by developers, regression testing is done by independent testers

**Answer: A**

Explanation:

Re-testing and regression testing are essential concepts in software testing, each serving a different purpose. According to the ISTQB CTFL Syllabus, version 4.0, the difference between these two types of testing is highlighted as follows:

Re-testing (also known as confirmation testing) involves executing the same tests again on the same version of the software after defects have been fixed. The primary goal is to confirm that the original defects have been successfully removed and that the software behaves as expected.

Regression testing, on the other hand, is performed to ensure that recent code changes have not adversely affected the existing functionalities of the software. This testing aims to detect any unexpected side effects introduced by the new changes.

The syllabus states: "Re-testing is running a test again; regression testing looks for unexpected side effects". This explanation clarifies that while re-testing focuses on verifying specific defect fixes, regression testing ensures the overall stability and integrity of the software after changes.

## Question: 34

Non-functional system testing includes:

A. Testing to see where the system does not function properly
B. Testing quality attributes of the system including performance and usability
C. Testing a system feature using only the software required for that action
D. Testing a system feature using only the software required for that function
E. Testing for functions that should not exist

**Answer: B**

Explanation:

Non-functional system testing evaluates various quality attributes of a system rather than specific behaviors or functionalities. According to the ISTQB CTFL Syllabus, version 4.0, non-functional testing encompasses several critical aspects:

Performance testing evaluates how the system performs under different conditions, such as load and stress, to ensure it meets performance requirements.

Usability testing assesses the user-friendliness and overall user experience of the system, ensuring that it is intuitive and accessible for end-users.

The syllabus outlines that non-functional system testing includes testing the quality attributes of the system, which goes beyond just functionality and delves into performance, usability, reliability, and other key attributes.

## Question: 35

A deviation from the specified or expected behavior that is visible to end-users is called:

A. an error
B. a fault
C. a failure

D. a defect

**Answer: C**

Explanation:

A deviation from the specified or expected behavior that is visible to end-users is defined as a "failure."
According to the ISTQB CTFL Syllabus, version 4.0:
An error is a human action that produces an incorrect result.
A fault (also known as a defect or bug) is a flaw in a component or system that can cause the system to fail to perform its required function.
A failure occurs when a fault is executed, leading to a deviation from the expected behavior, and this deviation becomes apparent to the end-users.
Thus, a visible deviation from the specified or expected behavior is termed a failure in the context of software testing.

## Question: 36

Which of the following characteristics of good testing apply to any software development life cycle model?

A. Acceptance testing is always the final test level to be applied.
B. All test levels are planned and completed for each developed feature.
C. Testers are involved as soon as the first piece of code can be executed.
D. For every development activity there is a corresponding testing activity.

**Answer: D**

Explanation:

The characteristics of good testing practices that apply to any software development life cycle model include the principle that for every development activity, there is a corresponding testing activity. This ensures that testing is integrated into every stage of the software development process, from requirements gathering to maintenance.
According to the ISTQB CTFL Syllabus, version 4.0, this alignment ensures that:
Testing activities are planned alongside development activities.
Testing begins early in the development process to identify and mitigate defects as soon as possible. Testing is not an afterthought but a continuous process that supports quality throughout the lifecycle. This principle helps in identifying defects early, reducing the cost and effort required to fix them later in the development cycle, and ensures comprehensive coverage of both functional and nonfunctional requirements.

## Question: 37

When a defect is detected and fixed then the software should be retested to confirm that the original defect has been successfully removed. This is called:

A. Regression testing
B. Maintenance testing

C. Confirmation testing

D. None of the above

**Answer: C**

Explanation:

When a defect is detected and fixed, the software should be retested to confirm that the original

defect has been successfully removed. This process is called confirmation testing (or re-testing). According to the ISTQB CTFL Syllabus, version 4.0, confirmation testing involves re-running a test to verify that a specific defect has been fixed. The main objective of this type of testing is to ensure that the initial defect no longer exists after the fix has been applied.

The syllabus states that confirmation testing is distinct from regression testing, which checks for unintended side effects of changes in the software. Confirmation testing specifically targets the areas where defects were found and fixed to ensure those defects are resolved.

## Question: 38

Test Implementation and execution has which of the following major tasks?

i. Developing and prioritizing test cases, creating test data, writing test procedures and optionally preparing the test harnesses and writing automated test scripts.

ii. Creating the test suite from the test cases for efficient test execution.

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

iv. Determining the exit criteria.

A. i, ii, iii are true and iv is false

B. i, iv are true and ii is false

C. i, ii are true and iii, iv are false

D. ii, iii, iv are true and i is false

**Answer: A**

Explanation:

Test implementation and execution involve several major tasks, which include:

i. Developing and prioritizing test cases, creating test data, writing test procedures, and optionally preparing the test harnesses and writing automated test scripts. This is crucial for ensuring that all aspects of the software are thoroughly tested and that the tests are prepared systematically.

ii. Creating the test suite from the test cases for efficient test execution. This helps organize the test cases in a way that optimizes the testing process and ensures coverage.

iii. Verifying that the test environment has been set up correctly. Ensuring that the test environment is correctly configured is critical for valid test results.

iv. Determining the exit criteria is part of the test planning phase, not the implementation and execution phase.

According to the ISTQB CTFL Syllabus, version 4.0, these activities are critical for the successful implementation

and execution of tests. Therefore, the statement "i, ii, iii are true and iv is false" accurately reflects the tasks involved in this phase.

## Question: 39

Which of the following statements contains a valuable objective for a test team?

A. Prove that the remaining defects will not cause any additional failures.
B. Run all of the tests that are defined for the test object as quickly as possible.
C. Prove that all faults have been identified through thorough testing.
D. Cause as many failures as possible so that faults can be identified and corrected

**Answer: D**

Explanation:

A valuable objective for a test team is to cause as many failures as possible so that faults can be identified and corrected. This approach is essential because the primary goal of testing is to uncover defects that could potentially lead to failures in the software. By intentionally causing failures, testers can identify and document faults, which can then be fixed to improve the overall quality of the software.

The ISTQB CTFL Syllabus, version 4.0, emphasizes that finding and fixing defects is a core purpose of testing. It is more important to identify and address issues than to prove the absence of defects. Therefore, causing failures to find faults aligns with the fundamental objectives of software testing.

## Question: 40

Which of the following statements is the MOST valid goal for a test team?

A. Determine whether enough component testing was executed.
B. Cause as many failures as possible so that faults can be identified and corrected.
C. Prove that all faults are identified.
D. Prove that any remaining faults will not cause any failures.

**Answer: B**

Explanation:

The most valid goal for a test team is to cause as many failures as possible so that faults can be identified and corrected. This goal aligns with the primary objective of testing, which is to discover defects and ensure they are addressed before the software is released to end-users. By focusing on causing failures, the test team can identify weaknesses in the software and provide valuable feedback for improvements.

According to the ISTQB CTFL Syllabus, version 4.0, the ultimate aim of testing is to find and fix defects to enhance the quality and reliability of the software. Proving that all faults are identified or that no faults will cause failures are unrealistic and less effective goals because it is impossible to guarantee that all defects have been found.

Therefore, the goal of causing failures to find and fix faults is the most valid and practical approach for a test team.

## Question: 41

Which of the following is not a type of incremental testing approach?

A. Top down
B. Big-bang
C. Bottom up
D. Functional incrementation.

**Answer: B**

Explanation:

Incremental testing approaches involve testing components of the software in increments or steps, allowing for integration and testing of parts of the system progressively. According to the ISTQB CTFL Syllabus, version 4.0, the common types of incremental testing approaches include:

Top-down: Testing starts from the top of the module hierarchy and progresses downwards.

Bottom-up: Testing begins with the lower-level modules and progresses upwards.

Functional incrementation: Adding and testing one function at a time.

Big-bang testing, however, is not an incremental approach. In big-bang testing, all components or modules are integrated simultaneously, and the system is tested as a whole. This approach does not allow for incremental testing and integration and is therefore not considered a type of incremental testing approach.

## Question: 42

Which of the following is MOST important in the selection of a test approach?

A. Availability of tools to support the proposed techniques.
B. The budget allowed for training in proposed techniques.
C. Available skills and experience in the proposed techniques.
D. The willingness of the test team to learn new techniques.

**Answer: C**

Explanation:

The selection of a test approach is crucial for the success of the testing process. According to the ISTQB CTFL Syllabus, version 4.0, one of the most important factors in selecting a test approach is the availability of skills and experience in the proposed techniques. This ensures that the team can effectively implement and execute the testing strategies.

While other factors like the availability of tools, budget for training, and willingness to learn new techniques are important, the skills and experience of the team are critical to ensure that the chosen test approach can be executed efficiently and effectively. Without the necessary skills and experience, even the best tools and techniques may not be utilized to their full potential.

## Question: 43

A deviation from the specified or expected behavior that is visible to end-users is called:

A. an error
B. a fault
C. a failure
D. a defect

**Answer: C**

Explanation:

A deviation from the specified or expected behavior that is visible to end-users is called a "failure." According to the ISTQB CTFL Syllabus, version 4.0, the terms are defined as follows:
An error is a human action that produces an incorrect result.
A fault (or defect) is a flaw in a component or system that can cause the system to fail to perform its required function.
A failure occurs when a fault is executed, leading to a deviation from the expected behavior, and this deviation becomes apparent to the end-users.
Therefore, when the deviation is visible to end-users, it is classified as a failure.

## Question: 44

According to the ISTQB Glossary, regression testing is required for what purpose?

A. To verify the success of corrective actions.
B. To prevent a task from being incorrectly considered completed.
C. To ensure that defects have not been introduced by a modification.
D. To motivate better unit testing by the programmers.

**Answer: C**

Explanation:
According to the ISTQB Glossary, regression testing is performed to ensure that changes or modifications in the software have not introduced new defects. The primary purpose of regression testing is to verify that the existing functionality of the software remains unaffected by recent changes, enhancements, or bug fixes.
The syllabus states that regression testing helps in confirming that the recent modifications do not negatively impact the previously working functionalities of the software.

## Question: 45

Maintenance testing is:

A. updating tests when the software has changed
B. testing a released system that has been changed
C. testing by users to ensure that the system meets a business need
D. testing to maintain business advantage

**Answer: B**

Explanation:

Maintenance testing refers to the testing that is conducted on a software system that has been changed, either through corrections, enhancements, or other modifications. Its main purpose is to ensure that the changes have been implemented correctly and that the system as a whole continues

to function as intended. This type of testing is crucial after a system has been released and needs to be updated or fixed. Maintenance testing ensures that the new changes do not adversely affect the existing functionality of the system.REREE33

## Question: 46

Hand over of Testware is a part of which Phase:

A. Test Analysis and Design
B. Test Planning and control
C. Test Closure Activities
D. Evaluating exit criteria and reporting

**Answer: C**

Explanation:

The handover of testware occurs during the Test Closure Activities phase. This phase includes collecting data from completed test activities to consolidate the experience, testware, and any other relevant information for future use. Testware handover involves delivering all test artifacts such as test cases, scripts, and test data to maintenance teams, ensuring that valuable test assets are not lost and can be reused or referenced in future maintenance or development cycles.

## Question: 47

One Key reason why developers have difficulty testing their own work is:

A. Lack of technical documentation
B. Lack of test tools on the market for developers
C. Lack of training
D. Lack of Objectivity

**Answer: D**

Explanation:

Developers often have difficulty testing their own work primarily due to a lack of objectivity. When developers test their own code, they might unconsciously overlook errors because they are too familiar with the code and its intended functionality. This lack of impartiality can lead to missed defects and a biased view of the code's quality. Independent testing by other testers or teams helps to mitigate this issue by providing an objective perspective.

## Question: 48

The purpose of exit criteria is:

A. Define when to stop testing B. End of test level

C. When a set of tests has achieved a specific pre condition D. All of the above

**Answer: D**

Explanation:

Exit criteria are defined to determine when a specific testing activity or the entire testing process should be considered complete. This includes setting conditions such as when to stop testing, the end of a test level, and when certain pre-defined conditions or coverage levels have been achieved. Exit criteria help ensure that testing is systematic and meets the project's quality standards before moving on to the next phase or release.

## Question: 49

What is important to do when working with software development models?

A. To adapt the models to the context of project and product characteristics.
B. To choose the waterfall model because it is the first and best proven model.
C. To start with the V-model and then move to either iterative or incremental models.
D. To only change the organization to fit the model and not vice versa.

**Answer: A**

Explanation:

When working with software development models, it is crucial to adapt the models to the context of the specific project and product characteristics. Different projects and products may require different approaches, and flexibility in choosing or adapting a model ensures that the development process is efficient and effective. This customization allows for better alignment with project requirements, timelines, and risks, ultimately leading to a higher quality product.

## Question: 50

Which statement below BEST describes non-functional testing?

A. The process of testing an integrated system to verify that it meets specified requirements.
B. The process of testing to determine the compliance of a system to coding standards.
C. Testing without reference to the internal structure of a system.
D. Testing system attributes, such as usability, reliability or maintainability.

**Answer: D**

Explanation:

Non-functional testing is focused on testing the attributes of a system such as usability, reliability, maintainability, performance, and security. These attributes define how the system operates rather than what the system does. Non-functional testing ensures that the software meets certain criteria that are critical to its operational performance and user satisfaction.

## Question: 51

For which of the following would maintenance testing be used?

A. Correction of defects during the development phase.
B. Planned enhancements to an existing operational system.
C. Complaints about system quality during user acceptance testing.
D. Integrating functions during the development of a new system.

**Answer: B**

Explanation:

Maintenance testing is performed on an existing operational system that has undergone planned enhancements, corrections, or other modifications. The goal is to ensure that the changes have been implemented correctly and that the system continues to function as intended. This type of testing is not typically used for defect correction during development, user acceptance testing complaints, or the integration of functions in new system development.

## Question: 52

Reporting Discrepancies as incidents is a part of which phase:

A. Test Analysis and Design
B. Test Implementation and execution
C. Test Closure Activities
D. Evaluating exit criteria and reporting

**Answer: B**

Explanation:

Reporting discrepancies as incidents is part of the Test Implementation and Execution phase. During this phase, any deviations from expected results, bugs, or issues discovered are logged as incidents. These incidents are then managed and tracked until they are resolved. This phase is crucial for identifying and documenting problems that occur during the testing process.

## Question: 53

Important consequences of the impossibility of complete testing are:

A. We can never be certain that the program is bug free.

B. We have no definite stopping point for testing, which makes it easier for some managers to argue for very little testing.

C. We have no easy answer for what testing tasks should always be required, because every task takes time that could be spent on other high importance tasks.

D. All of the above

**Answer: D**

Explanation:

The impossibility of complete testing has several significant consequences:
It means that we can never be completely sure that the software is free of bugs (A).
Without a definite stopping point, it becomes easier for some managers to justify minimal testing efforts (B).
Since every testing task requires time and resources, it becomes challenging to determine which tasks are always necessary (C).
These points highlight the need for a balanced and risk-based approach to testing, where priorities are set based on the context and criticality of the software being tested.

## Question: 54

Which of the following is the main purpose of the component build and integration strategy?

A. to ensure that all of the small components are tested
B. to ensure that the system interfaces to other systems and networks
C. to ensure that the integration testing can be performed by a small team
D. to specify how the software should be divided into components
E. to specify which components to combine when, and how many at once

**Answer: E**

Explanation:

The main purpose of the component build and integration strategy is to specify which components to combine when, and how many at once. This strategy helps manage the complexity of integrating different parts of the system by providing a structured approach to combining components incrementally. It ensures that integration is done systematically, reducing the risk of integration issues and making it easier to identify and resolve problems as they arise.
Top of Form
Bottom of Form

## Question: 55

What should be the MAIN objective during development testing?

A. To cause as many failures as possible so that defects in the software are identified and can be fixed
B. To confirm that the system works as expected and that requirements have been met
C. To assess the quality of the software with no intention of fixing defects

D. To give information to stakeholders of the risk of releasing the system at a given time

**Answer: A**

Explanation:
The main objective during development testing is to cause as many failures as possible to identify defects in the software. This proactive approach ensures that defects are discovered and addressed early in the development process, improving the overall quality and reliability of the final product.

## Question: 56

Which of the following is not a part of the Test Implementation and Execution Phase?

A. Creating test suites from the test cases
B. Executing test cases either manually or by using test execution tools
C. Comparing actual results
D. Designing the Tests

**Answer: D**

Explanation:

Designing the tests is part of the Test Analysis and Design phase, not the Test Implementation and Execution phase. The Test Implementation and Execution phase involves creating test suites, executing test cases, and comparing actual results with expected outcomes to identify any discrepancies.

## Question: 57

Designing the test environment set-up and identifying any required infrastructure and tools are a part of which phase:

A. Test Implementation and execution
B. Test Analysis and Design
C. Evaluating the Exit Criteria and reporting
D. Test Closure Activities

**Answer: B**

Explanation:

Designing the test environment setup and identifying any required infrastructure and tools are part of the "Test Analysis and Design" phase. According to the ISTQB Certified Tester Foundation Level Syllabus v4.0, the activities in this phase include creating high-level and detailed test designs, test cases, and test data, and specifying the necessary test environment, tools, and infrastructure. These preparations are essential for the subsequent phases of test execution and monitoring.

## Question: 58

Which of the following statements BEST describes one of the seven key principles of software testing?

A. Automated tests are better than manual tests for avoiding the Exhaustive Testing.
B. Exhaustive testing is, with sufficient effort and tool support, feasible for all software.
C. It is normally impossible to test all input / output combinations for a software system.
D. The purpose of testing is to demonstrate the absence of defects. The purpose of testing is to demonstrate the absence of defects.

**Answer: C**

Explanation:

One of the seven key principles of software testing, as outlined in the ISTQB Certified Tester Foundation Level Syllabus v4.0, states that exhaustive testing is impossible except for trivial cases. This principle emphasizes that it is not feasible to test all possible input/output combinations due to the sheer number of permutations, and therefore, testing should be risk-based and prioritized to ensure the most critical aspects are tested effectively.

## Question: 59

What is the benefit of independent testing?

A. More work gets done because testers do not disturb the developers all the time.
B. Independent testers tend to be unbiased and find different defects than the developers
C. Independent testers do not need extra education and training.
D. Independent testers reduce the bottleneck in the incident management process.

**Answer: B**

Explanation:
Independent testing brings objectivity and a fresh perspective, which can uncover defects that developers might overlook due to familiarity with the system. The ISTQB Certified Tester Foundation Level Syllabus v4.0 explains that independent testers are less likely to be influenced by the same assumptions as the developers, thus increasing the likelihood of detecting unique and different defects.

## Question: 60

What is the purpose of a test completion criterion?

A. to know when a specific test has finished its execution
B. to ensure that the test case specification is complete
C. to set the criteria used in generating test inputs
D. to determine when to stop testing

**Answer: D**

Explanation:

A test completion criterion is used to determine when testing activities can be concluded. According to the ISTQB Certified Tester Foundation Level Syllabus v4.0, setting test completion criteria helps in defining when the testing objectives have been achieved, which includes reaching a defined coverage level, running out of time or budget, or meeting the expected level of risk acceptance. These criteria are essential for ensuring that the testing process is concluded appropriately and resources are utilized effectively.

## Question: 61

Which activities form part of test planning?

i) Developing test cases.
ii) Defining the overall approach to testing.
iii) Assigning resources.
iv) Building the test environment.
v) Writing test conditions.

A. i, ii & iv are true, iii & v are false.
B. ii & iii are true, i, iv & v are false.
C. iv & v are true, i, ii & iii are false.
D. i, ii & iii are true iv & v are false.

**Answer: B**

Explanation:

Test planning involves defining the overall approach to testing and assigning resources. According to the ISTQB CTFL syllabus v4.0, the activities that form part of test planning include determining the scope, approach, resources, and schedule of testing activities. Developing test cases, building the test environment, and writing test conditions are activities that belong to the later phases of the testing process, such as test design and implementation.

## Question: 62

Testing should be stopped when:

A. All the planned tests have been run
B. Time has run out
C. All faults have been fixed correctly
D. Both A and C
E. It depends on the risks for the system being tested

**Answer: E**

Explanation:

Testing should be stopped based on risk criteria defined for the system being tested. The ISTQB CTFL syllabus v4.0

outlines that the decision to stop testing is influenced by risk levels, coverage achieved, time and budget constraints, and the criticality of the system under test. Stopping criteria should ensure that the risk of remaining defects is acceptable for the project stakeholders.

## Question: 63

Pick the best definition of quality:

A. Quality is job one
B. Zero defects
C. Conformance to requirements
D. Work as designed

**Answer: C**

Explanation:

Quality in software testing is best defined as conformance to requirements. The ISTQB CTFL syllabus v4.0 states that quality is the degree to which a component, system, or process meets specified requirements and/or user/customer needs and expectations.

## Question: 64

Which of these are objectives for software testing?

A. Determine the productivity of programmers
B. Eliminate the need for future program maintenance
C. Eliminate every error prior to release
D. Uncover software errors

**Answer: D**

Explanation:

One of the primary objectives of software testing is to uncover software errors. The ISTQB CTFL syllabus v4.0 emphasizes that testing aims to find defects, verify that the software works as expected, and ensure that it meets the specified requirements. Other objectives include preventing defects, evaluating the quality of the software, and providing information to stakeholders.

