Number: CTFL4
Passing Score: 800.0
Time Limit: 120.0
File Version: 3.2

Exam Code: CTFL4

Exam Name: ISTQB Certified Tester Foundation Level 4.0

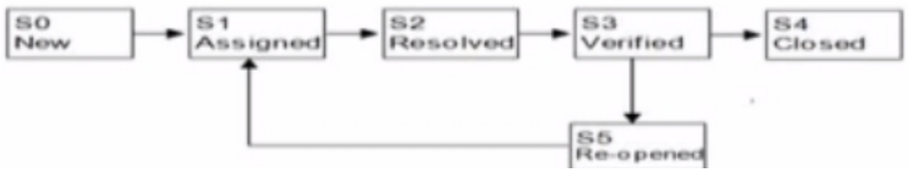
Exam A

QUESTION 1
Which of the following statements is true?

A. In Agile software development, work product documentation tends to be lightweight and manual tests tend to be often unscripted as they are often produced using experience-based test techniques
B. Sequential development models impose the use of systematic test techniques and do not allow the use of experience-based test techniques
C. In Agile software development, the first iterations are exclusively dedicated to testing activities, as testing will be used to drive development, which will be performed in the subsequent iterations
D. Both in Agile software development and in sequential development models, such as the V-model, test levels tend to overlap since they do not usually have defined entry and exit criteria

Correct Answer: A
Section:
Explanation:
This answer is correct because in Agile software development, work product documentation, such as user stories, acceptance criteria, or test cases, tends to be lightweight and concise, as the focus is on working software and frequent communication rather than comprehensive documentation. Manual tests tend to be often unscripted, as they are often produced using experience-based test techniques, such as error guessing or exploratory testing, which rely on the tester's skills, knowledge, and creativity to find defects and provide feedback.Reference: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.2, Section 3.2.1.2

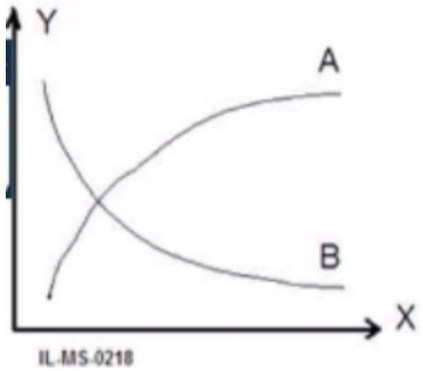
QUESTION 2
Which sequence of stated in the answer choices is correct in accordance with the following figure depicting the life-cycle of a defect?



A. S0->S1->S2->S3->S5->S1
B. S0->S1->S2->S3->S5->S1->S2->S3
C. S0->S1->S2~>S3->S4
D. S0->S1 ->S2->S3->S5->S3->S4

Correct Answer: D
Section:
Explanation:
According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0, the life cycle of a defect typically follows a sequence from its discovery to its closure. In the provided figure, it starts with S0 (New), moves to S1 (Assigned), then to S2 (Resolved), followed by S3 (Verified). If the defect is not fixed, it can be Re-opened (S5) and goes back for verification (S3). Once verified, it is Closed (S4).Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.4.3, Page 17.

QUESTION 3
The following chart represents metrics related to testing of a project that was competed. Indicate what is represented by tie lines A, B and the axes X.Y



A)

X - Time
Y - Cost
A - Cost of test (per week)
B - Cost of finding a single bug (per week)

B)

X - Time
Y - Number of defects
A - Number of open defects
B - Number of closed defects

C)

X - Time
Y - Percent
A - % of functional tests in the test suite
B - % of non-functional tests in the test suite

D)

X - Time
Y - Count
A - Total number of executed tests
B - Number of open bugs

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: D
Section:
Explanation:
Option D correctly explains what is represented by the lines A, B and the axes X, Y in a testing metrics chart. According to option D:
X-axis represents Time
Y-axis represents Count

Line A represents Number of open bugs
Line B represents Total number of executed tests
This information is essential in understanding and analyzing the testing metrics of a completed project.

QUESTION 4
Which of the following statements about estimation of the test effort is WRONG?

A. Once the test effort is estimated, resources can be identified and a schedule can be drawn up.
B. Effort estimate can be inaccurate because the quality of the product under tests is not known.
C. Effort estimate depends on the budget of the project.
D. Experience based estimation is one of the estimation techniques.