## Question: 65

Consider the following statements about early test design:

i. Early test design can prevent fault multiplication
ii. Faults found during early test design are more expensive to fix
iii. Early test design can find faults
iv. Early test design can cause changes to the requirements

v. Early test design takes more effort

A. i, iii & iv are true. ii & v are false
B. iii is true, i, ii, iv & v are false
C. iii & iv are true. i, ii & v are false
D. i, iii, iv & v are true, ii us false
E. i & iii are true, ii, iv & v are false

**Answer: A**

Explanation:

Early test design can prevent fault multiplication, find faults, and cause changes to the requirements. The ISTQB CTFL syllabus v4.0 explains that starting test design early in the lifecycle helps to identify defects sooner, which can prevent the propagation of defects into later stages and reduce the overall cost of fixing them. While it can influence the requirements, it does not necessarily take more effort, and faults found early are typically less expensive to fix than those found later.

## Question: 66

Which of the following are the typical defects found by static analysis tools?

a. Variables that are never used.
b. Security vulnerabilities.
c. Poor performance.
d. Unreachable code.
e. Business processes not followed.

A. b, c and d are true; a and e are false
B. a is true; b, c, d and e are false
C. c, d and e are true; a and b are false
D. a, b and d are true; c and e are false

**Answer: D**

Explanation:

Static analysis tools are used to detect certain types of defects without executing the code. According to the ISTQB CTFL syllabus v4.0, typical defects found by static analysis tools include variables that are never used, security vulnerabilities, and unreachable code. Poor performance and business processes not followed are not typically identified by static analysis tools.

## Question: 67

During the software development process, at what point can the test process start?

A. When the code is complete.
B. When the design is complete.
C. When the software requirements have been approved.

D. When the first code module is ready for unit testing

**Answer: C**

Explanation:
The test process can start as soon as the software requirements have been approved. The ISTQB CTFL syllabus v4.0 emphasizes that early involvement of testing helps in identifying defects early in the lifecycle, which includes the validation of requirements and planning of testing activities based on those requirements.

## Question: 68

Which is not a major task of test implementation and execution?

A. Develop and prioritizing test cases, creating test data, writing test procedures and optionally, preparing test harness and writing automated test scripts.
B. Logging the outcome of test execution and recording the identities and versions of the software under test, test tools and testware.
C. Checking test logs against the exit criteria specified in test planning.
D. Verifying that the test environment has been set up correctly.

**Answer: C**

Explanation:
Checking test logs against the exit criteria specified in test planning is not a major task of test implementation and execution. This activity is more related to evaluating exit criteria and reporting. Major tasks of test implementation and execution include developing and prioritizing test cases, creating test data, writing test procedures, optionally preparing test harnesses, and writing automated test scripts, logging the outcome of test execution, and verifying that the test environment has been set up correctly.

## Question: 69

Which of the following could be a reason for a failure?

1) Testing fault
2) Software fault
3) Design fault
4) Environment Fault
5) Documentation Fault

A. 2 is a valid reason; 1, 3, 4 & 5 are not
B. 1, 2, 3, 4 are valid reasons; 5 is not
C. 1, 2, 3 are valid reasons; 4 & 5 are not
D. All of them are valid reasons for failure

**Answer: D**

Explanation:

Failures in software can be caused by various reasons, including testing faults, software faults, design faults,

environment faults, and documentation faults. The ISTQB CTFL syllabus v4.0 discusses that these types of issues can all contribute to failures in software, highlighting the importance of thorough testing and review processes throughout the software development lifecycle.

## Question: 70

Which is not the testing objective?

A. Finding defects
B. Gaining confidence about the level of quality and providing information
C. Preventing defects.
D. Debugging defects

**Answer: D**

Explanation:

Debugging defects is not a testing objective. According to the ISTQB CTFL syllabus v4.0, the primary objectives of testing include finding defects, gaining confidence about the level of quality, providing information, and preventing defects. Debugging is a developer's activity that involves identifying the cause of defects and fixing them, which is different from the objectives of testing.

## Question: 71

Which of the following is usually the test basis for integration testing?

A. Program specification
B. Functional specification
C. Technical specification
D. Requirement specification

**Answer: C**

Explanation:

The test basis for integration testing typically includes technical specifications. According to the ISTQB CTFL syllabus v4.0, integration testing focuses on the interactions between components or systems, and the technical specification provides the necessary details about these interactions and interfaces.

## Question: 72

Which of the following are examples of iterative development models?

(i) V-model
(ii) Rapid Application Development model
(iii) Waterfall model
(iv) Agile development model

A. (i) and (ii)

B. (ii) and (iii)
C. (ii) and (iv)
D. (iii) and (iv)

**Answer: C**

Explanation:

Iterative development models include the Rapid Application Development (RAD) model and Agile development model. The ISTQB CTFL syllabus v4.0 explains that these models involve repeated cycles of development activities, allowing for incremental improvements and adjustments based on feedback.

## Question: 73

Which of the following is not true of regression testing?

A. It can be carried out at each stage of the life cycle.
B. It serves to demonstrate that the changed software works as intended.
C. It serves to demonstrate that software has not been unintentionally changed.
D. It is often automated.

**Answer: B**

Explanation:

Regression testing serves to demonstrate that unchanged areas of the software have not been unintentionally affected by changes. The ISTQB CTFL syllabus v4.0 states that while regression testing can be carried out at various stages of the lifecycle and is often automated, its primary purpose is to ensure that existing functionality remains unaffected after changes.

## Question: 74

One of the roles in a review is that of moderator, which of the following best describes this role?

A. Plans the review, runs the review meeting and ensures that follow-up activities are completed.
B. Allocates time in the plan, decides which reviews will take place and that the benefits are delivered.
C. Writes the document to be reviewed, agrees that the document can be reviewed, and updates the document with any changes.
D. Documents all issues raised in the review meeting, records problems and open points.

**Answer: A**

Explanation:

The moderator in a review is responsible for planning the review, running the review meeting, and ensuring that follow-up activities are completed. This role is crucial for the effectiveness of the review process, as highlighted in

the ISTQB CTFL syllabus v4.0, which details the responsibilities and importance of the moderator in facilitating a productive and structured review.

## Question: 75

What do static analysis tools analyze?

A. Design
B. Test cases
C. Requirements
D. Program code

**Answer: D**

Explanation:

Static analysis tools analyze the program code to identify potential issues such as syntax errors, unreachable code, and variables that are never used. These tools perform their analysis without executing the code, thus allowing early detection of defects. The ISTQB CTFL syllabus v4.0 highlights that static analysis tools focus on the code itself to find these types of defects.

## Question: 76

Which of the following is most likely to be a benefit of using static techniques?

A. Fewer performance defects.
B. Productivity improvements in the development process.
C. More efficient regression testing.
D. Quick return on investment in static analysis tools.

**Answer: B**

Explanation:

One of the most likely benefits of using static techniques is productivity improvements in the development process. The ISTQB CTFL syllabus v4.0 states that static techniques, such as reviews and static analysis, help in identifying defects early, which reduces the time and effort required for rework, thus improving overall productivity in the development process.

## Question: 77

Which of the following are static techniques?

A. Walkthrough.
B. State transition testing.
C. Decision table testing.
D. Statement testing.

**Answer: A**

Explanation:
Walkthroughs are a type of static technique used in software testing. According to the ISTQB CTFL syllabus v4.0, static techniques include activities such as reviews, inspections, and walkthroughs, where the evaluation of the software product is done without executing the code.

## Question: 78

Which one of the following roles is typically used in a review?

A. Champion.
B. Author.
C. Project sponsor.
D. Custodian.

**Answer: B**

Explanation:

The Author is the only role that is typically used in a review.
A Champion might sponsor the review process but is not a defined role within an actual review; a Project Sponsor, if technically competent, might be asked to play a defined role within the review process, but whilst using that role they will not be a Project Sponsor; finally, a Custodian might ensure the results are stored safely but would not be involved in the actual review itself.

## Question: 79

Which of the following defines the expected result of a test?

A. Test case
B. Test procedure
C. Test execution schedule
D. Test condition

**Answer: A**

Explanation:

A test case defines the expected result of a test. The ISTQB CTFL syllabus v4.0 explains that a test case includes a set of input values, execution preconditions, expected results, and execution postconditions developed for a particular objective or test condition.

## Question: 80

Which of the following describes structure-based (white-box) test case design techniques?

A. Test cases are derived systematically from models of the system.
B. Test cases are derived systematically from the tester's experience.
C. Test cases are derived systematically from the delivered code.
D. Test cases are derived from the developers' experience.

**Answer: C**

Explanation:

Structure-based (white-box) test case design techniques involve deriving test cases systematically from the delivered code. The ISTQB CTFL syllabus v4.0 describes that these techniques focus on the internal structure of the software, using knowledge of the code to create test cases that ensure comprehensive coverage of the code's pathways and logic.

## Question: 81

Which of the following is a structure-based (white-box) technique?

A. Decision table testing
B. State transition testing
C. Statement testing
D. Boundary value analysis

**Answer: C**

Explanation:

Statement testing is a structure-based (white-box) technique. This technique involves testing the internal structure of the application, focusing on the execution of the statements within the code. The goal is to ensure that every statement in the code has been executed at least once during the testing process. Other options like decision table testing, state transition testing, and boundary value analysis are black-box techniques, which do not require knowledge of the internal code structure. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.1.

## Question: 82

What is the main purpose of use case testing?

A. To identify defects in process flows related to typical use of the system.
B. To identify defects in the connections between components.
C. To identify defects in the system related to extreme scenarios.
D. To identify defects in the system related to the use of unapproved programming practices.

**Answer: A**

Explanation:

The main purpose of use case testing is to identify defects in process flows related to the typical use of the system. Use case testing is a black-box test design technique that is based on use case scenarios. These scenarios help in validating that the system performs correctly for the specific ways in which the users will interact with it. This method is particularly effective in finding issues that users might encounter during real-world

usage of the system. Reference: ISTQB CTFL Syllabus V4.0, Section 4.2.4.

## Question: 83

What is the purpose of exit criteria?

A. To identify how many tests to design.
B. To identify when to start testing.
C. To identify when to stop testing.
D. To identify who will carry out the test execution.

**Answer: C**

Explanation:
The purpose of exit criteria is to determine when to stop testing. Exit criteria are specific conditions or set of conditions agreed upon to declare that a particular test level or a test activity is completed. These criteria are used to ensure that the testing process has reached a certain level of coverage and quality before it is considered complete. Typical exit criteria include achieving a certain level of code coverage, passing a specified number of tests, or fixing a certain number of defects. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.3 .

## Question: 84

What can a risk-based approach to testing provide?

A. The types of test techniques to be employed.
B. The total tests needed to provide 100 per cent coverage.
C. An estimation of the total cost of testing.
D. Only that test execution is effective at reducing risk.

**Answer: A**

Explanation:

A risk-based approach to testing provides information on the types of test techniques to be

employed. This approach prioritizes the testing of features and functions based on their risk of failure and their impact on the organization. By identifying the areas of highest risk, test managers can allocate resources and select test techniques that are most likely to uncover defects in these critical areas. This approach ensures that testing efforts are focused where they are most needed.
Reference: ISTQB CTFL Syllabus V4.0, Section 5.2 .

## Question: 85

When assembling a test team to work on an enhancement to an existing system, which of the following has the highest level of test independence?

A. A business analyst who wrote the original requirements for the system.
B. A permanent programmer who reviewed some of the new code, but has not written any of it.
C. A permanent tester who found most defects in the original system.

D. A contract tester who has never worked for the organization before.

**Answer: D**

Explanation:

In this scenario, the contract tester who has never worked for the organization before has the highest level of test independence. The three others are less independent as they are likely to make assumptions based on their previous knowledge of the requirements, code and general functionality of the original system.

Note that independence does not necessarily equate to most useful. In practice most test or project managers would recruit a permanent tester who has worked on the original system in preference to a contract tester with no knowledge of the system. However, when assembling a team it would be useful to have staff with varying levels of test independence and system knowledge.

## Question: 86

Which of the following terms is used to describe the management of software components comprising an integrated system?

A. Configuration management
B. Incident management
C. Test monitoring
D. Risk management

**Answer: A**

Explanation:

Incident management is the collection and processing of incidents raised when errors and defects are discovered. Test monitoring identifies the status of the testing activity on a continuous basis. Risk management identifies, analyses and mitigates risks to the project and the product. Configuration management is concerned with the management of changes to software components and their

associated documentation and testware.

## Question: 87

A new system is about to be developed. Which of the following functions has the highest level of risk?

A. Likelihood of failure = 20%; impact value = £100,000
B. Likelihood of failure = 10%; impact value = £150,000
C. Likelihood of failure = 1%; impact value = £500,000
D. Likelihood of failure = 2%; impact value = £200,000

**Answer: D**

Explanation:

The highest level of risk is calculated by considering both the likelihood of failure and the impact value. Risk can be quantified as the product of these two factors:

A: 20% * £100,000 = £20,000
B: 10% * £150,000 = £15,000
C: 1% * £500,000 = £5,000
D: 2% * £200,000 = £4,000

Therefore, Option A (20% likelihood of failure and £100,000 impact value) represents the highest risk value. Reference: ISTQB CTFL Syllabus V4.0, Section 5.2.3 .

## Question: 88

Which of the following statements about risks is most accurate?

A. Project risks rarely affect product risk.
B. Product risks rarely affect project risk.
C. A risk-based approach is more likely to be used to mitigate product rather than project risks.
D. A risk-based approach is more likely to be used to mitigate project rather than product risks.

**Answer: C**

Explanation:

In general, project risk and product risk can be hard to differentiate. Anything that impacts on the quality of the delivered system is likely to lead to delays or increased costs as the problem is tackled. Anything causing delays to the project is likely to threaten the delivered system's quality. The riskbased approach is an approach to managing product risk through testing, so it impacts most directly on product risk.

## Question: 89

For which of the following activities in the fundamental test process would an incident management

tool be most useful?

A. Test planning and control
B. Test analysis and design
C. Test implementation and execution
D. Evaluating exit criteria and reporting

**Answer: C**

Explanation:

Incident management tools are most useful during test implementation and execution as this is the stage at which the tool is used to raise, manage, retest and close incidents.

The data collected during the defect life cycle can then be manipulated into information that is useful for other activities within the fundamental test process.

Information on numbers of defects outstanding may be useful for evaluating exit criteria (option (D)). This information could also be used for planning future testing and for taking control (option (A)).

Incident management tools can also assist in test analysis and design (option (B)) as information about defects found when testing the previous release of the system could be used when analyzing the type of testing required for the next enhancement.

## Question: 90

Which of the following defects is most likely to be found by a test harness?

A. Variance from programming standards.
B. A defect in middleware.
C. Memory leaks.
D. Regression defects.

**Answer: B**

Explanation:

Variance from programming standards defects (option (A)) are found during the review or static testing process. Therefore a test harness is unlikely to find a defect in programming standards.

Memory leak defects (option (C)) could potentially be found by a test harness designed to run many test cases.

Regression defects (option (D)) could be found using many types of test tool.

Defects in middleware (option (B)) are generally more likely to be found by a test harness or a dynamic analysis tool than by any other type of tool.

## Question: 91

A test management tool is most likely to integrate with which of the following tools?

A. Performance testing tool
B. Test data preparation tool
C. Static analysis tool
D. Requirements management tool

**Answer: D**

Explanation:

Requirements management tools (option (D)) often have interfaces with test management tools. In some cases

they will be sold as a package or in other cases a test management tool may have its own requirements module. The use of such interfaces or integrated packages aids traceability from requirements through to test scripts and defects.

Performance management tools (option (A)), test data preparation tools (option (B)) and static analysis tools (option (C)) are unlikely to have an interface or be integrated with a test management tool. They serve different purposes and therefore there is little need for such interfaces.

## Question: 92

Which of the following are aids to good communication, and which hinder it?

i. Try to understand how the other person feels.
ii. Communicate personal feelings, concentrating upon individuals.
iii. Confirm the other person has understood what you have said and vice versa.
iv. Emphasize the common goal of better quality.
v. Each discussion is a battle to be won.

A. (i), (ii) and (iii) aid, (iv) and (v) hinder.
B. (iii), (iv) and (v) aid, (i) and (ii) hinder.
C. (i), (iii) and (iv) aid, (ii) and (v) hinder.
D. (ii), (iii) and (iv) aid, (i) and (v) hinder.

**Answer: C**

Explanation:

Good communication is aided by trying to understand how the other person feels (i), confirming mutual understanding (iii), and emphasizing common goals such as better quality (iv). These actions foster a collaborative environment and ensure that all parties are on the same page. On the other hand, communicating personal feelings concentrating upon individuals (ii) and viewing each discussion as a battle to be won (v) hinder communication because they create a confrontational and

subjective atmosphere. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.2 .

## Question: 93

Which pair of definitions is correct?

A. Regression testing is checking that the reported defect has been fixed; retesting is testing that there are no additional problems in previously tested software.
B. Regression testing is checking there are no additional problems in previously tested software; retesting enables developers to isolate the problem.
C. Regression testing involves running all tests that have been run before; retesting runs new tests. D. Regression testing is checking that there are no additional problems in previously tested software, retesting is demonstrating that the reported defect has been fixed.

**Answer: D**

Explanation:

Regression testing is testing that nothing has regressed. Retesting (or confirmation testing) confirms the fix is correct by running the same test after the fix has been made. No other option has both of these as true.

## Question: 94

The following statements relate to activities that are part of the fundamental test process.

i. Evaluating the testability of requirements.
ii. Repeating testing activities after changes.
iii. Designing the test environment set-up.
iv. Developing and prioritizing test cases.
v. Verifying the environment is set up correctly.

Which statement below is TRUE?

A. (i) and (ii) are part of analysis and design, (iii), (iv) and (v) are part of test implementation and execution.
B. (i) and (iii) are part of analysis and design, (ii), (iv) and (v) are part of test implementation and execution.
C. (i) and (v) are part of analysis and design, (ii), (iii) and (iv) are part of test implementation and execution.
D. (i) and (iv) are part of analysis and design, (ii), (iii) and (v) are part of test implementation and execution.

**Answer: D**

Explanation:

Evaluating the testability of requirements (i) and developing and prioritizing test cases (iv) are part of the test analysis and design activities. Repeating testing activities after changes (ii), designing the test environment set-up (iii), and verifying the environment is set up correctly (v) are part of the test implementation and execution activities. These distinctions are part of the fundamental test process, which ensures that each phase of testing is appropriately planned and executed. Reference: ISTQB CTFL Syllabus V4.0, Section 1.4.1.

## Question: 95

Which statement correctly describes the public and profession aspects of the code of ethics?

A. Public: Certified software testers shall act in the best interests of their client and employer (being consistent with the wider public interest). Profession: Certified software testers shall advance the integrity and reputation of their industry consistent with the public interest.
B. Public: Certified software testers shall advance the integrity and reputation of the profession consistent with the public interest. Profession: Certified software testers shall consider the wider public interest in their actions.
C. Public: Certified software testers shall consider the wider public interest in their actions. Profession: Certified software testers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of their profession.

D. Public: Certified software testers shall consider the wider public interest in their actions.

Profession: Certified software testers shall advance the integrity and reputation of their industry consistent with the public interest.

**Answer: D**

Explanation:

The code of ethics for certified software testers includes considering the wider public interest in their actions for the public aspect, and advancing the integrity and reputation of their industry consistent with the public interest for the professional aspect. This ensures that testers uphold high standards of ethical conduct and professionalism, contributing positively to the public and the industry.

Reference: ISTQB CTFL Syllabus V4.0, Code of Ethics Section.

## Question: 96

Which of the following is true about the V-model?

A. It has the same steps as the waterfall model for software development.
B. It is referred to as a cyclical model for software development.
C. It enables the production of a working version of the system as early as possible.
D. It enables test planning to start as early as possible.

**Answer: D**

Explanation:

The V-model enables test planning to start as early as possible. This model is an extension of the waterfall model and emphasizes the importance of early test planning and development of test

cases, which corresponds to the development stages. By planning tests early, the V-model ensures that testing is integrated into every phase of the software development lifecycle. Reference: ISTQB CTFL Syllabus V4.0, Section 2.1.2.

## Question: 97

Which of the following is true of iterative development?

A. It uses fully defined specifications from the start.
B. It involves the users in the testing throughout.
C. Changes to the system do not need to be formally recorded.
D. It is not suitable for developing websites.

**Answer: B**

Explanation:

Iterative development involves users in the testing throughout the development process. This approach allows for continuous feedback from users, ensuring that the evolving system meets user requirements and expectations. Unlike the traditional waterfall model, iterative development accommodates changes and improvements based on

user feedback, making it suitable for dynamic environments like website development. Reference: ISTQB CTFL Syllabus V4.0, Section 2.1.3.

## Question: 98

A top-down development strategy affects which level of testing most?

A. Component testing
B. Integration testing
C. System testing
D. User acceptance testing

**Answer: B**

Explanation:

The development strategy will affect the component testing (option (A)), in so far as it cannot be tested unless it has been built. Options (C) and (D) require the system to have been delivered; at these points the development strategy followed is not important to the tester. Option (B) needs knowledge of the development strategy in order to determine the order in which components will be integrated and tested.

## Question: 99

Which of the following is a non-functional requirement?

A. The system will enable users to buy books.
B. The system will allow users to return books.
C. The system will ensure security of the customer details.
D. The system will allow up to 100 users to log in at the same time.

**Answer: C**

Explanation:
Non-functional requirements define how the system performs a certain function, rather than specific behaviors or functions of the system. Security, performance, usability, reliability, and maintainability are common examples of non-functional requirements. Ensuring the security of customer details is a non-functional requirement because it specifies a quality attribute that the system must have.
Reference: ISTQB CTFL Syllabus V4.0, Section 4.1.

## Question: 100

Which of the following statements are true?

(i) For every development activity there is a corresponding testing activity.
(ii) Each test level has the same test objectives.
(iii) The analysis and design of tests for a given test level should begin after the corresponding development activity.

(iv) Testers should be involved in reviewing documents as soon as drafts are available in the development life cycle.

A. (i) and (ii)
B. (iii) and (iv)
C. (ii) and (iii)
D. (i) and (iv)

**Answer: D**

Explanation:
For every development activity, there is a corresponding testing activity to ensure that testing is integrated into the development process. Additionally, involving testers early in the document review process allows for the identification of defects and issues early in the lifecycle, which aligns with the principles of early and continuous testing. Reference: ISTQB CTFL Syllabus V4.0, Section 1.4.1.

## Question: 101

Which of the following statements are correct for walkthroughs?

(i) Often led by the author.
(ii) Documented and defined results.
(iii) All participants have defined roles.
(iv) Used to aid learning.
(v) Main purpose is to find defects.
A. (i) and (v) are correct.
B. (ii) and (iii) are correct.
C. (i) and (iv) are correct.
D. (iii) and (iv) are correct.

**Answer: C**

Explanation:
Walkthroughs are often led by the author of the document and are used to aid learning and understanding among team members. These informal review meetings help in discussing the content and obtaining feedback from participants, but they are less formal than inspections and do not necessarily involve defined roles for all participants or result in documented outcomes. Reference: ISTQB CTFL Syllabus V4.0, Section 3.2.1.

## Question: 102

Which of the following has the typical formal review activities in the correct sequence?

A. Kick-off, review meeting, planning, follow-up.
B. Kick-off, planning, review meeting, re-work.
C. Planning, kick-off, individual preparation, review meeting.
D. Planning, individual preparation, follow-up, re-work.

**Answer: C**

Explanation:

The typical sequence of formal review activities includes planning, kick-off, individual preparation, and review meeting. The planning phase sets the scope and objectives, the kick-off initiates the review process, individual preparation involves participants reviewing the materials on their own, and then the review meeting is held to discuss the findings. Reference: ISTQB CTFL Syllabus V4.0, Section 3.2.2.

## Question: 103

Which of the following statements are true?

(i) Defects are likely to be found earlier in the development process by using reviews rather than static analysis.

(ii) Walkthroughs require code but static analysis does not require code.

(iii) Informal reviews can be performed on code and specifications.

(iv) Dynamic techniques are generally used before static techniques.

(v) Dynamic techniques can only be used after code is ready to be executed.

A. (i), (ii), (vi).

B. (ii), (iii), (v).

C. (i), (iv), (v).

D. (i), (iii), (v).

**Answer: D**

Explanation:

The other answers are incorrect because:

(11) Walkthroughs do not require code and static analysis does require code.

(iv) Static techniques do not execute the code and therefore can be run before and after the code is ready for execution.

## Question: 104

Which of the following is most likely to be performed by developers?

A. Technical review of a functional specification.

B. Walkthrough of a requirements document.

C. Informal review of a program specification.

D. Static analysis of a software model.

**Answer: D**

Explanation:

Static analysis of a software model is most likely to be performed by developers. This involves examining the code and software structure without executing the program, using tools to identify potential issues, such as coding standards violations, security vulnerabilities, and other quality issues. Developers often use static analysis as part

of their development process to ensure code quality and maintainability. Reference: ISTQB CTFL Syllabus V4.0, Section 3.1.1.

## Question: 105

Which of the following are most characteristic of structure-based testing?

(i) Information about how the software is constructed is used to derive test cases.
(ii) Statement coverage and/or decision coverage can be measured for existing test cases.
(iii) The knowledge and experience of people are used to derive test cases.
(iv) Test cases are derived from a model or specification of the system.

A. (i) and (ii)
B. (ii) and (iii)
C. (ii) and (iv)
D. (i) and (iii)

**Answer: A**

Explanation:

Structure-based testing, also known as white-box testing, uses information about the software's internal structure to derive test cases. This method often involves measuring coverage, such as statement coverage or decision coverage, to ensure that different parts of the code are exercised by the tests. These techniques require a deep understanding of the software's construction and are used to validate the implementation rather than the functionality. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.1.

## Question: 106

Which of the following are the most important factors to be taken into account when selecting test techniques?

(i) Tools available.
(ii) Regulatory standards.
(iii) Experience of the development team.
(iv) Knowledge of the test team.

The need to maintain levels of capability in each technique.

A. (i) and (ii)
B. (ii) and (iv)
C. (iii) and (iv)
D. (i) and (v)

**Answer: B**

Explanation:

Answer (i) looks temptingly right, and the availability of tools might make the use of a technique more or less

attractive, but it would not be decisive in the way that regulatory standards and tester knowledge are.

Answer (iii) is irrelevant because testing should be independent of development anyway, but it could tempt someone who is unsure about the relationship between development and testing.

Answer (v) is a factor in managing the test team, and experience would need to be maintained, but this should not influence the selection of techniques for a live project.

## Question: 107

Which of the following are most likely to be used when developing a test strategy or test approach?

(i) Failure-based approach
(ii) Test specification approach

(iii) Model-based approach (iv)Finance-based approach

A. (iii) and (ii)
B. (i) and (iv)
C. (ii) and (i)
D. (i) and (iii)

**Answer: D**

Explanation:
Developing a test strategy or test approach often involves using a failure-based approach and a model-based approach. The failure-based approach focuses on anticipating potential failures and designing tests to expose these failures. The model-based approach uses models of the system's behavior or structure to design test cases systematically. These approaches help in creating a robust and comprehensive test strategy. Reference: ISTQB CTFL Syllabus V4.0, Section 5.2.2.

## Question: 108

What test roles (or parts in the testing process) is a developer most likely to perform?

(i) Executing component integration tests.
(ii) Static analysis.
(iii) Setting up the test environment.
(iv) Deciding how much testing should be automated.

A. (i) and (ii)
B. (i) and (iv)
C. (ii) and (iii)
D. (iii) and (iv)

**Answer: A**

Explanation:

(i) Executing component integration tests is usually done by developers. Developers are usually responsible for unit and component integration testing. Independent testing usually follows at system and acceptance test levels.

(ii) Static analysis is usually done by developers because: it requires an understanding of the code and therefore the person doing this needs skills in the programming language; and it can be done as soon as the code is written. Therefore it is quick and effective for the developer to do it. The risk of a lack of test independence can be mitigated by performing independent system and acceptance testing.

(iii) Setting up the test environment is an activity typically performed by a tester. It may require support from developers and staff from other departments and on some occasions environments

could be set up by developers. However, it is a task that could be done by a tester rather than a developer.

(iv) Deciding how much testing should be automated is typically a decision made by the test leader, who will consult other staff in the decision-making process. Developers may be involved and their skills may be required to automate some tests. However, the decision on how much to automate should not be made by developers.

## Question: 109

Which of the following are valid justifications for developers testing their own code during unit testing?

(i) Their lack of independence is mitigated by independent testing during system and acceptance testing.
(ii) A person with a good understanding of the code can find more defects more quickly using whitebox techniques.
(iii) Developers have a better understanding of the requirements than testers.
(iv) Testers write unnecessary incident reports because they find minor differences between the way in which the system behaves and the way in which it is specified to work.

A. (i) and (ii)
B. (i) and (iv)
C. (ii) and (iii)
D. (iii) and (iv)

**Answer: A**

Explanation:

It is unlikely that developers will have a better understanding of the requirements than testers, partly because testers work closely with the user community (and may be drawn from it) and partly because developers seldom work with the complete set of requirements in a medium to large development.

Testers may raise incidents related to the difference between user expectations and the specification, but these are not unnecessary. Such issues are more likely to arise at the later stages of testing.

Early testing (unit testing) is usually done most effectively by developers with a good understanding of the code and the development environment; they can be more efficient and more effective at this level. Later independent stages of testing offset any disadvantage from the lack of independence at unit testing level.

## Question: 110

Which of the following pairs of test tools are likely to be most useful during the test analysis and design stage of the fundamental test process?

(i) Test execution tool
(ii) Test data preparation tool
(iii) Test management tool
(iv) Requirements management tool

A. (i) and (ii)
B. (i) and (iv)
C. (ii) and (iii)
D. (iii) and (iv)

**Answer: D**

Explanation:

During the test analysis and design stage, test management tools and requirements management tools are most useful. Test management tools help in organizing and managing the testing activities, including planning, tracking, and reporting test progress. Requirements management tools assist in ensuring that the test cases are aligned with the requirements, thereby improving traceability and coverage. These tools support the efficient design and management of tests. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.2.

## Question: 111

Which of the following is most likely to cause failure in the implementation of a test tool?

A. Underestimating the demand for a tool.
B. The purchase price of the tool.
C. No agreed requirements for the tool.
D. The cost of resources to implement and maintain the tool.

**Answer: C**

Explanation:

## Question: 112

What benefits do static analysis tools have over test execution tools?

A. Static analysis tools find defects earlier in the life cycle.
B. Static analysis tools can be used before code is written.
C. Static analysis tools test that the delivered code meets business requirements.
D. Static analysis tools are particularly effective for regression testing.

**Answer: A**

Explanation:

## Question: 113

Which of the following principles should be followed when introducing a test tool into an organization?

(i) Assessing organizational maturity to establish whether a tool will provide expected benefits.
(ii) Requiring a quick payback on the initial investment.
(iii) Including a requirement for the tool to be easy to use without having to train unskilled testers.
(iv) Identifying and agreeing requirements before evaluating test tools.

A. (i) and (ii)
B. (i) and (iv)
C. (ii) and (iii)
D. (iii) and (iv)

**Answer: B**

Explanation:

Assessing organizational maturity (i) is very important when deciding whether to introduce a test tool, as implementing a tool in an immature test organization with poor processes is unlikely to produce any benefits.

A quick return on the initial investment (ii) in a test tool is rare.

Having a requirement that a tool should be easy to use for untrained and unskilled testers (iii) is generally a false hope. This is comparable with expecting someone who has never driven a car to be able to drive safely and effectively.

Agreeing requirements before evaluating tools (iv) is essential. Not to do so would be comparable with building and testing a system without requirements.

In conclusion, (i) and (iv) are good principles to follow when introducing a tool and (ii) and (iii) are not.

## Question: 114

How can test execution tools be of most benefit during exploratory testing?

A. They can record user actions so that defects are easier to recreate.
B. They can be used to perform the regression aspects of exploratory testing.
C. They can help to mitigate the risk of low test coverage.
D. They can use data-driven tests to increase the amount of exploratory testing performed.

**Answer: A**

Explanation:

Exploratory testing is used when it is unclear what the system is supposed to do. Therefore test execution tools are of little use because expected results cannot be predicted.

However, the record feature of a test execution tool can be used to log the actions performed so that defects can be recreated (option (A)) and rectified more easily.

## Question: 115

Which of the following types of test tool are most likely to include traceability functions?

(i) Performance testing tool
(ii) Requirements management tool
(iii) Configuration management tool
(iv) Static analysis tool

A. (i) and (ii)
B. (i) and (iv)
C. (ii) and (iii)
D. (iii) and (iv)

**Answer: C**

Explanation:

Requirements management tools (ii) have traceability because they enable test conditions and subsequently test scripts and defects to be traced back to requirements. Configuration management tools (iii) also need to trace the appropriate version of a test script to the release or version of a system or module.

Performance monitoring tools (i) and static analysis tools (iv) are designed for specific objectives. Neither of these tools particularly need traceability functions.

## Question: 116

A system is designed to accept values of examination marks as follows:

Fail: 0–39 inclusive
Pass: 40–59 inclusive
Merit: 60–79 inclusive
Distinction: 80–100 inclusive

In which of the following sets of values are all values in different equivalence partitions?

A. 25, 40, 60, 75
B. 0, 45, 79, 87
C. 35, 40, 59, 69
D. 25, 39, 60, 81

**Answer: B**

Explanation:
Equivalence partitioning is a testing technique that divides input data into partitions of equivalent data from which test cases can be derived. The goal is to ensure that at least one value from each partition is tested. The partitions for the examination marks are:
Fail: 0–39
Pass: 40–59
Merit: 60–79
Distinction: 80–100
In set B (0, 45, 79, 87), each value belongs to a different equivalence partition:
0 (Fail)
45 (Pass)
79 (Merit)
87 (Distinction)
Reference: ISTQB CTFL Syllabus V4.0, Section 4.2.1.

## Question: 117

A washing machine has three temperature bands for different kinds of fabrics: fragile fabrics are washed at temperatures between 15 and 30 degrees Celsius; normal fabrics are washed at temperatures between 31 and 60 degrees Celsius; heavily soiled and tough fabrics are washed at temperatures between 61 and 100 degrees Celsius.

Which of the following contains only values that are in different equivalence partitions?

A. 15, 30, 60
B. 20, 35, 60
C. 25, 45, 75
D. 12, 35, 55

**Answer: C**

Explanation:

Using equivalence partitioning, we divide the temperature ranges into the following partitions:
Fragile fabrics: 15–30
Normal fabrics: 31–60
Heavily soiled and tough fabrics: 61–100
In set C (25, 45, 75), each value belongs to a different equivalence partition:
25 (Fragile fabrics)
45 (Normal fabrics)
75 (Heavily soiled and tough fabrics)
Reference: ISTQB CTFL Syllabus V4.0, Section 4.2.1.

## Question: 118

Consider the following pseudo code:

```
1  Begin
2  Read Time
3  If Time < 12 Then
4  Print(Time, "am")
5  Endif
6  If Time > 12 Then
7  Print(Time -12, "pm")
8  Endif
9  If Time = 12 Then
10 Print (Time, "noon")
11 Endif
12 End
```

How many test cases are needed to achieve 100 per cent decision coverage?

A. 1
B. 2
C. 3
D. 4

**Answer: C**

Explanation:

The three decisions are in sequence and the conditions are all mutually exclusive (if any one is true the others must be false). Hence a test case that makes the first decision true will make the second and third decisions false and so on.

So test case 1 (say Time = 6) would exercise the path True, False, False, test case 2 (say Time = 15) would exercise the path False, True, False. Test case 3 would have to be Time = 12. This combination achieves 100 per cent decision coverage because each decision has been exercised through its true and its false outcomes.

## Question: 119

Consider the following pseudo code:

```
1  Begin
2  Read Time
3  If Time < 12 Then
4  Print(Time, "am")
5  Endif
6  If Time > 12 Then
7  Print(Time -12, "pm")
8  Endif
9  If Time = 12 Then
```

```
10  Print (Time, "noon")
11  Endif
12  End
```

If the test cases Time = 11 and Time = 15 were input, what level of decision coverage would be achieved?

A. 100% or 6/6
B. 50% or 3/6
C. 67% or 4/6
D. 83% or 5/6

**Answer: D**

Explanation:

Test case 1 exercises the decision outcomes True, False, False
Test case 2 exercises the decision outcomes False, True, False
This leaves the True outcome of decision 3 not exercised.
Of the 6 possible decision outcomes, 5 have been exercised, so the decision coverage is 5/6 (about 83%).

## Question: 120

A software component has the code shown below:

```
Program BiggestA,
Biggest: Integer
Begin
Read A
Biggest = 10
While A > 0
Do
If A > Biggest
Then Biggest = A
Endif
Read A
Enddo
End
```

The component has exit criteria for component testing that include 100% statement coverage. Which of the following test cases will satisfy this criterion?

A. 0
B. 10, 0
C. 10, 5, 0
D. 10, 11, 0

**Answer: D**

Explanation:

100% statement coverage means that every statement in the code must be executed at least once. To achieve this, we need to ensure that the condition If A > Biggest is both true and false at least once during the test.

Test case D (10, 11, 0) will cover all the statements:

Initial Biggest set to 10.

When A is 10, the condition A > Biggest is false.

When A is 11, the condition A > Biggest is true, and Biggest is updated to 11.

When A is 0, the loop exits.

This covers all possible paths through the code, ensuring 100% statement coverage. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.1.

## Question: 121

Given the Following program IF X <>= ZTHEN Statement 2;ENDMcCabe's Cyclomatic Complexity is :

A. 2
B. 3
C. 4
D. 5

**Answer: B**

Explanation:

McCabe's Cyclomatic Complexity is a metric used to measure the complexity of a program. It is calculated using the formula:

Cyclomatic Complexity=E-N+2P\text{Cyclomatic Complexity} = E - N +

2PCyclomatic Complexity=E-N+2P

where:

E = number of edges in the flow graph

N = number of nodes in the flow graph

P = number of connected components (usually 1 for a single program)

In the given program, there are 2 decision points (IF and THEN). The cyclomatic complexity is calculated as:

Cyclomatic Complexity=2-1+2=3\text{Cyclomatic Complexity} = 2 - 1 + 2 =

3Cyclomatic Complexity=2-1+2=3

Hence, the answer is B. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.2.

## Question: 122

An input field takes the year of birth between 1900 and 2004. The boundary values for testing this field are:

A. 0,1900,2004,2005

B. 1900, 2004

C. 1899,1900,2004,2005

D. 1899, 1900, 1901,2003,2004,2005

**Answer: C**

Explanation:

Boundary value analysis focuses on the values at the edges of equivalence partitions. For an input range of 1900 to 2004, the boundary values are just inside and just outside the range:

Just inside: 1900, 2004

Just outside: 1899, 2005

Therefore, the boundary values are 1899, 1900, 2004, and 2005. Reference: ISTQB CTFL Syllabus V4.0, Section 4.2.1.

## Question: 123

Code Coverage is used as a measure of what?

A. Defects
B. Trends analysis
C. Test Effectiveness
D. Time Spent Testing

**Answer: C**

Explanation:

Code Coverage is a measure used to describe the degree to which the source code of a program is tested by a particular test suite. It helps in determining the effectiveness of testing by indicating the parts of the code that have been exercised by the tests and those that have not. It is an important metric for evaluating test coverage and identifying areas that may require additional testing. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.2.

## Question: 124

How many test cases are necessary to cover all the possible sequences of statements (paths) for the following program fragment?

Assume that the two conditions are independent of each other : -if (Condition 1)then statement 1else statement 2if (Condition 2)then statement 3

A. 2 Test Cases
B. 3 Test Cases
C. 4 Test Cases
D. Not achievable

**Answer: C**

Explanation:

To cover all possible sequences of statements (paths) in the given program fragment, considering two independent conditions, we have the following scenarios:

Condition 1 true, Condition 2 true

Condition 1 true, Condition 2 false

Condition 1 false, Condition 2 true

Condition 1 false, Condition 2 false

These scenarios require 4 test cases to ensure all paths are tested. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.3.

## Question: 125

To test a function, the programmer has to write a _____, which calls the function to be tested and passes it test data:

A. Stub
B. Driver
C. Proxy
D. None of the above

**Answer: B**

Explanation:

A driver is a piece of code that calls a function or module to be tested and passes it the necessary test data. It is commonly used in bottom-up integration testing to simulate the higher-level modules that invoke the function being tested. Stubs, on the other hand, are used to simulate lower-level modules. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.1.

## Question: 126

In a review meeting a moderator is a person who:

A. Takes minutes of the meeting
B. Mediates between people
C. Takes telephone calls
D. Writes the documents to be reviewed

**Answer: B**

Explanation:

In a review meeting, the moderator is responsible for ensuring the meeting runs smoothly, mediating between participants, and managing time effectively. The moderator's role includes

facilitating communication and making sure all participants have the opportunity to contribute. Taking minutes of the meeting is typically done by a recorder or scribe, not the moderator.

Reference: ISTQB CTFL Syllabus V4.0, Section 3.2.4.

## Question: 127

Incorrect form of Logic coverage is:

A. Statement Coverage
B. Pole Coverage
C. Condition Coverage
D. Path Coverage

**Answer: B**

Explanation:

Logic coverage includes different forms such as Statement Coverage, Condition Coverage, and Path Coverage. Pole Coverage is not a recognized form of logic coverage in software testing. The three correct forms measure different aspects of the code's logic to ensure thorough testing. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.1.

## Question: 128

Independent Verification & Validation is:

A. Done by the Developer
B. Done by the Test Engineers
C. Done By Management
D. Done by an Entity Outside the Project's sphere of influence

**Answer: D**

Explanation:

Independent Verification & Validation (IV&V) is performed by an entity outside the project's sphere of influence to provide an unbiased assessment of the software product. This independence helps ensure that the verification and validation processes are objective and not influenced by the project's internal dynamics. Reference: ISTQB CTFL Syllabus V4.0, Section 1.4.1.

## Question: 129

Boundary value testing:

A. Is the same as equivalence partitioning tests
B. Test boundary conditions on, below and above the edges of input and output equivalence classes
C. Tests combinations of input circumstances
D. Is used in white box testing strategy

**Answer: B**

Explanation:

Boundary value testing focuses on testing at the boundaries between partitions. It tests conditions on, below, and above the edges of input and output equivalence classes to identify defects that occur at the boundaries. This technique is different from equivalence partitioning, which divides input data into partitions but does not specifically target boundary conditions. Reference: ISTQB CTFL Syllabus V4.0, Section 4.2.1.

## Question: 130

Acceptance test cases are based on what?

A. Requirements
B. Design
C. Code
D. Decision table

**Answer: A**

Explanation:

Acceptance test cases are derived from requirements to ensure that the system meets the agreed- upon criteria and works as expected from an end-user perspective. These tests validate that the system satisfies business needs and requirements before it is accepted for deployment. Reference: ISTQB CTFL Syllabus V4.0, Section 4.5.3.

## Question: 131

Which of the following is not a quality characteristic listed in ISO 9126 Standard?

A. Functionality
B. Usability
C. Supportability
D. Maintainability

**Answer: C**

Explanation:

ISO 9126 defines a quality model that includes the following quality characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability. Supportability is not listed as one of the quality characteristics in the ISO 9126 standard. This standard focuses on the aforementioned characteristics to evaluate the quality of software products. Reference: ISTQB CTFL Syllabus V4.0, Section 2.4.

## Question: 132

Fault Masking is:

A. Error condition hiding another error condition
B. Creating a test case which does not reveal a fault
C. Masking a fault by developer
D. Masking a fault by a tester

**Answer: A**

Explanation:

Fault masking occurs when one defect prevents the detection of another defect. In other words, an error condition hides another error condition, making it more difficult to identify and correct all defects. This can happen in various scenarios, such as when a certain path of code execution is blocked by an existing error, preventing other potential errors from being executed and detected. Reference: ISTQB CTFL Syllabus V4.0, Section 1.5.5.

## Question: 133

Statement Coverage will not check for the following:

A. Missing Statements
B. Unused Branches
C. Dead Code
D. Unused Statement

**Answer: A**

Explanation:
Statement coverage measures the percentage of executable statements in the code that have been executed. However, it does not check for missing statements since these are not present in the code to be executed in the first place. Statement coverage can identify unused branches, dead code, and unused statements within the code. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.2.

## Question: 134

"How much testing is enough?"

A. This question is impossible to answer
B. This question is easy to answer
C. The answer depends on the risk for your industry, contract and special requirements
D. This answer depends on the maturity of your developers

**Answer: C**

Explanation:
The amount of testing that is considered enough depends on various factors, including the level of risk associated with the application, industry standards, contractual obligations, and specific requirements. It is a risk-based decision that balances the need for thorough testing with time and resource constraints. Reference: ISTQB CTFL Syllabus V4.0, Section 5.2.2.

## Question: 135

Which of the following is not decided in the test-planning phase..

A. Schedules and deliverables
B. Hardware and software
C. Entry and exit criteria
D. Types of test cases

**Answer: D**

Explanation:

The test-planning phase involves deciding on schedules and deliverables, hardware and software requirements, and entry and exit criteria. However, the specific types of test cases to be used are typically defined during the test design phase, not during the test planning phase. Test planning focuses on the overall approach and resources needed for testing. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.1.

## Question: 136

What is the concept of introducing a small change to the program and having theeffects of that change show up in some test..?

A. Introducing mutations
B. Performance testing
C. A mutation error
D. Debugging a program

**Answer: A**

Explanation:
The concept of introducing a small change to the program and having the effects of that change show up in some test is known as "introducing mutations." This is part of mutation testing, where small changes (mutations) are made to the code to see if the existing test cases can detect the errors introduced by these changes. The goal is to evaluate the effectiveness of the test cases in identifying defects. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.4.

## Question: 137

Which is the best definition of complete testing..?

A. You have discovered every bug in the program
B. You have tested every statement, branch, and combination of branches in the program
C. You have reached the scheduled ship date
D. You have completed every test in the test plan

**Answer: B**

Explanation:

Complete testing is best defined as having tested every statement, branch, and combination of branches in the program. This definition aligns with achieving thorough code coverage, ensuring that all parts of the code have

been exercised by tests. However, it is important to note that complete testing in a practical sense is rarely achievable, as it assumes exhaustive testing of all possible scenarios. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.2.

## Question: 138

Security falls under..?

A. compliance testing
B. disaster testing
C. verifying compliance to rules
D. functional testing
E. ease of operations

**Answer: D**

Explanation:
Security testing falls under functional testing. It involves verifying that the system protects data and maintains functionality as intended, even in the presence of malicious inputs or attacks. Security testing is essential to ensure that the system meets its security requirements and is capable of resisting unauthorized access or other security breaches. Reference: ISTQB CTFL Syllabus V4.0, Section 2.4.3.

## Question: 139

What if the project isn't big enough to justify extensive testing..?

A. Use automation tool for testing
B. Use risk based analysis to find out which areas need to be tested
C. Both a and b
D. None of the above

**Answer: B**

Explanation:
If the project isn't big enough to justify extensive testing, using risk-based analysis to find out which areas need to be tested is the best approach. Risk-based testing prioritizes testing activities based on the risk of failure and the impact of those failures, ensuring that critical areas of the application are tested thoroughly even with limited resources. This approach helps in focusing testing efforts where they are most needed. Reference: ISTQB CTFL Syllabus V4.0, Section 5.2.3.

## Question: 140

Complete statement and branch coverage means..?

A. That you have tested every statement in the program
B. That you have tested every statement and every branch in the program
C. That you have tested every IF statement in the program
D. That you have tested every combination of values of IF statements in the program

**Answer: B**

Explanation:
Complete statement and branch coverage means that you have tested every statement and every branch in the program. Statement coverage ensures that every executable statement in the code is executed at least once, while branch coverage ensures that every possible path (true and false conditions) through the program's branches (such as IF statements) is executed. Achieving both covers a significant portion of the code, providing a thorough validation of its logic. Reference: ISTQB CTFL Syllabus V4.0, Section 4.3.2.

## Question: 141

In the MASPAR case study..?

A. Security failures were the result of untested parts of code
B. The development team achieved complete statement and branch coverage but missed a serious bug in the MASPAR operating system
C. An error in the code was so obscure that you had to test the function with almost every input value to find its two special-case failures
D. All the above

**Answer: D**

Explanation:
In the MASPAR case study, several key issues were highlighted. Security failures were attributed to untested parts of code (A), despite achieving complete statement and branch coverage, a serious bug was still missed in the MASPAR operating system (B). Additionally, an obscure error required testing the function with nearly every input value to find the two special-case failures (C). Therefore, all these points together (D) illustrate the complexity and challenges in software testing. Reference:
ISTQB CTFL Syllabus V4.0.

## Question: 142

Which is not in sequence in 11 Step Software Testing process..

A. Assess development plan and status
B. Develop the test plan
C. Test software design
D. Test software requirement

**Answer: C**

Explanation:

In the 11 Step Software Testing process, the sequence typically involves assessing the development plan and status (A), developing the test plan (B), and testing software requirements (D) early in the process. Testing software design (C) is generally done after the requirements have been tested and the test plan has been developed, making it out of sequence if placed before these steps. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.

## Question: 143

Important consequences of the impossibility of complete testing are ..?

A. We can never be certain that the program is bug free
B. We have no definite stopping point for testing, which makes it easier for some managers to argue for very little testing
C. We have no easy answer for what testing tasks should always be required, because every task takes time that could be spent on other high importance tasks
D. All of the above

**Answer: D**

Explanation:
The impossibility of complete testing has several important consequences. It means we can never be certain that the program is entirely bug-free (A). There is no definite stopping point for testing, which can make it easier for some managers to justify minimal testing (B). Additionally, there is no simple answer for which testing tasks should always be required, as each task takes time that could be allocated to other critical activities (C). Therefore, all these consequences (D) are valid and important considerations in software testing. Reference: ISTQB CTFL Syllabus V4.0, Section 2.2.

## Question: 144

Tools like change Man, Clear case are used as..

A. functional automation tools
B. performance testing tools
C. configuration management tools
D. none of the above

**Answer: C**

Explanation:
Change Man and Clear Case are examples of configuration management tools. These tools are used to manage changes in the software, track revisions, and control the versions of the code and documentation throughout the software development lifecycle. They help ensure that the integrity and consistency of the product are maintained as it evolves. Reference: ISTQB CTFL Syllabus V4.0, Section 5.5.2.

## Question: 145

What do you mean by “Having to say NO”..?

A. No, the problem is not with testers
B. No, the software is not ready for production
C. Both a & b
D. None of the above

**Answer: B**

Explanation:
"Having to say NO" in the context of software testing typically means informing stakeholders that the software is not ready for production deployment. Testers have the responsibility to ensure that the software meets the required quality standards before it is released. Saying "no" to the release of software that does not meet these standards is crucial to prevent potential issues in the production environment. This decision is based on the identification of critical defects or incomplete functionality that could adversely affect the user experience or system performance.

## Question: 146

The selection of test cases for regression testing..?

A. Requires knowledge on the bug fixes and how it affect the system
B. Includes the area of frequent defects
C. Includes the area which has undergone many/recent code changes D. All of the above

**Answer: D**

Explanation:
The selection of test cases for regression testing requires a comprehensive approach. It involves: A . Knowledge on the bug fixes and how they affect the system to ensure that new changes do not reintroduce old defects.

B . Includes the area of frequent defects to prioritize testing on parts of the software that have a history of issues, thus ensuring that known problem areas are thoroughly tested.

C . Includes the area which has undergone many/recent code changes to verify that new changes have not adversely affected existing functionality.

Combining all these aspects ensures a robust regression testing process, which minimizes the risk of defects in the software after changes are made.

## Question: 147

What are the key features to be concentrated upon when doing a testing forworld wide web sites ..

A. Interaction between html pages
B. Performance on the client side
C. Security aspects
D. All of the above

**Answer: D**

Explanation:
When testing worldwide web sites, it is crucial to focus on several key features:

A . Interaction between HTML pages to ensure that navigation and link functionalities work correctly. B . Performance on the client side to make sure that the website loads efficiently and performs well under different conditions.

C . Security aspects to protect the website and its users from potential vulnerabilities and attacks. These combined

aspects ensure that the website provides a good user experience, performs well, and remains secure.

## Question: 148

What if the project isn't big enough to justify extensive testing..?

A. Use risk based analysis to find out which areas need to be tested
B. Use automation tool for testing
C. Both a and b
D. None of the above

**Answer: A**

Explanation:

If the project isn't big enough to justify extensive testing, the best approach is to use risk-based analysis. This method prioritizes testing efforts based on the risk of failure and the impact of defects. It helps in identifying critical areas that need more focus, thus optimizing the testing process by concentrating on high-risk components and functionalities.

## Question: 149

Who is responsible for conducting test readiness review..

A. Project Manager
B. Test Engineer
C. Test Manager
D. None of the above

**Answer: C**

Explanation:

The responsibility of conducting a test readiness review typically falls to the Test Manager. The Test Manager ensures that all necessary preparations for testing have been completed and that the testing environment, resources, and schedule are ready for the test execution phase. This role involves coordinating with different stakeholders to confirm that the testing process can proceed smoothly.

## Question: 150

Faults found by users are due to..

A. Poor quality software
B. Poor software and poor testing
C. Bad luck
D. Insufficient time for testing

**Answer: B**

Explanation:

Faults found by users are typically due to a combination of poor software quality and inadequate testing. Poor software quality can result from defects in the code, design, or requirements, while poor testing can fail to identify these defects before the software is released. Therefore, both factors contribute to the presence of faults in the software that users encounter.

## Question: 151

Which of the following is the odd one out..

A. White box
B. Glass box
C. Structural
D. Functional

**Answer: D**

Explanation:

The odd one out among the given options is "Functional". The other options—White box, Glass box, and Structural—refer to testing techniques that involve examining the internal structure of the application. In contrast, functional testing focuses on testing the software against its functional requirements without considering its internal structures.

## Question: 152

When what is visible to end-users is a deviation from the specific or expected behavior, this is called..?

A. An error
B. A fault
C. A failure
D. A defect
E. A mistake

**Answer: C**

Explanation:

When what is visible to end-users is a deviation from the specific or expected behavior, it is called a failure. A failure occurs when the software does not perform a function as expected or produces incorrect results, which can be observed by the users. This is distinct from an error, which is a human action that produces an incorrect result, a defect (or bug), which is a flaw in the software, or a fault, which is another term for a defect.

## Question: 153

Which of the following is NOT part of configuration management..

A. Status accounting of configuration items
B. Auditing conformance to ISO9001
C. Identification of test versions
D. Record of changes to documentation over time
E. Controlled library access

**Answer: B**

Explanation:

Configuration management involves practices like status accounting of configuration items, identification of test versions, record of changes to documentation over time, and controlled library access. However, auditing conformance to ISO9001 is related to quality management and not specifically part of configuration management.

## Question: 154

A tool that supports traceability, recording of incidents or scheduling of tests is called..

A. A dynamic analysis tool
B. A test execution tool
C. A debugging tool
D. A test management tool
E. A configuration management tool

**Answer: D**

Explanation:

A tool that supports traceability, recording of incidents, or scheduling of tests is called a test management tool. These tools help in planning, managing, and tracking the testing process, providing features for managing test cases, tracking defects, and maintaining traceability between test artifacts.

Top of Form

Bottom of Form

Topic 2, Mix Questions Set B

## Question: 155

The four test levels defined for a common V-model testing approach are:

A. Unit, integration, system and maintenance.
B. Functional, glass box, incremental and maintenance.

C. Component, integration, system and acceptance.
D. Unit, component, functional and alpha/beta.

**Answer: C**

Explanation:

The four test levels defined for a common V-model testing approach are Component, Integration, System, and Acceptance testing. These levels align with the development stages in the V-model, where each level of testing corresponds to a specific phase in the software development lifecycle, ensuring comprehensive verification and validation of the software.

## Question: 156

Typically, exit criteria may consist of:

A. Defining the amount, level of detail structure, and templates for the test documentation.
B. Estimates of defect density or reliability measures.
C. Adequacy of the test approaches taken.
D. Discussions on disaster recovery.

**Answer: B**

Explanation:

Typically, exit criteria may consist of various metrics that indicate whether testing has been sufficiently thorough. Estimates of defect density or reliability measures are common exit criteria used to decide if the software is ready for the next stage or release. These criteria help in assessing the quality and stability of the software based on the number and severity of defects found during testing.

## Question: 157

A data driven approach to test automation design is best described as:

A. Using action words to describe the actions to be taken, the test data.
B. Scaling to support large numbers of users.
C. Being based on Equivalence Partitioning testing techniques.
D. Separating out the test data inputs and using a generic script that can read the test data and perform the same test steps with different data.

**Answer: D**

Explanation:
A data-driven approach to test automation design involves separating the test data inputs from the test scripts. This approach allows the same generic script to read different sets of test data and execute the same test steps with various inputs. This method enhances reusability and efficiency in automated testing by allowing testers to run multiple test scenarios using the same script but different data sets.

## Question: 158

Which of the following risks represents the highest level of risk to the project?

A. Likelihood of failure = 1%, potential cost of impact = $1m.
B. Likelihood of failure = 10%, potential cost of impact = $500,000.
C. Likelihood of failure = 20%, potential cost of impact = $150,000.
D. Likelihood of failure = 5%, potential cost of impact = $500,000.

**Answer: B**

Explanation:
When evaluating risks, both the likelihood of failure and the potential cost of impact must be considered. The risk with the highest level is determined by the product of these two factors. In this case, a 10% likelihood of failure with a potential cost of impact of $500,000 results in a higher risk (expected impact) compared to the other options. The expected impact is calculated as likelihood multiplied by impact, which is higher for option B.
Top of Form

Bottom of Form

## Question: 159

What factors should be considered to determine whether enough testing has been performed?

(i) The exit criteria.
(ii) The budget.
(iii) How big the test team is.
(iv) The product's riskprofile.
(v) How good the testing tools are.
(vi) Sufficient details of the system status to allow decisions

A. i and ii and iv and vi B. i and ii and iii and vi C. ii and iii and iv and v D. i and ii and v and vi

**Answer: A**

Explanation:
Determining whether enough testing has been performed involves considering several factors. These include: (1) The exit criteria, which define the conditions that must be met to conclude testing. (ii) The budget, which affects the extent of testing that can be conducted. (iv) The product's risk profile, which helps prioritize testing efforts on high-risk areas. (vi) Sufficient details of the system status to allow decisions, ensuring that stakeholders have the necessary information to make informed decisions about the software's readiness.
These factors together ensure a balanced and informed decision-making process regarding the sufficiency of testing.

## Question: 160

Which of the following statements is most true about test conditions?

A. An item or event of a component or system that can be verified by one or more test cases.

B. The grouping of a composite set of test cases which, when tested as a whole, reveal a positive or negative result.

C. A testable component derived from business requirements.

D. Applies to software testing only.

**Answer: A**

Explanation:

A test condition is defined as an item or event of a component or system that can be verified by one or more test cases. This means that a test condition is something that can be tested to determine if it meets the expected behavior. It forms the basis for creating test cases that will verify the condition's correctness.

## Question: 161

Consider the following pseudo code

1. Begin
2. Read Gender
3. Print "Dear"
4. If Gender = 'female'
5. Print ("Ms")
6. Else
7. Print ( "Mr")
8. Endif
9. End

How many test cases are needed to achieve 100 per cent decision coverage?

A. 1

B. 2

C. 3

D. 4

**Answer: B**

Explanation:

To achieve 100% decision coverage, each decision point in the pseudo code must be tested for both true and false outcomes. In the provided pseudo code, the decision point is the if-else statement checking the gender. Therefore, we need two test cases: one where Gender is 'female' and one where Gender is not 'female' (e.g., 'male'). This ensures that both paths (true and false) are covered.

## Question: 162

What is static analysis?

A. The decision between using white or black box test techniques.
B. Executing software to validate the most common path through the code.
C. A technique to find defects in software source code and software models, performed without executing code.
D. It is a testing technique used during system testing.

**Answer: C**

Explanation:
Static analysis is a technique used to find defects in software source code and software models without executing the code. It involves reviewing the code or models to identify potential issues, such as syntax errors, logical errors, or security vulnerabilities. Static analysis helps improve code

quality and maintainability by catching defects early in the development process.
Top of Form
Bottom of Form

## Question: 163

Component testing may include:

A. Sociability testing.
B. User acceptance testing.
C. Beta testing.
D. The use of stubs and drivers.

**Answer: B**

Explanation:

Component testing, also known as unit testing, involves testing individual components of the software separately. This type of testing focuses on the functionality of a specific section of the code. To facilitate this process, stubs and drivers are often used. Stubs are dummy modules that simulate the behavior of lower-level modules that are not yet integrated. Drivers are used to simulate higher- level modules that control the component being tested.
Reference: ISTQB CTFL Syllabus, version 4.0, mentions the use of stubs and drivers in the context of component testing.

## Question: 164

Which type of review has the following main purposes:
discussing, making decisions, evaluating alternatives, finding defects, solving technical problems and checking conformance to specifications, plans, regulations, and standards?

A. Technical Review
B. Inspection
C. Walkthrough
D. Informal review

**Answer: A**

Explanation:
A technical review is a formal type of review that focuses on discussing, making decisions, evaluating alternatives, finding defects, solving technical problems, and checking conformance to specifications, plans, regulations, and standards. It is typically conducted by a team of experts and aims to ensure the technical quality of the product.
Reference: ISTQB CTFL Syllabus, version 4.0, defines technical review with the described main purposes.

## Question: 165

Consider the following pseudo code:

1. Begin
2. Input X, Y
3. If X > Y
4. Print (X, 'is greater than', Y)
5. Else
6. Print (Y, is greater than or equal to', X)
7. EndIf
8. End

What is the minimum number of test cases required to guarantee both 100% statement coverage and 100% decision coverage?

A. Statement coverage = 3, Decision coverage = 3
B. Statement coverage = 2, Decision coverage = 2
C. Statement coverage = 1, Decision coverage = 2
D. Statement coverage = 2, Decision coverage = 1

**Answer: B**

Explanation:
To achieve 100% statement coverage and 100% decision coverage for the provided pseudo code, at least two test cases are required. The first test case should cover the scenario where X > Y, which will execute the statements in the "if" branch. The second test case should cover the scenario where X ≤ Y, which will execute the statements in the "else" branch. This ensures that both the true and false branches of the decision are tested, and all statements in the code are executed at least once.
Reference: The ISTQB CTFL Syllabus, version 4.0, discusses the need for different test cases to ensure statement and decision coverage.

## Question: 166

Which of the following best describes the Black-box technique?

A. It uses decision coverage for completeness.
B. It ensures all possible branches in the code are tested.

C. It is based on the internal structure of the system.
D. It can be done without reference to the internal structure of the component or system.

**Answer: D**

Explanation:
The Black-box testing technique is a method where the internal structure or workings of the item being tested are not known to the tester. This technique focuses on the input and output of the software system, ensuring that it meets the specified requirements without considering the internal code structure. It contrasts with White-box testing, which involves testing internal structures or workings of an application.
Reference: ISTQB CTFL Syllabus, version 4.0, explains that Black-box testing is conducted without reference to the internal structure of the system.

## Question: 167

System testing is:

A. Used to search for defects in software modules that are separately testable.
B. The responsibility of the users of a system.
C. Concerned with the behavior of a whole system/product as defined by the scope of a development project.
D. Triggered by modifications, migration or retirement of the software system.

**Answer: C**

Explanation:
System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System testing is conducted on the entire system to verify that it behaves as expected and meets the functional and non-functional requirements defined in the scope of the project.
Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes that system testing is concerned with the behavior of the whole system/product.

## Question: 168

A client-server system for a web development must support a minimum of 200 enquiries per hour. In peak times, it must be available 24 hours x 7 days due to the critical nature of the application, and must have a response time lower than 20 seconds during peak loads.

Which of the following set of test types would be most appropriate to verify the non-functional requirements expressed in this statement?

A. Performance, Usability, Regression.
B. System, Load, Fail-over.
C. Performance, Load, Stress.
D. Load, Volume and Component.

**Answer: C**

Explanation:

To verify the non-functional requirements such as the support for a minimum number of enquiries per hour, availability, and response time during peak loads, the most appropriate set of test types would include Performance Testing to assess the responsiveness and stability, Load Testing to verify the system's ability to handle expected user loads, and Stress Testing to determine the system's behavior under extreme conditions. These tests collectively ensure that the system meets the

specified non-functional criteria.

Reference: ISTQB CTFL Syllabus, version 4.0, outlines the importance of these test types in assessing non-functional requirements.

## Question: 169

When an organization considers the use of testing tools, they should:

A. Use a tool in order to help define a good test process because the tool will force process repeatability and therefore enforce good test process.

B. Always start by bringing in automated test execution tools as these tools have the greatest return on investment and therefore should be introduced first.

C. Perform analysis of the test process and then assess whether it can be supported through the introduction of tool support.

D. Allow the developers to select the testing tools because tools are technical and developers have the appropriate skills to advise on test tool selection and configuration.

**Answer: C**

Explanation:

When considering the use of testing tools, it is essential to first perform an analysis of the current test process. This analysis helps identify areas where tool support can provide the most benefit. Introducing tools without this assessment might lead to inefficiencies or the selection of inappropriate tools. A thorough evaluation ensures that the tools will effectively support and enhance the existing test process.

Reference: ISTQB CTFL Syllabus, version 4.0, advises conducting a detailed analysis of the test process before introducing tools.

## Question: 170

Which of the following test design techniques is not a black box technique?

A. Equivalence partitioning
B. State transition testing
C. Boundary value analysis
D. Statement coverage

**Answer: D**

Explanation:

Statement coverage is a white-box testing technique that involves executing all the statements in the code at least once. It is not considered a black-box technique because it requires knowledge of the internal structure of the code being tested. Black-box techniques, such as equivalence partitioning, state transition testing, and boundary value analysis, focus on testing the functionality of the software without considering its internal code structure.

Reference: ISTQB CTFL Syllabus, version 4.0, differentiates between black-box and white-box testing techniques, clearly identifying statement coverage as a white-box technique.

## Question: 171

Which of the following are valid test objectives?

(i) Finding defects.
(ii) Gaining confidence about the level of quality and providing information.
(iii) Preventing defects.
(iv) Debugging the code.

A. i, ii and iii
B. i, ii and iv
C. ii and iii
D. i and iv

**Answer: A**

Explanation:

Valid test objectives include finding defects (i), gaining confidence about the level of quality and providing information (ii), and preventing defects (iii). Debugging the code (iv) is not a valid test objective as it is a part of the development process rather than the testing process.

Reference: ISTQB CTFL Syllabus, version 4.0, lists the valid objectives of testing, which include finding defects, gaining confidence, and preventing defects.

## Question: 172

What can testing demonstrate?

A. Testing can demonstrate that there are no defects
B. Testing can demonstrate that there are defects
C. Testing can demonstrate that the software is correct
D. Testing can demonstrate that there are no hidden defects in the software

**Answer: B**

Explanation:

Testing is primarily used to identify defects in the software. While it can show the presence of defects, it cannot guarantee that no defects are present. Thus, testing helps in demonstrating that there are defects, which can then be addressed.

Reference: ISTQB CTFL Syllabus, version 4.0, explains that testing can demonstrate the presence of defects but cannot prove their absence.

## Question: 173

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 testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. It helps in reducing the number of test cases while maintaining adequate test coverage.

Reference: ISTQB CTFL Syllabus, version 4.0, defines Equivalence Partitioning as dividing inputs into groups expected to exhibit similar behaviors.

## Question: 174

Dynamic Analysis Tools are used to:

A. Determine differences between files or databases.
B. Monitor and report on how a system behaves under a variety of conditions.
C. Find defects, such as memory leaks, while software is executing.
D. Measure the percentage of specific types of code structure that have been exercised.

**Answer: C**

Explanation:
Dynamic Analysis Tools are used during the execution of software to find defects that can occur while the software is running. These tools can identify issues such as memory leaks, which are not detectable by static analysis tools.

Reference: ISTQB CTFL Syllabus, version 4.0, details the use of dynamic analysis tools for identifying defects during software execution.

## Question: 175

What other details should be included in the following incident report when it is first submitted?

Date of Issue: 23/11/05

Severity: P1
Build: Version15.6
Details: Expected field to be limited to 15 chars, able to enter 27

A. Suggested solution, priority and number of defects assigned to this developer.
B. Status of the incident, degree of impact, Test Case Number.
C. History, related defects and expected fix time.
D. Line of code, number of defects found, time of day.

**Answer: B**

Explanation:

An incident report should include additional details such as the status of the incident (e.g., new, open, closed), the degree of impact (e.g., critical, major, minor), and the Test Case Number to provide a complete understanding of the context and severity of the issue.
Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes including the status, impact, and test case number in an incident report for clarity and proper tracking.
Top of Form
Bottom of Form

## Question: 176

Which of the following are 'Exit Criteria?'

A. Acceptance criteria, completion criteria, pass/fail criteria.
B. Coverage of code, schedule, estimates of defect density.
C. The last executable statement within a component.
D. Cost overruns.

**Answer: A**

Explanation:

Exit criteria are the conditions that must be met to conclude a particular phase of testing. These criteria include acceptance criteria, completion criteria, and pass/fail criteria. They help ensure that the testing process is thorough and that the system meets the required standards before moving on to the next phase.
Reference: ISTQB CTFL Syllabus, version 4.0, outlines various exit criteria including acceptance criteria, completion criteria, and pass/fail criteria.

## Question: 177

What is decision table testing?

A. It's a testing design technique based in the internal software structure.
B. It's a static test design technique.
C. It's a testing design technique to verify decisions.
D. It's a testing design technique based in the system requirements.

**Answer: D**

Explanation:

Decision table testing is a black-box test design technique used to test system requirements and to ensure that different combinations of inputs result in appropriate outputs. It is especially useful for testing complex business logic and rules where various combinations of conditions need to be verified.

Reference: ISTQB CTFL Syllabus, version 4.0, explains decision table testing as a technique based on system requirements.

## Question: 178

Of the following, select the best description of the fundamental test process:

A. Planning and control, analysis, design, implementation, execution, evaluation of exit criteria and reporting, test closure.
B. Executing tests, identifying and recording defects and preparing recommendation and closure reports.
C. Controls, preparation of test cases, execution of test cases, execution of regression packs, communicating with developers, writing recommendations for release.
D. Static testing, dynamic testing, defect reporting, reporting and closure.

**Answer: A**

Explanation:

The fundamental test process consists of several sequential steps: planning and control, analysis, design, implementation, execution, evaluation of exit criteria and reporting, and test closure. This comprehensive process ensures that testing is methodical and covers all necessary aspects from preparation to conclusion.

Reference: ISTQB CTFL Syllabus, version 4.0, describes the fundamental test process in detail, outlining each step.

## Question: 179

Which of the following processes ensures that all items of testware are identified, version controlled, tracked for changes, so that traceability can be maintained throughout the test process?

A. Software traceability process
B. Incidence management process
C. Testing design process
D. Configuration management process

**Answer: D**

Explanation:

The configuration management process ensures that all items of testware are identified, version controlled, and tracked for changes. This process helps maintain traceability throughout the test process, ensuring that all test artifacts are managed systematically and changes are monitored effectively.

Reference: ISTQB CTFL Syllabus, version 4.0, details the configuration management process as essential for tracking and version controlling testware.

## Question: 180

What is the value of static code analysis?

A. Detection of failures not easily found by other types of testing
B. Early defect detection
C. Detection of suspicious operations caused by deviations from regulations
D. Detect deviations in standards by executing the source code

**Answer: B**

Explanation:

The value of static code analysis lies in its ability to detect defects early in the development process. By analyzing the code without executing it, static analysis can identify potential issues, such as coding errors or violations of coding standards, which can be addressed before they lead to more significant problems.

Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes the role of static code analysis in early defect detection.

## Question: 181

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:

Use-case testing involves scenarios based on the use cases of a system. In a use-case diagram, an actor represents a type of user (i). A use-case describes interactions between actors and the system (iii). Test cases can be based on use-case scenarios (v), and use-case testing often identifies gaps not found by testing individual components (vi). However, an actor is not always a human user (iv), and use-cases are not the most common test basis for unit testing (ii).

Reference: ISTQB CTFL Syllabus, version 4.0, details use-case testing and its application in testing scenarios .

## Question: 182

It is recommended to perform exhaustive tests for covering all combinations of inputs and preconditions.

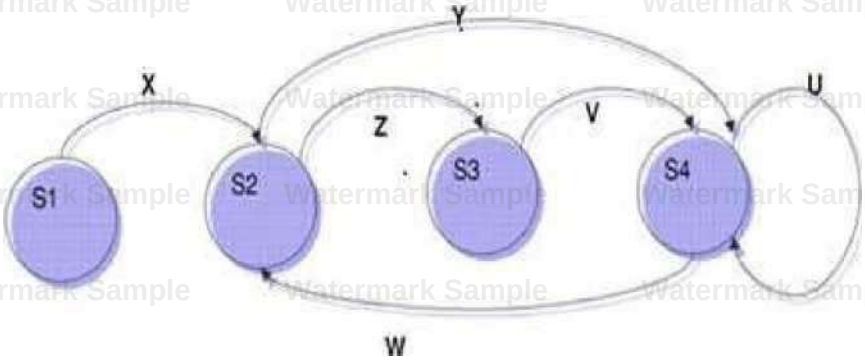
A. Yes, it's strongly recommended.
B. No, risk analysis and priorities should be used to focus testing efforts
C. Yes, and it's also necessary to include all the exit combinations
D. Only the expert testers can make exhaustive tests.

**Answer: B**

Explanation:
It is generally not feasible to perform exhaustive testing due to time and resource constraints. Instead, testing efforts should be focused using risk analysis and priorities to ensure the most critical areas are tested thoroughly. This approach helps in maximizing the effectiveness of the testing process while managing resources efficiently.
Reference: ISTQB CTFL Syllabus, version 4.0, advises against exhaustive testing and recommends focusing testing efforts based on risk and priorities .

## Question: 183

Refer to the exhibit




Given the following State Transition diagram, match the test cases below with the relevant set of state transitions.

(i) X-Z-V-W
(ii) W-Y-U-U

A. (i) =S1– S2 – S3 – S4 – S2 and (ii) = S4 – S2 – S4 – S4 – S4
B. (i) =S1 – S2 – S3 – S4 – S4 and (ii) = S2 –S4 – S4 – S4 – S2
C. (i) =S2– S3 – S4 – S2 – S2 and (ii) = S4 – S2 – S4 – S4 – S4
D. (i) =S2 – S3 – S4 - S4 – S2 and (ii) = S2 –S3 – S4 –S4 – S4

**Answer: A**

Explanation:

In state transition testing, test cases are designed to execute specific sequences of state transitions. Referring to the provided state transition diagram:

For test case (i) X-Z-V-W, the corresponding state transitions are S1 (X) → S2, S2 (Z) → S3, S3 (V) → S4, and S4 (W) → S2. Therefore, the sequence of state transitions is S1 – S2 – S3 – S4 – S2.

For test case (ii) W-Y-U-U, the corresponding state transitions are S4 (W) → S2, S2 (Y) → S4, S4 (U) → S4, and S4 (U) → S4. Therefore, the sequence of state transitions is S4 – S2 – S4 – S4 – S4.

This ensures that the transitions accurately reflect the states and paths described in the test cases. Reference: ISTQB CTFL Syllabus, version 4.0, provides guidelines on state transition testing and mapping test cases to state transitions.

## Question: 184

A system calculates the amount of customs duty to be paid:

_ No duty is paid on goods value up to, and including, $2,000.
_ The next $8,000 is taxed at 10%.
_ The next $20,000 after that is taxed at 12%.
_ Any further amount after that is taxed at 17%.

To the nearest $, which of these groups of numbers fall into three DIFFERENT equivalence classes?

A. $20,000 $20,001 $30,001
B. $2,000 $2,001 $10,000
C. $2,000 $8,000 $20,000
D. $1,500 $2,000 $10,000

**Answer: B**

Explanation:

Equivalence partitioning divides input data into partitions where each partition is expected to be treated the same way by the system. The values $2,000 (no duty), $2,001 (taxed at 10%), and $10,000 (partially taxed at 12%) fall into three different equivalence classes based on the provided tax brackets, ensuring different behaviors for each class.

Reference: ISTQB CTFL Syllabus, version 4.0, describes equivalence partitioning and its application in defining different classes of input data for testing .

## Question: 185

What are metrics NOT used for?

A. To identify the percentage of work done in test environment preparation.
B. To identify the percentage of work done in test case preparation.
C. To apply to the RAD development model.
D. To measure whether dates of test milestones were met.

**Answer: C**

Explanation:

Metrics are used to measure various aspects of the testing process, such as the percentage of work done in test environment preparation, test case preparation, and whether test milestones were met. However, metrics are not specifically used to apply to the RAD (Rapid Application Development) development model. Metrics focus on providing data to evaluate and improve the testing process rather than applying to specific development methodologies.

Reference: ISTQB CTFL Syllabus, version 4.0, discusses the purposes of metrics in the context of the testing process, excluding their application to development models like RAD.

## Question: 186

Which of the following is a white box testing design characteristic?

A. To be based on specifications
B. To be based on an analysis of the test basis documentation
C. To be based on an analysis of the structure of the component or system
D. To include both functional and non-functional testing

**Answer: C**

Explanation:

White box testing design is characterized by its focus on the internal structure of the component or system. It involves testing based on knowledge of the code structure, architecture, and internal design. This approach contrasts with black box testing, which is based on functional specifications without any knowledge of the internal workings of the software.

Reference: ISTQB CTFL Syllabus, version 4.0, defines white box testing as based on the analysis of the internal structure of the component or system.

## Question: 187

In a system designed to work out the employee tax to be paid:

_ An employee has $4,000 of salary tax free.
_ The next $1,500 is taxed at 10%.
_ The next $28,000 after that is taxed at 22%.
_ Any further amount is taxed at 40%.

Which of these is a valid Boundary Value Analysis test case?

A. $28,000
B. $1,500
C. $33,501
D. $5,000

**Answer: D**

Explanation:

Boundary Value Analysis (BVA) involves testing at the boundaries between partitions. For the given tax calculation system, the boundaries include $4,000 (tax-free), $5,500 (next bracket start), and $33,500 (higher bracket start). Testing with $5,000 is valid because it is a boundary value between the tax-free and the 10% tax bracket.
Reference: ISTQB CTFL Syllabus, version 4.0, explains Boundary Value Analysis and its application in testing at partition boundaries.

## Question: 188

What is a test condition?

A. A statement of test objectives and test ideas on how to test.
B. An item or event that could be verified by one or more test cases.
C. The process of identifying differences between the actual results and the expected results for a test.
D. All documents from which the requirements of a component or system can be inferred.

**Answer: B**

Explanation:
A test condition is defined as an item or event that can be verified by one or more test cases. It can include functions, transactions, structural elements, or quality attributes that need to be tested to ensure the system behaves as expected.
Reference: ISTQB CTFL Syllabus, version 4.0, provides the definition of a test condition in the context of items or events that are verifiable by test cases.

## Question: 189

Which of the following test organizations has the highest level of independence?

A. Independent testers within the development teams
B. Independent testers from the user community
C. Independent test specialists for specific test types, such as usability, performance or certification test specialists
D. Code tested by another developer from the development team

**Answer: C**

Explanation:

The highest level of independence in testing is achieved by having independent test specialists who focus on specific test types such as usability, performance, or certification testing. This independence helps in unbiased defect detection and evaluation, minimizing cognitive biases that might affect testers who are closer

to the development process.

Reference: ISTQB CTFL Syllabus, version 4.0, discusses the levels of independence in testing and emphasizes the role of independent test specialists for achieving high levels of objectivity.

## Question: 190

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:

Testing support tools are typically used to automate activities that are resource-intensive when done manually, repetitive tasks, and repetitive inspections. However, they are not usually designed to automate activities that cannot be executed manually at all, as such activities may require complex and sophisticated human intervention or decision-making.

Reference: ISTQB CTFL Syllabus, version 4.0, provides an overview of the typical targets of testing support tools, which exclude activities that cannot be executed manually.

## Question: 191

What type of test design technique is the most effective in testing screen-dialog flows?

A. Use case testing
B. Boundary value testing
C. Statement testing and coverage
D. State transition testing

**Answer: D**

Explanation:

State transition testing is the most effective test design technique for testing screen-dialog flows. This technique is used to model the different states of a system and the transitions between those states, which is ideal for verifying that the system behaves correctly as the user navigates through different screens and dialogs.

Reference: ISTQB CTFL Syllabus, version 4.0, discusses the application of state transition testing in scenarios involving screen-dialog flows.

## Question: 192

What content would be in an incident report if that incident report was based on the IEEE 829 Standard for SoftwareTest Documentation?

(i) Identification of configuration items of the software or system.
(ii) Software or system lifecycle process in which the incident was observed.
(iii) Description of the anomaly to enable reproduction of the incident.
(iv) Number of occurrences of the incident.
(v) Classification of the cause of the incident for metrics and for reporting purposes.

Number of correct answers: 1

A. i, ii, iii
B. ii, iii
C. i, iii, iv
D. i, ii, iii, v

**Answer: D**

Explanation:

According to the IEEE 829 Standard for Software Test Documentation, an incident report should include the identification of configuration items of the software or system (i), the software or system lifecycle process in which the incident was observed (ii), a description of the anomaly to enable reproduction of the incident (iii), and the classification of the cause of the incident for metrics and reporting purposes (v). The number of occurrences of the incident is not typically included. Reference: ISTQB CTFL Syllabus, version 4.0, aligns with the IEEE 829 Standard for Software Test Documentation regarding the content of an incident report.

## Question: 193

"Experience based" test design techniques, typically...

A. Use decision tables to generate the Boolean test conditions to be executed.
B. Identify the structure of the system or software at the component, integration or system level.
C. Use the skill, intuition and experience of the tester to derive the test cases, using error guessing and exploratory testing.
D. Establish traceability from test conditions back to the specifications and requirements.

**Answer: C**

Explanation:

Experience-based test design techniques rely on the tester's skill, intuition, and experience to derive test cases. Techniques like error guessing and exploratory testing fall under this category. These methods do not depend on formal specifications but rather on the tester's understanding of the system and potential areas where defects might exist.
Reference: ISTQB CTFL Syllabus, version 4.0, elaborates on experience-based testing techniques and

their reliance on tester expertise.

## Question: 194

Testers are often seen as the bearer of unwanted news regarding defects. What are effective ways to improve the communication and relationship between testers and others?

a) Communicate factual information in a constructive way.
b) Try to understand how the other person feels and why they react the way they do.
c) Always outsource testing activities.
d) Never record information that could be used to apportion blame to an individual or team.

A. a and b
B. a, b and c
C. a, b and d
D. a and c

**Answer: A**

Explanation:
Effective ways to improve communication and relationships between testers and other team members include communicating factual information in a constructive way (a) and trying to understand how the other person feels and why they react the way they do (b). These approaches foster a collaborative and empathetic environment, reducing friction and enhancing teamwork. Reference: ISTQB CTFL Syllabus, version 4.0, discusses strategies for improving communication and relationships within teams.

## Question: 195

Which of the following is a valid reason for writing test cases based on experience and intuition? [K1]

A. Use of formal techniques requires expensive training
B. Only experience can ensure all functionality is covered
C. Tests based on experience and intuition can supplement formal techniques
D. Formal techniques require the use of expensive tools

**Answer: C**

Explanation:

Experience-based and intuition-based testing techniques, such as error guessing and exploratory testing, can be valuable in uncovering defects that formal techniques might miss. These approaches leverage the tester's knowledge, intuition, and experience, providing a complementary layer to more structured and formal testing techniques.
Reference: ISTQB CTFL Syllabus, version 4.0, discusses the role and benefits of experience-based testing techniques in supplementing formal techniques.

## Question: 196

Which of the following test design techniques is classified as a structure-based (white box) technique? [K1]

A. Exploratory testing
B. Decision table testing
C. State transition testing
D. Statement testing

**Answer: D**

Explanation:

Statement testing is a structure-based (white box) testing technique. It involves executing all the executable statements in the code at least once. This technique requires knowledge of the internal structure of the code, making it a white box testing method.
Reference: ISTQB CTFL Syllabus, version 4.0, identifies statement testing as a structure-based technique.

## Question: 197

Why is independent testing important? [K1]

A. Because independent testers make fewer assumptions than developers
B. Because independent testers are isolated from the development team
C. Because independent testers can verify assumptions made during specification and implementation of the system
D. Because independent testers have a greater sense of responsibility for quality than developers

**Answer: C**

Explanation:
Independent testing is crucial because independent testers can provide an unbiased perspective and verify the assumptions made during the specification and implementation phases. This helps in identifying defects that might be overlooked by developers who are closely involved in the creation of the system.
Reference: ISTQB CTFL Syllabus, version 4.0, highlights the importance of independent testing in verifying assumptions and providing an unbiased evaluation.

## Question: 198

Which statement BEST describes when test planning should be performed? [K1]

A. Test planning is performed only once, at the beginning of the life cycle, andgenerates a Master Test Plan
B. Test planning is performed at the beginning of the life cycle and again at the beginning of test execution
C. Test planning is performed at the beginning of the life cycle and again at every test level
D. Test planning is performed continuously in all life cycle processes and activities

**Answer: D**

Explanation:

Test planning should be a continuous activity throughout the software development lifecycle. This ensures that the test plans remain relevant and effective, adapting to any changes in the project scope, requirements, or timelines.

Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes the need for continuous test planning throughout the lifecycle.

## Question: 199

Refer to the exhibit

The following test cases need to be run, but time is limited, and it is possible that not all will be completed before the end of the test window

| # | Description | Priority | Note |
|---|---|---|---|
| a | Re-test defect no 52 | Low | Re-test |
| b | Ability to amend transaction type | High | |
| c | Re-test defect no 26 | High | Re-test |
| d | Run regression test script | Medium | Regression |
| e | Print monthly sales figures | Medium | |
| f | Add special invoice to previous month | Low | |
| g | Reprint selected previous sales figures | High | Must be run after item e |
| h | Account administrator able to amend any previous month's sales figures | Low | |
| i | Print year-to-date figures | Medium | |

The first activity is to run any re-tests, followed by the regression test script. Users have supplied their priority order to tests.

Which of the following gives an appropriate test execution schedule, taking account of the prioritisation and other constraints? [K3]

A. b, c, g,d,e,i, a, f, h
B. a, c, d,b,g,e, i, f, h
C. c, a, d,b,e,g, i, h, f
D. d, c, a,e,b,g, i, f, h

**Answer: B**

Explanation:

The appropriate test execution schedule considering the prioritization and constraints should start with re-tests, followed by the regression test script. After these, tests should be run based on their priority levels:
a (Re-test defect no 52, Low priority, Re-test)
c (Re-test defect no 26, High priority, Re-test)

d (Run regression test script, Medium priority, Regression)
b (Ability to amend transaction type, High priority)
g (Reprint selected previous sales figures, High priority, Must be run after item e)
e (Print monthly sales figures, Medium priority)
i (Print year-to-date figures, Medium priority)
f (Add special invoice to previous month, Low priority)
h (Account administrator able to amend any previous month's sales figures, Low priority)
This sequence ensures that re-tests and high-priority tasks are completed first, followed by mediumpriority tasks, and then low-priority tasks.
Reference: The provided prioritization and constraints are aligned with standard test execution scheduling practices as discussed in the ISTQB CTFL Syllabus, version 4.0.

## Question: 200

Which of the following factors will MOST affect the testing effort required to test a software product? [K1]

A. The number of staff available to execute tests
B. The level of detail in the test plan
C. The requirements for reliability and security in the product
D. The test estimation method used

**Answer: C**

Explanation:
The requirements for reliability and security in a product significantly impact the testing effort. High requirements for reliability and security necessitate more extensive and thorough testing to ensure that the product meets these stringent criteria. This often includes additional test cases, more complex test environments, and longer test execution times.
Reference: ISTQB CTFL Syllabus, version 4.0, highlights that requirements for reliability and security are critical factors affecting the testing effort.

## Question: 201

Which of the following metrics could be used to monitor progress along with test preparation and execution? [K1]

A. The total number of tests planned
B. The total number of requirements to be tested
C. The failure rate in testing already completed
D. The number of testers used for test execution so far

**Answer: C**

Explanation:

The failure rate in testing already completed is a useful metric to monitor progress along with test preparation and execution. It provides insight into the effectiveness of the tests and the quality of the product, helping to identify

areas that may require additional attention or resources.

Reference: ISTQB CTFL Syllabus, version 4.0, discusses the use of various metrics, including failure rates, to monitor testing progress and quality.

## Question: 202

Test objectives for systems testing of a safety critical system include completion of all outstanding defect correction. Regression testing is required following defect correction at all test levels. Which TWO of the following metrics would be MOST suitable for determining whether the test objective has been met? [K2]

a. Regression tests run and passed in systems testing
b. Incidents closed in systems testing
c. Planned tests run and passed in system testing
d. Planned tests run and passed at all levels of testing
e. Incidents raised and closed at all levels of testing

A. a and e
B. b and c
C. d and e
D. a and b

**Answer: A**

Explanation:

For a safety-critical system, it is essential to ensure that regression tests are run and passed following defect corrections at all test levels. Additionally, tracking incidents raised and closed at all levels of testing helps ensure that all defects are addressed and the system meets safety standards. These metrics provide a comprehensive view of both the completeness of testing and the effectiveness of defect resolution.

Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes the importance of regression testing and incident tracking for safety-critical systems.

## Question: 203

Test script TransVal 3.1 tests transaction validation via screen TRN 003B. According to the specification (PID ver 1.3 10b iv) the validation screen should not accept future dated transactions.

Test script TransVal 3.1 passes. Test script eod 1.4 tests end of day processing and is run after the execution of TransVal 3.1 using data entered during that test

Which of the following is the BEST detail on an incident report? [K3]

A. Title. End of Day failure.. Reproducible. Yes. Description. Script eod 1.4 fails when the first transaction of the day is a future dated transaction. Screen shot of the failure attached.
B. Title. Transaction input screen validation..Reproducible. Yes. Description. Script eod 1..4 fails . Screen shot of the failure attached. Validation of transaction entryon screen TRN-003B should not allow future dated transactions – see PID ver 1.3 para 10b iv.
C. Title. Screen TRN-003B validation of transaction date.. Reproducible. No. Description. When a future dated transaction is processed by the end of day process, a failure can occur. This does not always happen. Screen

shot of the failure attached.

D. Title. Screen TRN-003B validation of transaction date. Reproducible. Yes. Description. Script eod 1.4 fails when the first transaction of the day is a future dated transaction. Screen shot of the failure attached. Validation of transaction entry on screen TRN-003B should not allow future dated transactions – see PID ver 1.3 para 10b iv.

**Answer: D**

Explanation:

An effective incident report should provide a clear and detailed description of the issue, including the title, reproducibility, a detailed description of the failure, and any relevant references to specifications. This helps in accurately identifying, reproducing, and resolving the defect. In this case, including the specific reference to the PID and a screenshot helps provide all necessary information for resolving the issue.

Reference: ISTQB CTFL Syllabus, version 4.0, outlines the components of a well-documented incident report according to the IEEE 829 Standard for Software Test Documentation.

## Question: 204

Which TWO of the following test tools would be classified as test execution tools? [K2]

a. Test data preparation tools
b. Test harness
c. Review tools
d. Test comparators
e. Configuration management tools

A. a and b
B. c and d
C. c and e
D. b and d

**Answer: D**

Explanation:

Test execution tools include tools that directly assist in the execution of tests and comparison of actual outcomes with predicted outcomes. A test harness (b) is used to execute tests by providing the necessary context and data inputs for running a test. Test comparators (d) are used to compare the actual results produced by the system under test with the expected results, which is a key part of test execution.

Reference: ISTQB CTFL Syllabus, version 4.0, classifies test harnesses and test comparators as test execution tools.

## Question: 205

Which of the following is a consideration when deploying test execution tools? [K1]

A. Data-driven testing cannot be used with test execution tools
B. Recorded manual tests may become unstable in use

C. Keyword-driven testing cannot be used with test execution tools
D. Expected results for tests are not required because the tool generates expected results

**Answer: B**

Explanation:
One of the considerations when deploying test execution tools is that recorded manual tests may become unstable. This instability can be due to changes in the user interface, environment, or other dependencies that the recorded script relies on. As a result, maintaining these recorded tests can be challenging and may require frequent updates.
Reference: ISTQB CTFL Syllabus, version 4.0, discusses the potential instability of recorded manual tests when using test execution tools.

## Question: 206

Which of the following correctly states a limitation in the use of static analysis tools? [K1]

A. Static analysis tools can be applied to new code but cannot be applied to existing code
B. Static analysis tools can be used to enforce coding standards
C. Static analysis tools always generate large numbers of warning messages when applied to new code, even if the code meets coding standards
D. Static analysis tools do not generate warning messages when applied to existing code

**Answer: C**

Explanation:

One limitation of static analysis tools is that they tend to generate a large number of warning messages when applied to new code. This can occur even if the code meets established coding standards, as the tools are designed to identify a wide range of potential issues, leading to a high volume of output that needs to be reviewed and addressed.

Reference: ISTQB CTFL Syllabus, version 4.0, outlines the limitations of static analysis tools, including the generation of numerous warning messages.

## Question: 207

What is the main reason for using a pilot project to introduce a testing tool into an organization? [K1]

A. To identify the requirements for using a tool
B. To make a selection between alternative tools
C. To assess whether the tool will be cost- effective
D. To ensure the tools fits existing processes without change

**Answer: C**

Explanation:

The main reason for using a pilot project to introduce a testing tool into an organization is to assess whether the tool will be cost-effective. A pilot project allows the organization to evaluate the tool's benefits, identify any potential issues, and determine whether the investment in the tool is justified based on its performance and the value it adds.

Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes the importance of using a pilot project to evaluate the cost-effectiveness of a new testing tool.

## Question: 208

Which of the following BEST describes the relationship between test planning and test execution? [K2]

A. Test planning ensures the level of detail in test procedures is appropriate for test execution
B. Test planning schedules test execution but does not assign resources
C. Test planning defines the overall approach to testing but does not schedule specific activities such as test execution
D. Test planning identifies test objectives related to scope and risk but does not define the level of detail for test procedures used in test execution

**Answer: A**

Explanation:

Test planning involves defining the overall approach to testing, including setting objectives, strategies, and resources. It ensures that the level of detail in test procedures is appropriate for the subsequent test execution phase. Effective test planning helps in aligning the testing activities with the project goals and ensures that the tests are executed as per the defined procedures. Reference: ISTQB CTFL Syllabus, version 4.0, outlines the role of test planning in ensuring the appropriate level of detail for test procedures and scheduling test execution.

**Question: 209**

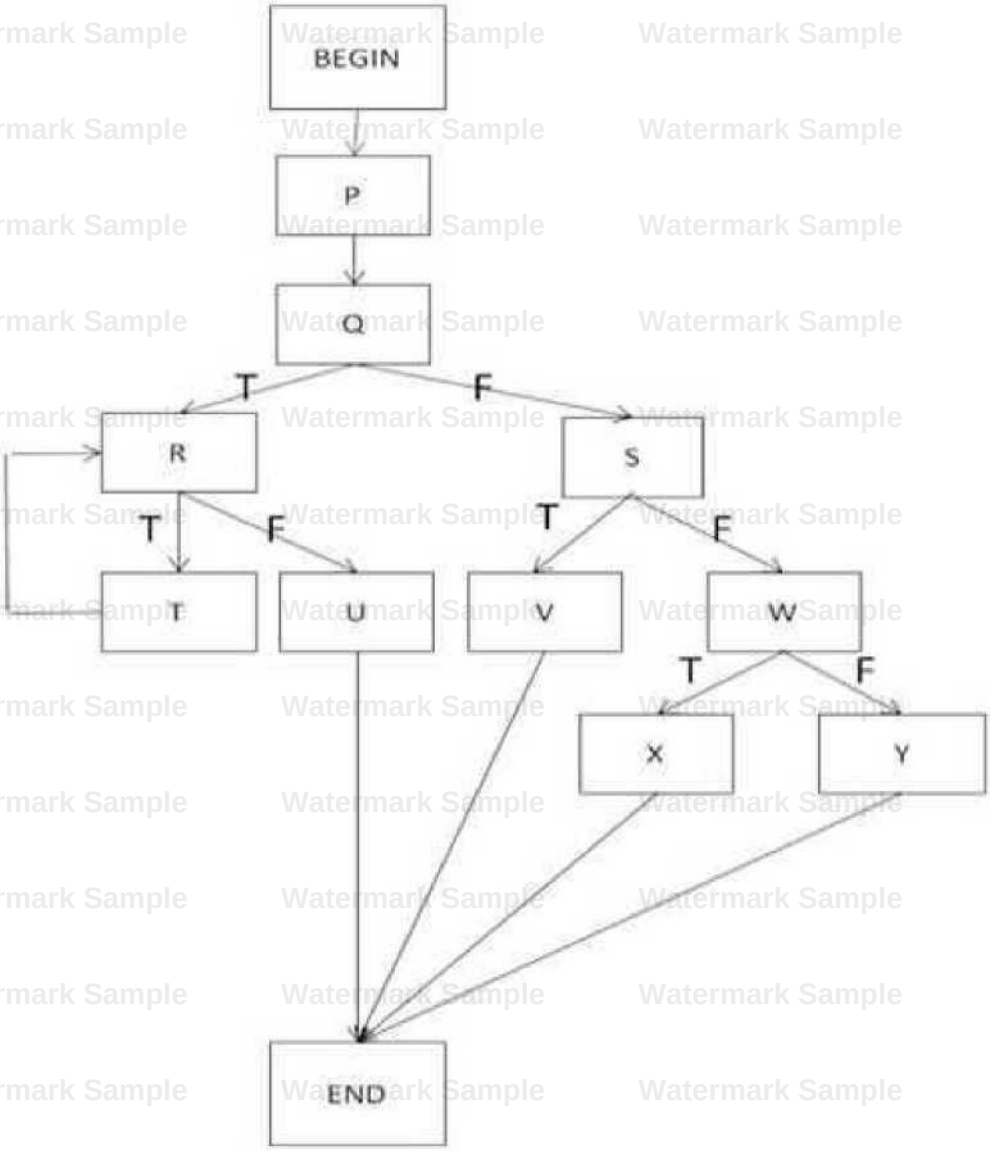
The flow graph below shows the logic of a program for which 100% statement coverage and 100% decision coverage is required on exit from component testing. [K4]

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: B**

Explanation:

Based on the given flow graph and the test cases provided:

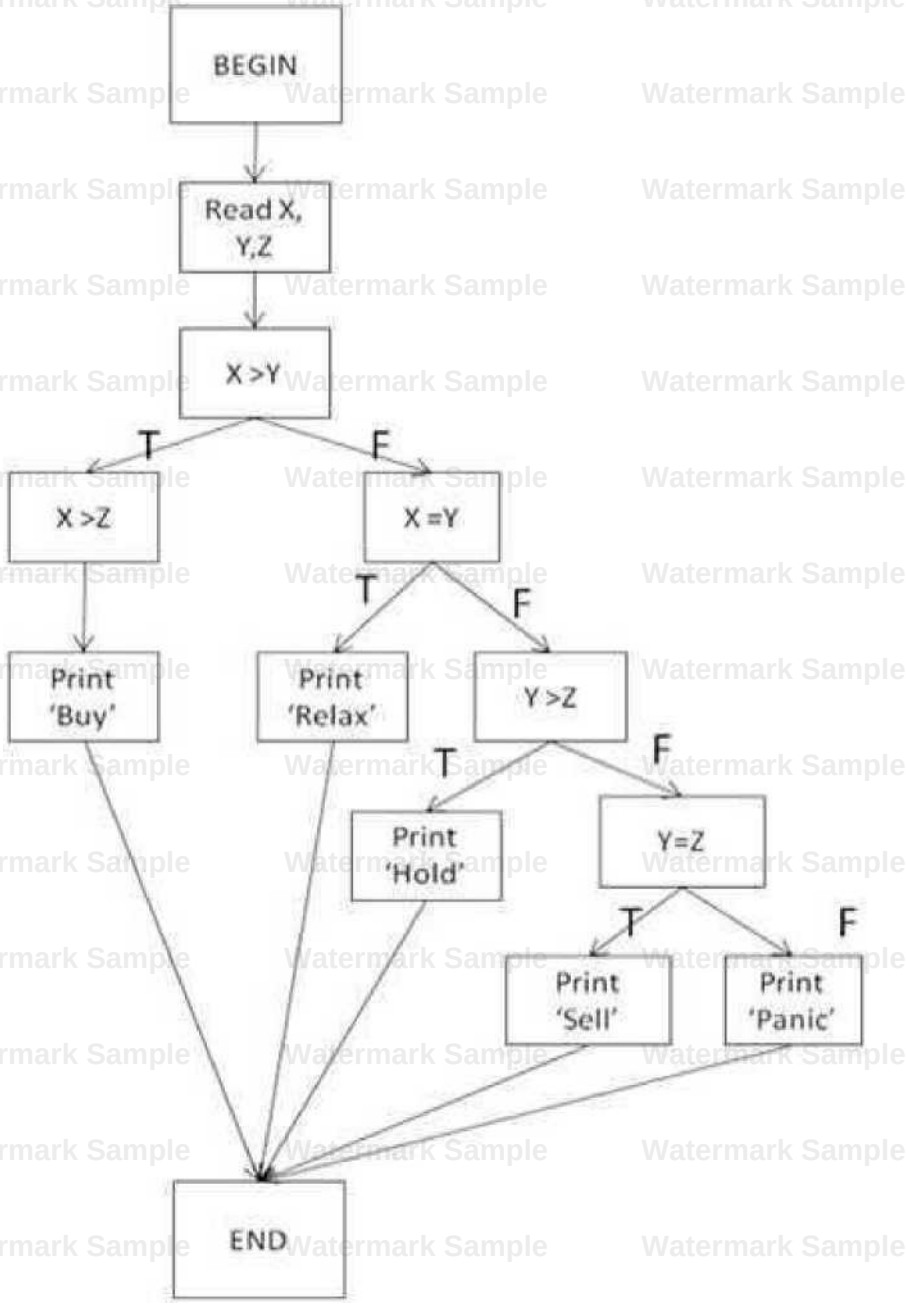
Test Case 1 covers the path P → Q → R→ U.

Test Case 2 covers thepath P → Q → S → V.
Test Case 3 covers thepath P → Q → S→ W → X.
Test Case 4 covers thepath P → Q → S→ W → Y.
All decision points (P, Q, R, S, W) are covered at least once, ensuring 100% decision coverage.
However, not all statements (T) are covered, as there is no test case that covers the path leading to statement T.
Reference: ISTQB CTFL Syllabus, version 4.0, provides guidelines on achieving statement and decision coverage, emphasizing the need for covering all statements and decision points.

## Question: 210

Which of the following test cases will ensure that the statement 'Print 'Hold'' 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: C**

Explanation:

To ensure that the statement 'Print 'Hold'' is exercised:

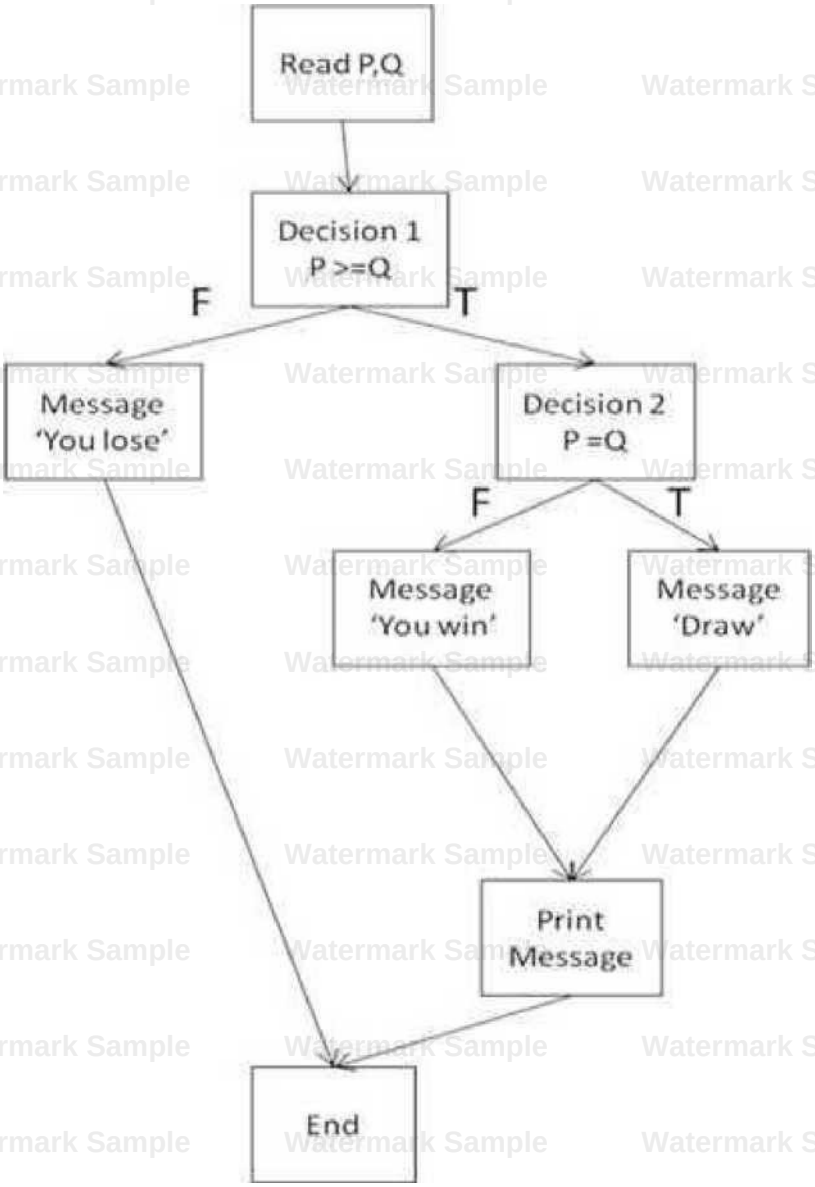
X=2, Y=4, Z=3 This test case will follow the path where X ≤ Y (False), Y > Z (True), leading to the 'Print 'Hold'' statement.

Reference: ISTQB CTFL Syllabus, version 4.0, discusses the importance of defining test cases that cover specific paths to ensure all relevant statements are exercised.

## Question: 211

Which of the test cases below will exercise both outcomes from decision 2? [K3]

Refer to the exhibit




A. P = 24, Q = 20, P=24, Q=25 B. P = 36, Q = 36, P=37, Q=35 C. P = 42, Q = 43, P=42, Q=42 D. P = 37, Q = 36, P=35, Q=36

**Answer: B**

Explanation:

## Question: 212

The Cambrian Pullman Express has special ticketing requirements represented by the partial decision table below.

Refer to the exhibit

| Conditions | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 |
|---|---|---|---|---|---|---|
| First Class ticket | Y | N | N | N | Y | N |
| Std Class Flexible ticket | N | Y | N | N | N | Y |
| Std Class Day Return | N | N | Y | N | N | N |
| Std Class Super Saver | N | N | N | Y | N | N |
| Railcard holder | N | N | N | N | Y | Y |
| **Actions** | | | | | | |
| OK to travel | Y | N | N | N | Y | N |
| Eligible for upgrade | N | Y | N | N | N | Y |
| Concessionary fare | N | N | N | N | Y | Y |

Carol has a student railcard and is travelling on a Flexible Standard Class ticket. James has a senior railcard and is travelling on a super saver ticket. Which of the options represents the correct actions for these two test cases? [K3]

A. Carol is eligible to upgrade; James cannot use the service
B. Carol is OK to travel; James is eligible for an upgrade
C. Carol and James are both eligible to upgrade
D. Carol is OK to travel; James cannot use the service

**Answer: A**

Explanation:
From the decision table:
Carol has a student railcard and a Flexible Standard Class ticket, which means she is eligible to upgrade.
James has a senior railcard and a Super Saver ticket, which means he cannot use the service as per the conditions specified in the table.
Reference: ISTQB CTFL Syllabus, version 4.0, explains the use of decision tables to determine the correct actions based on different conditions.

## Question: 213

How does software testing contribute to the quality of delivered software? [K2]

A. By detecting and removing all the defects in the delivered code and ensuring that all tests adhere to the quality standards set for the project
B. By measuring reliability of the software and ensuring that it is always above 99.99%
C. By identifying root causes of defects from past projects and using the lessons learned to improve processes

and thus help to reduce the defect count

D. By detecting all deviations from coding good practice and ensuring that these are corrected

**Answer: C**

Explanation:

Software testing contributes to the quality of delivered software by identifying root causes of defects from past projects. This allows the team to learn from previous mistakes and improve their processes, ultimately reducing the defect count in future projects.

Reference: ISTQB CTFL Syllabus, version 4.0, highlights the importance of continuous improvement and learning from past defects to enhance software quality.

## Question: 214

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 due to a defect in the purchase identification module. This defect could have been caused by a programmer mistake or an error in the specification. The incorrect counting of purchase attempts indicates a flaw in the logic or implementation of the module responsible for identifying successful purchases.

Reference: ISTQB CTFL Syllabus, version 4.0, discusses the causes of defects and the importance of identifying whether they stem from programming mistakes or specification errors.

Top of Form

Bottom of Form

## Question: 215

Which of the following is a valid objective of testing? [K1]

A. Correcting defects

B. Locating defects in the code
C. Preventing defects
D. Ensuring no defects are present

**Answer: C**

Explanation:
A valid objective of testing is to prevent defects. While correcting defects and locating defects in the code are part of the testing process, the primary goal is often to prevent defects by identifying issues early in the development cycle and improving the overall quality of the software. Ensuring no defects are present is an unrealistic objective, as testing can demonstrate the presence of defects but not their absence.

Reference: ISTQB CTFL Syllabus, version 4.0, emphasizes the objectives of testing, including defect prevention as a key goal.

## Question: 216

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:

For user acceptance testing (UAT) of the first release of a new software product aimed at a general market and built using Agile methods, appropriate test objectives would be to ensure the product works as expected (c) and to assess the overall quality of the product (d). These objectives focus on validating that the software meets user needs and overall quality standards.

Reference: ISTQB CTFL Syllabus, version 4.0, outlines suitable objectives for UAT, particularly in an Agile context.

## Question: 217

Which statement correctly describes debugging? [K2]

A. Testers identify defects, developers locate and correct defects, testers confirm the correction has cleared the original defect
B. Developers identify defects, testers locate defects, developers correct and confirm the correction has cleared the original defect

C. Testers identify and locate defects, developers correct defects and confirm the correction has cleared the original defect

D. Developers identify, locate and correct defects, testers confirm the correction has cleared the original defect

**Answer: D**

Explanation:

Debugging is the process where developers identify, locate, and correct defects. Testers are then responsible for confirming that the correction has resolved the original defect. This separation of responsibilities ensures that defects are properly fixed and verified.

Reference: ISTQB CTFL Syllabus, version 4.0, explains the roles of developers and testers in the debugging process.

## Question: 218

Which of the main activities of the fundamental test process does the task 'verify the test environment set up is correct' relate to? [K1]

A. Planning and control
B. Analysis and design
C. Implementation and execution
D. Evaluating exit criteria and reporting

**Answer: C**

Explanation:

The task "verify the test environment set up is correct" relates to the Implementation and Execution phase of the fundamental test process. This phase includes setting up the test environment, creating test data, and executing test cases.

Reference: ISTQB CTFL Syllabus, version 4.0, describes the main activities of the fundamental test process, including verifying the test environment during the implementation and execution phase.

## Question: 219

Which of the following characteristics is most likely to promote effective software testing? [K1]

A. Independence from the production process
B. A belief that programmers always make mistakes
C. Knowledge of the number of defects typically found in a program
D. Confidence that the next stage will find defects missed at this stage

**Answer: A**

Explanation:

Independence in software testing refers to the separation of testing activities from the development and production processes. This separation helps ensure objectivity and reduces the risk of bias in the testing results. Independent testers are more likely to identify defects that developers might overlook due to familiarity with the code. The ISTQB syllabus highlights the importance of independence in testing as a key factor in promoting effective and unbiased testing practices.

## Question: 220

Which of the following statements about software development models is most accurate? [K1]

A. The 4 stage V model is always the best choice of software development model for any project
B. The agile development model is usually most appropriate for short projects
C. The choice of software development model depends on product and project characteristics
D. The 2 stage V model is the most appropriate development model for simple products

**Answer: C**

Explanation:

The selection of a software development model should be based on the specific characteristics of the product and project. Factors such as project size, complexity, requirements stability, and team expertise influence the choice of the most suitable development model. The ISTQB syllabus explains that there is no one-size-fits-all approach, and the decision should be tailored to the context and needs of the project.

## Question: 221

When should testers be involved in reviewing a UAT specification? [K1]

A. At the beginning of the project
B. As soon as requirements have been approved
C. As soon as the UAT specification has been drafted
D. At any time before UAT begins

**Answer: C**

Explanation:

Testers should be involved in the review of the User Acceptance Testing (UAT) specification as soon as it is drafted. Early involvement of testers ensures that the UAT criteria are clear, complete, and testable. This approach aligns with the shift-left testing strategy, which advocates for early testing

activities to identify defects and issues as soon as possible. The ISTQB syllabus emphasizes the importance of early and continuous testing throughout the software development lifecycle.

## Question: 222

Which of the following accurately defines the integration testing test level? [K2]

A. Test basis includes software and system design, test objects include interfaces, and tests concentrate on the interactions between different parts of a system

B. Test basis includes component requirements, test objects include database modules, and tests concentrate on the behaviour of the system as a whole.

C. Test basis includes business processes, test objects include system configuration and configuration data, and tests concentrate on establishing confidence in the system

D. Test basis includes use cases, test objects include user procedures and tests concentrate on a high level model of system behaviour

**Answer: A**

Explanation:

Integration testing focuses on the interactions between integrated components or systems. It aims to identify defects in the interfaces and interaction points between different parts of the system.
According to the ISTQB syllabus, the test basis for integration testing typically includes software and system design documents, while the test objects are the interfaces and interactions between components. The goal is to ensure that the integrated parts function together as intended.

## Question: 223

A bank is developing a new service that will be delivered via the web. The user interactions are defined as a set of use cases and the service is designed to be available continuously 24/7. In view of the challenging characteristics of the service the test manager has decided that the code should be thoroughly tested at component level.

Which of the following test types will be required during the development? [K2]

A. Functional testing to test security at the system level, load testing at the system level to ensure the system availability is acceptable, regression testing at all levels, structure based testing at the component level only

B. Functional testing to test the use cases at component level, reliability testing to test availability at the integration level, regression testing at the system testing level only, structure based testing at all levels

C. Functional testing of the use cases at system level, load testing at component level to ensure availability is acceptable, regression testing at the system and acceptance levels only, and structure based testing at the integration level only

D. Functional testing to test security at the acceptance level, load testing at the acceptance test level to ensure availability is acceptable, regression testing at the acceptance level only to ensure late changes are made correctly, and structure based testing at the component level

**Answer: A**

Explanation:
For a web-based service designed to be available 24/7, a comprehensive testing approach is necessary to ensure

functionality, security, reliability, and performance. Functional testing at the system level will verify the security aspects. Load testing at the system level will ensure the system can handle expected loads and remain available. Regression testing across all levels will confirm that changes do not introduce new defects. Structure-based testing at the component level will ensure the internal structure of components is robust. The ISTQB syllabus outlines these testing types and their importance in different contexts and levels.

Top of Form

Bottom of Form

## Question: 224

Functional and structural tests are alternative test types that may be used separately or together at which test level? [K1]

A. At the component test level only
B. At all test levels
C. At integration testing and system testing levels only
D. At all levels from integration testing to acceptance testing

**Answer: B**

Explanation:

Functional and structural tests are types of testing techniques that can be applied at various levels of testing. According to the ISTQB CTFL syllabus, functional and structural tests are not restricted to any specific test level but are applicable across all levels of testing. This includes component, integration, system, and acceptance testing. The use of both types of tests helps ensure comprehensive testing coverage, addressing both the functional requirements and the internal structure of the software product.

## Question: 225

Which of the following statements best characterises maintenance testing? [K2]

A. Maintenance testing is triggered by changes to delivered software and uses impact analysis to minimise the amount of regression testing needed
B. Maintenance testing is triggered by changes to software under development before initial delivery and uses the test plan to determine how much regression testing to do
C. Maintenance testing is triggered by changes to the test environment and uses testing tools to perform regression testing
D. Maintenance testing is triggered by changes to the software environment and uses structural testing to ensure the changes function correctly

**Answer: A**

Explanation:

Maintenance testing is conducted on the software after it has been delivered to address any issues or changes required due to defects, environment changes, or new requirements. The ISTQB syllabus explains that

maintenance testing typically involves impact analysis to determine the extent of regression testing required, thus optimizing the testing effort and focusing on affected areas.

## Question: 226

Under which of the following circumstances is maintenance testing required? [K1]

A. Migration of software onto a new platform
B. Testing during initial development of a replacement for an existing system
C. Purchase of a new software tool
D. Updating of a regression suite

**Answer: A**

Explanation:

Maintenance testing is essential when there is a need to ensure that the software continues to function correctly after it has been modified. Migration to a new platform is a significant change that can impact the software's functionality, requiring thorough maintenance testing to verify that the software performs as expected in the new environment. This ensures that any platform-specific issues are identified and resolved.

## Question: 227

Which of the following BEST defines static techniques? [K1]

A. Executing the software work product
B. Manually examining the code or project documentation
C. Automated analysis of the code or project documentation
D. Manual examination and automated analysis of code or project documentation

**Answer: D**

Explanation:
Static techniques involve the examination and evaluation of software products without executing them. According to the ISTQB syllabus, these techniques include both manual and automated reviews and analyses of code and project documentation. This helps identify defects early in the development process, improving the overall quality and reducing the cost of fixing defects later in the lifecycle.

## Question: 228

Which of the following is a role of a formal review? [K1]

A. Adjudicator
B. Moderator
C. Governor
D. Corrector

**Answer: B**

Explanation:
In formal reviews, such as inspections, the role of the moderator is crucial. The moderator is responsible for leading the review process, ensuring that the review is conducted according to the defined procedures, and facilitating the review meeting. The ISTQB syllabus highlights that the moderator's role is to guide the review team, manage any issues that arise during the review, and ensure that the objectives of the review are met.

## Question: 229

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.

**Answer: C**

Explanation:
For a review process to be successful, several factors must be in place. According to the ISTQB syllabus, successful reviews generally have clearly defined objectives (a), involve the right people who understand the objectives of the review (c), and place an emphasis on learning and process improvement (d). These elements ensure that the review is focused, effective, and contributes to the overall improvement of the software development process.

## Question: 230

Which of the following would typically be identified using static analysis by tools? [K1]

A. Spelling mistake on an error message
B. A potential infinite loop
C. Memory leakage
D. A variable set to the wrong value

**Answer: B**

Explanation:
Static analysis tools are designed to examine the code without executing it. They are particularly effective at identifying structural issues in the code, such as potential infinite loops (B), which can occur due to incorrect logic that would cause the code to execute indefinitely. These tools analyze the code's structure and logic flow to find

such issues early in the development process, helping to improve the code's quality before it is run.

## Question: 231

Before an invoice can be created, an account is required. Before an account can be set up, an account user is required (in order to set up the account). The software is delivered with a master user only, who can only create other types of users. The following test cases have been written to test the high- level structure of the software

a. Create an invoice
b. Amend an invoice
c. Process an invoice (send to customer)
d. Delete an invoice
e. Create an account
f. Create an account user
g. Amend an account user
h. Delete an account user
i. Amend an account
j. Delete an account

Which of the following test procedures would enable all tests to be run? [K3]

A. f, g, a, c, b, d, e, i, j, h
B. e, i, a, c, b, d, f, g, h, j
C. e, i, f, g, a, c, b, d, h, j
D. f, g, e, i, a, b, c, d, j, h

**Answer: C**

Explanation:
To run all the tests successfully, one must follow the logical sequence dictated by the dependencies between the actions. The process should start with creating the account (e) and setting it up (i), then creating the account user (f) and making any amendments (g). Only after these steps can an invoice be created (a) and processed (c), followed by amending (b) and deleting (d) the invoice. Finally, the
sequence concludes with managing the account user and the account itself (h, j). This logical order ensures all dependencies are respected and all functionalities are tested.

## Question: 232

Which of the following test case design techniques is white box (structure-based)? [K1]

A. Use case testing
B. State transition testing
C. Decision testing
D. Equivalence partitioning

**Answer: C**

Explanation:

White box testing techniques, also known as structure-based techniques, focus on the internal structure of the software. Decision testing (C) is a white box technique that involves creating test cases to execute specific decision points in the code, such as conditional statements. This ensures that all possible paths through the code are tested, helping to identify defects related to the control flow of the application. Other techniques listed, such as use case testing and state transition testing, are black box techniques that focus on the software's functionality from the user's perspective.

## Question: 233

From the following list, which of the following apply to experience-based techniques? [K2]

a. Test cases are derived from a model of the problem to be solved or the software
b. Test cases are derived from the knowledge of the testers
c. The knowledge of testers, developers and users is used to drive testing
d. The internal structure of the code is used to derive test cases

A. a and b.
B. c and d.
C. a and d.
D. b and c.

**Answer: D**

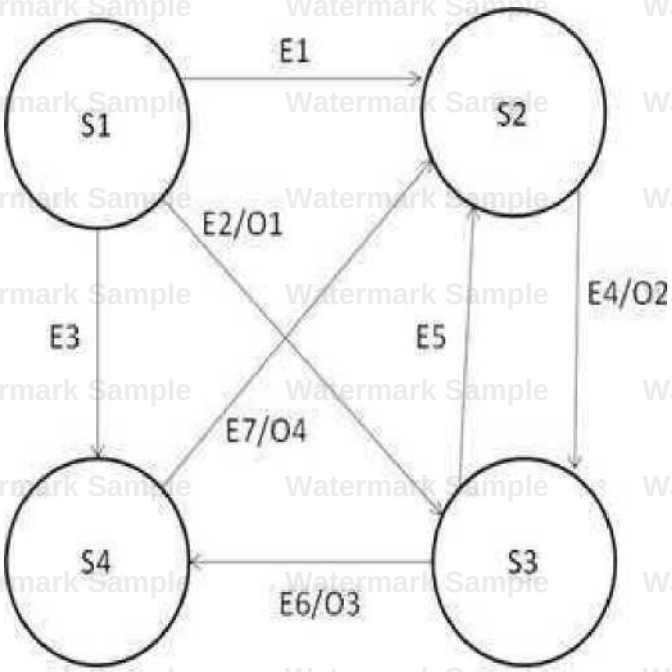
Explanation:

Experience-based techniques rely on the testers' knowledge, skills, and experience, as well as insights from developers and users, to design test cases. These techniques do not depend on formal models or the internal structure of the software but rather on practical experience and understanding of how the software is likely to be used and where defects are likely to be found.

## Question: 234

A test case starts at S1 and triggers 4 events in sequence: E1, E4, E5, E7. What will be the finishing state and the output(s) from the test case? [K3]

Refer to the exhibit




A. S2 and O4
B. S4 and O2
C. S4 and O4
D. S2 and O2

**Answer: C**

Explanation:

According to the state transition table and the sequence of events, starting from state S1 and triggering events E1, E4, E5, and E7 will lead the system to state S4 with the output O4. Each event transitions the system from one state to another, producing the specified output at each step until the final state and output are reached.

Topic 3, Mix Questions Set C

## Question: 235

A booking system for a city bus service prices its fares according to the time of travel:

- Peak-time tariff starts at 0600 and finishes at 1000 am
- Off-peak tariff applies during all other times of service
- The bus service does not operate between 2300 and the start of the next day's peak service

Note that all times mentioned are inclusive.

When applying the equivalence partitioning test design technique, which of the following options, shows test case inputs that each fall into a different equivalence partition?

A. 0600, 1000, 1200
B. 1001, 1300, 2259

C. 0100, 0800, 2200
D. 2400, 1000, 2301

**Answer: A**

Explanation:

Equivalence partitioning is a black-box test design technique that divides input data into partitions that are expected to be treated similarly by the system. In this case, 0600 falls into the peak-time tariff, 1000 is at the boundary of peak-time but still included in it, and 1200 falls into the off-peak tariff. This ensures that each input is from a different equivalence partition, testing different functional areas of the system.

## Question: 236

Your company is developing a system with complex business rules and many branches in the structure of its code components. You need to choose one black box technique and one white box technique for test case design. Which one of the following offers the BEST choice?

A. Statement testing and exploratory testing
B. Decision testing and equivalence partitioning
C. Decision testing and decision table testing
D. Boundary value analysis and decision table testing

**Answer: D**

Explanation:
For a system with complex business rules and many branches, decision testing (a white-box technique) is appropriate because it ensures that each decision point (branch) in the code is tested. Equivalence partitioning (a black-box technique) complements this by ensuring that all functional areas (based on different inputs) are adequately tested. This combination provides thorough coverage of both the internal logic and the functional requirements of the system.

## Question: 237

Which of the following is a Black Box test design technique?

A. Decision Coverage
B. Error Guessing
C. Statement Coverage
D. Equivalence Partitioning

**Answer: D**

Explanation:
Equivalence Partitioning is a black box test design technique used to divide input data into partitions that are expected to be treated the same by the system. It focuses on ensuring that each partition represents a set of valid or invalid states for the input, and one test case from each partition is enough to cover the entire set. This technique does not rely on the internal structure of the code but rather on the functionality and behavior

as seen from the user's perspective.

## Question: 238

Which of the following is a white-box test technique?

A. Decision table testing
B. Exploratory testing
C. Statement testing
D. Error guessing

**Answer: C**

Explanation:
Statement testing is a white-box test technique that involves creating test cases to execute specific statements in the code. The goal is to ensure that all statements in the program are executed at least once during testing. This technique requires knowledge of the internal structure of the code and focuses on verifying the flow of execution and logic within the program.

## Question: 239

During which stage of the fundamental test process is the testability of requirements evaluated?

A. Test Implementation and Execution
B. Test Planning and Control
C. Evaluating Exit Criteria and Reporting
D. Test Analysis and Design

**Answer: D**

Explanation:
The testability of requirements is evaluated during the Test Analysis and Design stage of the fundamental test process. In this stage, the test team reviews the requirements to ensure they are clear, complete, and testable. The focus is on understanding the test basis and defining the test conditions, test cases, and test data necessary to verify the requirements.

## Question: 240

You are examining a document which gives the precise steps needed in order to execute a test. What is the correct definition of this document?

A. Test design specification
B. Test condition
C. Test procedure specification
D. Test case specification

**Answer: C**

Explanation:

A Test Procedure Specification is a document that outlines the precise steps required to execute a test. It includes detailed instructions on the setup, execution, and teardown of the test environment, as well as the specific actions to be performed and the expected results. This document ensures that tests are conducted consistently and repeatably.

## Question: 241

Which of the following is NOT a valid objective of testing?

A. Preventing defects from being introduced into the code
B. Investigating and fixing defects in the software under test
C. Gaining confidence that the system is fit-for-purpose
D. Providing information for stakeholders' decision making

**Answer: B**

Explanation:

Testing is primarily aimed at detecting defects, verifying the functionality, and providing information to stakeholders for decision-making. While investigating defects is part of the testing process, fixing defects is not typically considered an objective of testing; it falls under the development activities. The main objectives of testing include preventing defects (through early testing activities), gaining confidence in the system, and providing valuable information to stakeholders.

## Question: 242

Which of the following options explain why it is often beneficial to have an independent test function in an organisation?

A. To improve defect finding during reviews and testing
B. To ensure that developers adhere to coding standards
C. To limit communication between developers and testers
D. To provide better metrics for the stakeholders

**Answer: A**

Explanation:

An independent test function in an organization helps improve defect finding during reviews and testing by providing an unbiased perspective on the software. Independent testers are not influenced by the development team's constraints and can evaluate the software more objectively. This leads to the identification of defects that might be overlooked by developers who are too close to their own code. Independent testing enhances the quality and reliability of the software by bringing fresh insights and ensuring thorough testing.

## Question: 243

A live defect has been found where a code component fails to release memory after it has finished using it.
Which of the following tools would have been the MOST effective at detecting this defect prior to live implementation?

A. Dynamic analysis tool
B. Monitoring tool
C. Configuration management tool
D. Coverage measurement tool

**Answer: A**

Explanation:
A dynamic analysis tool would have been the most effective at detecting a defect where a code component fails to release memory after use. These tools analyze the behavior of the software while it is running, allowing them to identify issues like memory leaks, which occur when memory is not properly freed. Dynamic analysis tools can monitor the allocation and deallocation of memory in real-time, helping to pinpoint the exact location and cause of the memory leak before the software is deployed .

## Question: 244

Which option BEST describes how the level of risk is determined?

A. The likelihood of an adverse event happening multiplied by the cost of preventing it
B. The consequences of a potential problem multiplied by the cost of possible legal action
C. The impact of an adverse event multiplied by the likelihood of that event occurring
D. The likelihood and the probability of a hazard occurring

**Answer: C**

Explanation:
The level of risk is determined by evaluating both the impact and the likelihood of an adverse event occurring. This approach ensures that both the severity of potential consequences and the probability of the event are considered. The impact assesses how serious the consequences would

be if the event happened, while the likelihood evaluates how probable it is that the event will occur. This combination provides a comprehensive measure of risk, helping organizations prioritize their risk management efforts effectively.

## Question: 245

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:

Common metrics used for monitoring test preparation and execution typically include quantitative measures such as the number of test cases passed and failed, the percentage of planned test cases designed, and the percentage of tasks completed in test environment preparation. These metrics provide insights into the progress and effectiveness of the testing process. However, the number of test plan review comments is not a typical metric for this purpose, as it does not directly measure the progress or quality of test execution.

## Question: 246

Refer to the exhibit.

| | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 |
|---|---|---|---|---|---|---|
| **Conditions:** | | | | | | |
| Car Driver | No | Yes | Yes | Yes | Yes | No |
| Motorcycle Driver | No | No | No | No | No | Yes |
| Diesel | N/A | Yes | Yes | No | No | N/A |
| Petrol | N/A | No | No | Yes | Yes | N/A |
| Engine < 1600cc | N/A | Yes | No | Yes | No | N/A |
| Engine >1600cc | N/A | No | Yes | No | Yes | N/A |
| **Actions:** | | | | | | |
| Can claim expenses | No | Yes | Yes | Yes | Yes | Yes |
| Expenses claim band A | N/A | Yes | No | No | No | Yes |
| Expenses claim band B | N/A | No | Yes | No | No | No |
| Expenses claim band C | N/A | No | No | Yes | No | No |
| Expenses claim band D | N/A | No | No | No | Yes | No |

The decision table above shows a company's fuel expenses structure.

Which of the following Test Cases based on the decision table are Valid?

Test Case 1:

An employee who is not a car or motorcycle driver attempts to claim fuel expenses. Expected result: Expense claim not allowed.

Test Case 2:

An employee who drives a 1700cc diesel car attempts to claim fuel expenses. Expected result: Expense claim accepted at band C.

Test Case 3:

An employee who rides a motorcycle attempts to claim fuel expenses. Expected result: Expense claim accepted at band A.

A. Test Cases 1 and 3 are Valid, Test Case 2 is Invalid.
B. Test Cases 2 and 3 are Valid, Test Case 1 is Invalid
C. Test Cases 1, 2 and 3 are all Valid.
D. Test Cases 2 is Valid, Test Cases 1 and 3 are Invalid

**Answer: A**

Explanation:

According to the decision table, Test Case 1 is valid because an employee who is not a car or motorcycle driver cannot claim expenses, as shown in Rule 1. Test Case 3 is valid because a motorcycle driver can claim expenses, as indicated in Rule 2, with expenses claim band A. However, Test Case 2 is invalid because, according to Rule 5, a car driver with a diesel engine greater than 1600cc cannot claim expenses. The expected result should be that the expense claim is not allowed, not band C.

## Question: 247

A new testing tool has been selected for an organisation and a pilot project has successfully completed. The next step is to deploy the tool within the organization.

What is a key success factor in tool deployment?

A. Estimate a cost-benefit ratio based on a firm business case
B. Determine whether benefits will be achieved at reasonable cost
C. Provide support for the test team using the tool
D. Assessment of organisational maturity, strengths and weaknesses

**Answer: C**

Explanation:

Providing support for the test team using the tool is crucial for successful deployment. This includes training the test team on how to effectively use the tool, offering ongoing support to address any issues, and integrating the tool into existing processes. By ensuring that the team is well-equipped and confident in using the tool, the organization can maximize the benefits of the tool and improve

overall testing efficiency.

## Question: 248

Your task is to compile a test execution schedule for the current release of software. The system specification states the following logical dependencies:

- An admin user must create/amend/delete a standard user.
- A standard user is necessary to perform all other actions.

The test plan requires that re-tests must be performed first, followed by the highest priority tests. To save time, the test plan states that tests should be scheduled to create test data for the subsequent tests in the schedule. The following test cases have been designed, with an indication of priority (1 being the highest priority) and whether the test has previously failed.

| Id | Description | Priority | Failed |
|---|---|---|---|
| a | Log in as standard user and create customer account | 2 | Y |
| b | Order one item | 3 | N |
| c | Create invoice for order | 1 | Y |
| d | Order two items or more | 3 | Y |
| e | Log in as admin user and create a standard user | 2 | N |

Which test execution schedule meets the test plan requirements and logical dependencies?

A. a, d, c, b, e B. a, c, b, d, e C. e, a, b, c, d D. e, a, d, c, b

**Answer: D**

Explanation:
To meet the test plan requirements and logical dependencies, the test schedule should start with retests (those that have previously failed), followed by the highest priority tests. Additionally, tests should be scheduled to create necessary test data for subsequent tests. In this case, test e (log in as admin user and create a standard user) must be done first as it is necessary to perform all other actions. Following this, test a (log in as standard user and create customer account) creates the standard user needed for subsequent actions. Then, the highest priority test d (order two items or more), which has failed before, should be executed next, followed by test c (create invoice for order), and finally test b (order one item).

## Question: 249

Which of the following does NOT represent one of the three triggers for maintenance testing an operational system?

A. Data migration
B. System retirement
C. System modification
D. Introduction of a test management tool

**Answer: D**

Explanation:
Maintenance testing is typically triggered by changes to the operational system, such as data migration, system modification, or system retirement. These triggers require testing to ensure that the changes do not negatively impact the existing functionality. However, the introduction of a test management tool does not directly affect the operational system and is not considered a trigger for maintenance testing.

## Question: 250

A system requirement states that up to 100 users should be able to carry out a transaction, with responses returned within 5 seconds.
What type of non-functional testing would you carry out to verify these requirements?

A. Stress testing
B. Maintenance testing
C. Load testing

D. Usability testing

**Answer: C**

Explanation:
Load testing is a type of non-functional testing used to verify that the system can handle a specified load of users performing transactions within acceptable performance parameters. In this case, load testing would ensure that up to 100 users can carry out transactions simultaneously with response times within 5 seconds. This testing helps identify performance bottlenecks and ensure that the system meets the performance requirements under expected user load conditions.

## Question: 251

During the development of a software change for a system, the developer makes a mistake in his work, which leads to a fault in the code. Unfortunately the fault is not found by software testing and is released into live. What is the definite consequence of this mistake?

A. The system will fail, causing a defect
B. If the defect is executed, the system may fail
C. Loss of money, time, or business reputation
D. Contractual requirements have not been met by testing

**Answer: B**

Explanation:
When a developer makes a mistake that introduces a fault in the code, and this fault is not detected during testing, it becomes a latent defect. The definite consequence is that if this defect is executed under specific conditions, it may cause the system to fail. The actual failure occurs only when the defect is triggered during the system's operation. This emphasizes the importance of thorough testing to catch such defects before the software is released.

## Question: 252

Which of the following options BEST explain the pesticide paradox principle of testing?

A. If we do not regularly review and revise our tests, we'll stop finding defects
B. Repeatedly running a set of tests will ensure that a system is defect free
C. Defects are, paradoxically, often contained in a small number of modules
D. Testing, like spraying pesticide, is an effective bug / defect removal activity

**Answer: A**

Explanation:
The pesticide paradox in testing refers to the phenomenon where continuously using the same set of tests on the software eventually becomes less effective at finding new defects. Just as insects can develop resistance to pesticides over time, software defects become less detectable by tests that are not updated. Therefore, to maintain the effectiveness of testing, it is essential to regularly review and revise the test cases to introduce new tests and update existing ones.

## Question: 253

Which one of the following statements about testing techniques is TRUE?

A. Exploratory testing can replace black box techniques when testing time is very limited
B. Test execution scheduling should give priority to experienced based testing
C. Specification based techniques can be used as a substitute for a poorly defined test basis
D. Experienced based techniques are systematic and produce detailed test documentation

**Answer: C**

Explanation:
Specification-based techniques, also known as black-box techniques, derive test cases from the specifications or requirements of the system. These techniques can be particularly useful when the test basis (such as documentation or requirements) is poorly defined or incomplete, as they provide a structured approach to testing based on the available specifications. This ensures that testing can proceed even when detailed documentation is lacking, helping to identify defects related to missing or incorrect functionality.

## Question: 254

When can functional and structural testing BOTH be applied?

A. System and Component test levels only
B. All 'Developement' test levels, i.e. those before Acceptance testing
C. Component and Component integration test levels only
D. All test levels

**Answer: D**

Explanation:
Functional and structural testing can be applied at all test levels. Functional testing focuses on verifying that the software behaves as expected based on the requirements, while structural testing examines the internal structure and workings of the software. These testing approaches are complementary and can be used together at various stages of testing, including component, integration, system, and acceptance testing. Applying both types of tests ensures a comprehensive evaluation of the software's functionality and internal structure.

Top of Form
Bottom of Form

## Question: 255

Testing effort can depend on a number of factors, which one of following is MOST likely to impact the amount of effort required?

A. The predicted number of defects and the amount of rework required
B. The ratio of developers to testers in the project team
C. The planned use of a project management tool to schedule tasks
D. The responsibilities for testers and developers being clearly defined

**Answer: A**

Explanation:
The amount of testing effort is highly influenced by the predicted number of defects and the subsequent amount of rework required. If a high number of defects are expected, more effort will be needed to identify, document, and retest the defects after they are fixed. This impacts the overall testing workload significantly compared to other factors such as the ratio of developers to testers, which affects team dynamics but not the core testing effort.

## Question: 256

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:
Structural testing, also known as white-box testing, focuses on the internal structure of the software. It includes testing all paths, achieving decision coverage, and ensuring that all branches within the component calls are tested. Ensuring that users can navigate to all fields on the screen, however, is a form of functional testing, which focuses on the behavior and usability of the application from the user's perspective, rather than its internal structure.

## Question: 257

You have been asked to improve the way test automation tools are being used in your company. Which one of the following is the BEST approach?

A. Selecting and automating scripts that test new functionality to find the most defects
B. Using a keyword-driven testing approach to separate the actions and data from the tool's script C. Ensuring that all data, inputs and actions are stored in the tool's script for ease of maintenance D. Keeping expected results separate from the automation tool to allow the testers to check the results

**Answer: B**

Explanation:
A keyword-driven testing approach is effective because it separates the test logic (actions) from the test data. This makes the scripts more reusable and easier to maintain. By organizing tests this way, any changes in the data or the actions can be made independently without affecting the other, enhancing the efficiency and manageability of automated tests.

## Question: 258

Refer to the exhibit.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 9 Holes | Don't care | N | N | Y | Y | N | Y |
| Buggy/Cart Request | Don't care | N | Y | N | Y | Don't care | Don't care |
| Actions | | | | | | | |
| No charge on Green Fees | Y | N | N | N | N | N | N |
| £12 Green Fees | N | N | N | Y | Y | N | N |
| £16 Green Fees | N | N | N | N | N | N | Y |
| £18 Green Fees | N | Y | Y | N | N | N | N |
| £22 Green Fees | N | N | N | N | N | Y | N |
| Buggy/Cart allowed | Y | Y | Y | Y | Y | N | N |
| Buggy/Cart Free | Y | N | N | N | N | N | N |
| Buggy/Cart £5 | N | N | Y | N | Y | N | N |

The decision table above reflects a golf club's pricing structure for green fees and buggy/cart hire. What is the expected result (actions) for each of the following two test cases (TC1 and TC2)?

* TC 1 - Paul is not a full member, is a Loyalty Card holder and requests to play 18 holes with a buggy/cart

* TC 2 - Cheryl is not at full member, doesn't have a Loyalty Card and requests to play 9 holes with a buggy/cart

A. TC1 - £23 total charges including buggy hire; TC2 - £21 total charge including buggy hire
B. TC1 - £18 total charges including buggy hire; TC2 - £16 total charge but no buggy allowed
C. TC1 - £23 total charges including buggy hire; TC2 - £16 total charge but no buggy allowed
D. TC1 - £17 totalcharges but no buggy allowed; TC2 - £21 total charge including buggy hire

**Answer: C**

Explanation:

According to the decision table:
For TC1, since Paul is not a full member but is a Loyalty Card holder and requests 18 holes with a buggy/cart, the applicable action is £18 Green Fees plus £5 for buggy/cart, totaling £23.
For TC2, since Cheryl is not a full member, does not have a Loyalty Card, and requests 9 holes with a buggy/cart, the applicable action is £16 Green Fees with no buggy allowed.

## Question: 259

Which of the following type of defect would NOT be typically found by using a static analysis tool?

A. A variable is defined but is then not used
B. A variable is used in a calculation before it is defined
C. A variable has the wrong numeric value passed into it
D. A variable is used but not declared

**Answer: C**

Explanation:

Static analysis tools are designed to analyze code without executing it. They are highly effective at identifying issues such as syntax errors, variable declaration, and usage problems. Options A, B, and D are types of defects that static analysis tools can typically detect because they relate to the code structure and syntax. However, Option C refers to a variable having the wrong numeric value passed into it, which is more of a logical error that typically requires dynamic testing to identify. Static analysis tools cannot understand the correct logical flow or expected values in the code, thus making Option C the correct answer .

## Question: 260

Which of the following would achieve the HIGHEST level of testing independence for a project's test level?

A. Training developers to design good tests for the test team to execute
B. Outsourcing test design and execution to a different company
C. Having the company's independent test team design and execute the tests
D. Minimising contact between testers and developers during test design to avoid bias

**Answer: B**

Explanation:
The highest level of testing independence is achieved when the testing process is performed by individuals or teams who are completely independent of the development team and the organization. Option B, which involves outsourcing test design and execution to a different company, ensures that there is no bias or influence from the development team or company politics, thereby achieving the highest level of testing independence.

## Question: 261

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:
Boundary value analysis involves testing at the boundaries between partitions. For the Frequency input, valid boundaries are 1 and 5, while invalid boundaries are 0 and 6. For the Duration input, valid boundaries are 1 and 60 minutes, while invalid boundaries are 0 and 61 minutes. Therefore, Option C includes the correct set of valid and invalid boundary values for both inputs.

## Question: 262

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:

Good testing practice involves early involvement of testers in the software development process. This helps identify defects early and provides better understanding and clarification of requirements. Involving testers in reviewing requirements or user stories as soon as drafts are available allows for early detection of defects, which is more cost-effective and efficient. This practice is beneficial regardless of the software lifecycle model used .

Top of Form

Bottom of Form

## Question: 263

Which of the following options describe the causal chain in the correct sequence?

A. Error, fault, failure
B. Fault, bug, mistake
C. Mistake, failure, fault
D. Failure, bug, error

**Answer: A**

Explanation:

The correct sequence in the causal chain is "Error, fault, failure". An error (or mistake) made by a person can lead to a fault (or defect) in the software. When the fault is executed, it may cause a failure in the software system. This sequence is important to understand the cause and effect relationship in software testing and defect management .

## Question: 264

Debugging and Testing are key activities in the software development lifecycle.

Which of the following are 'Debugging' activities?

a) Identifying, a failure
b) Locating the cause of failure
c) Fixing the defect
d) Checking the fix has resolved the failure

A. a & d
B. a & b
C. b & c
D. c & d

**Answer: C**

Explanation:

Debugging is the process that involves locating the cause of a failure and fixing the defect. Therefore, activities b) "Locating the cause of failure" and c) "Fixing the defect" are debugging activities.

Identifying a failure and checking that the fix has resolved the failure are parts of testing, not debugging.

## Question: 265

Why is measurement of code coverage Important?

A. Because 100% code coverage implies 100% coverage of requirements
B. Because 100% code coverage guarantees that there are no coding errors
C. Because code coverage can be used to ensure that all code is exercised by tests
D. Because code coverage can ensure that all decisions are correctly implemented in the code

**Answer: C**

Explanation:

Measurement of code coverage is important because it helps ensure that all code is exercised by tests. This does not guarantee that there are no coding errors or that all requirements are covered, but it helps identify untested parts of the code, allowing for better test coverage and more robust testing processes.

## Question: 266

Which of the following activities is appropriate to the test planning stage?

A. Analysing the test basis
B. Assigning resources for the planned activities
C. Designing the test environments
D. Writing a test execution schedule

**Answer: B**

Explanation:

During the test planning stage, it is important to assign resources for the planned activities. This ensures that the necessary personnel, equipment, and time are allocated to execute the test plan effectively. Analyzing the test basis, designing test environments, and writing a test execution schedule are activities that come after the initial planning and resource allocation.

## Question: 267

The following Test Cases have been created for a simple web-based airline booking system.

Test Case 1: Search for an item Available Flights
Test Case 2: View selected item in My Flights
Test Case 3: Login to the system: Login is accepted
Test Case 4: Select an available flight: item added to My Flights
Test Case 5: Print confirmation receipt, then exit
Test Case 6: In My Flights, confirm details and book flight

Which of the following is the correct logical order for the test cases?

A. 6, 3, 1, 4, 2, 5
B. 3, 4, 1, 2, 5, 6
C. 3, 2, 1, 4, 6, 5
D. 3, 1, 4, 2, 6, 5

**Answer: D**

Explanation:
The correct logical order for the test cases ensures a smooth and realistic flow through the system functionalities. The sequence should start with logging into the system (Test Case 3), searching for available flights (Test Case 1), selecting an available flight (Test Case 4), viewing the selected item in My Flights (Test Case 2), confirming details and booking the flight in My Flights (Test Case 6), and finally printing the confirmation receipt and exiting (Test Case 5) .
Top of Form
Bottom of Form

## Question: 268

Which of the following would be a good test technique to use when under severe time pressure?

A. Exploratory testing
B. Structure based testing
C. Specification based testing
D. Use Case testing

**Answer: A**

Explanation:
Exploratory testing is highly effective when under severe time pressure because it allows testers to simultaneously learn, design, and execute tests. It is a flexible approach that does not require extensive preparation and documentation, making it ideal for situations where time is limited.

## Question: 269

Which of the following would you NOT expect to see on an incident report from test execution?

A. The version(s) of the software under test
B. The test execution schedule
C. Expected results and actual results
D. Precise steps to reproduce the problem

**Answer: B**

Explanation:

An incident report typically includes the version(s) of the software under test, expected results and actual results, and precise steps to reproduce the problem. However, the test execution schedule is not a standard part of an incident report. It is more relevant to test planning and management documents rather than incident reporting.

## Question: 270

Refer to the exhibit.

| State | Events | | | | |
|---|---|---|---|---|---|
| | A | B | C | D | E |
| S1 | S2 | | | S1 | |
| S2 | | S3 | | | |
| S3 | | S4 | S2 | | |
| S4 | | | | | S4 |

In the above State Table, which of the following represents an invalid transition?

A. Event C from S3
B. Event E from S4
C. Event B from S2
D. Event D from S4

**Answer: B**

Explanation:
In the provided state table, an invalid transition is represented by an empty cell. According to the table, Event E from State S4 is an invalid transition as the cell for this combination is empty. This indicates that the transition from State S4 on Event E is not defined.

Top of Form
Bottom of Form

## Question: 271

Which of the following is a defect that is more likely to be found by a static analysis tool than by other testing techniques?

A. Omission of a major requirement
B. Inadequate decision coverage

C. Component memory leakage
D. Variables that are not used improperly declared

**Answer: D**

Explanation:

Static analysis tools are designed to analyze code without executing it. They are particularly good at finding defects related to code structure, such as variables that are not used or improperly declared. These tools can parse the code and identify issues that may not be easily detected through dynamic testing methods. Other issues like omission of a major requirement (A), inadequate decision coverage (B), and component memory leakage (C) are typically found through dynamic testing or higher-level verification techniques .

## Question: 272

Which of the following is NOT a valid use of decision coverage?

A. Checking that all decisions have been exercised in a single program
B. Checking that all decisions have been exercised in a business process
C. Checking that all calls from one program module to another have been made correctly
D. Checking that at least 50% of decisions have been exercised by a test case suite

**Answer: C**

Explanation:
Decision coverage is a testing metric used to ensure that all possible paths (decisions) in the code have been executed at least once. It is not used to verify the correctness of module calls, which is more related to integration testing or control flow testing. Options A, B, and D describe valid uses of decision coverage, ensuring that decisions are exercised in programs, business processes, and measuring the extent of coverage by a test suite .

## Question: 273

When considering the roles of test leader and tester, which of the following tasks would NOT typically be performed by a tester?

A. Prepare and acquire the test data
B. Set up and check the test environment
C. Write test summary reports
D. Review tests developed by others

**Answer: C**

Explanation:
Writing test summary reports is typically the responsibility of a test leader or manager, who consolidates testing activities and results to communicate with stakeholders. Testers are more involved in preparing and acquiring test data, setting up and checking the test environment, and reviewing tests developed by others. These tasks are more hands-on and technical, aligning with the role of a tester .

Top of Form
Bottom of Form