Correct Answer: C
Section:
Explanation:
Effort estimate does not depend on the budget of the project, but rather on the scope, complexity, and quality of the software product and the testing activities1. Budget is a constraint that may affect the feasibility and accuracy of the effort estimate, but it is not a factor that determines the effort estimate.Effort estimate is the amount of work required to complete the testing activities, measured in terms of person-hours, person-days, or person-months2.
The other options are correct because:
A .Once the test effort is estimated, resources can be identified and a schedule can be drawn up, as they are interrelated aspects of the test planning process3.Resources are the people, tools, equipment, and facilities needed to perform the testing activities4.Schedule is the time frame and sequence of the testing activities, aligned with the project milestones and deadlines5.
B .Effort estimate can be inaccurate because the quality of the product under tests is not known, as it affects the number and severity of the defects that may be found and the rework that may be needed to fix them6.Quality is the degree to which the software product satisfies the specified requirements and meets the needs and expectations of the users and clients7.
D . Experience based estimation is one of the estimation techniques, which relies on the judgment and expertise of the testers and other project stakeholders to estimate the test effort based on similar projects or tasks done in the past. Experience based estimation can be useful when there is a lack of historical data, formal methods, or detailed information about the software product and the testing activities.
Reference=
1ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 154
2ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 155
3ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 156
4ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 157
5ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 158
6ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 159
7ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16
[8] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 160
[9] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 161

QUESTION 5
The tests at the bottom layer of the test pyramid:

A. run faster than the tests at the top layer of the pyramid
B. cover larger pieces of functionalities than the tests at the top layer of the pyramid
C. are defined as 'Ul Tests' or 'End-To-End tests' in the different models of the pyramid
D. are unscripted tests produced by experience-based test techniques

Correct Answer: A
Section:
Explanation:
The tests at the bottom layer of the test pyramid run faster than the tests at the top layer of the pyramid because they are more focused, isolated, and atomic. They usually test individual units or components of the software system, such as classes, methods, or functions. They are also easier to maintain and execute, as they have fewer dependencies and interactions with other parts of the system. The tests at the top layer of the test pyramid, on

the other hand, are slower because they cover larger pieces of functionalities, such as user interfaces, workflows, or end-to-end scenarios. They also have more dependencies and interactions with other systems, such as databases, networks, or external services. They are more complex and costly to maintain and execute, as they require more setup and teardown procedures, test data, and test environments.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 3.2.1, Test Pyramid1
ISTQB Glossary of Testing Terms v4.0, Test Pyramid2

QUESTION 6
Test automation allows you to:

A. demonstrate the absence of defects
B. produce tests that are less subject to human errors
C. avoid performing exploratory testing
D. increase test process efficiency by facilitating management of defects

Correct Answer: B
Section:
Explanation:
Test automation allows you to produce tests that are less subject to human errors, as they can execute predefined test scripts or test cases with consistent inputs, outputs, and expected results. Test automation can also reduce the manual effort and time required to execute repetitive or tedious tests, such as regression tests, performance tests, or data-driven tests. Test automation does not demonstrate the absence of defects, as it can only verify the expected behavior of the system under test, not the unexpected or unknown behavior. Test automation does not avoid performing exploratory testing, as exploratory testing is a valuable technique to discover new information, risks, or defects that are not covered by automated tests. Test automation does not increase test process efficiency by facilitating management of defects, as defect management is a separate activity that involves reporting, tracking, analyzing, and resolving defects, which may or may not be related to automated tests.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 3.3.1, Test Automation1
ISTQB Glossary of Testing Terms v4.0, Test Automation2

QUESTION 7
Which of the following statements about how different types of test tools support testers is true?

A. The support offered by a test data preparation tool is often leveraged by testers to run automated regression test suites
B. The support offered by a performance testing tool is often leveraged by testers to run load tests
C. The support offered by a bug prediction tool is often used by testers to track the bugs they found
D. The support offered by a continuous integration tool is often leveraged by testers to automatically generate test cases from a model

Correct Answer: B
Section:
Explanation:
The support offered by a performance testing tool is often leveraged by testers to run load tests, which are tests that simulate a large number of concurrent users or transactions on the system under test, in order to measure its performance, reliability, and scalability. Performance testing tools can help testers to generate realistic workloads, monitor system behavior, collect and analyze performance metrics, and identify performance bottlenecks.
The other statements are false, because:
A test data preparation tool is a tool that helps testers to create, manage, and manipulate test data, which are the inputs and outputs of test cases. Test data preparation tools are not directly related to running automated regression test suites, which are test suites that verify that the system still works as expected after changes or modifications. Regression test suites are usually executed by test execution tools, which are tools that can automatically run test cases and compare actual results with expected results.
A bug prediction tool is a tool that uses machine learning or statistical techniques to predict the likelihood of defects in a software system, based on various factors such as code complexity, code churn, code coverage, code smells, etc. Bug prediction tools are not used by testers to track the bugs they found, which are the actual defects that have been detected and reported during testing. Bugs are usually tracked by defect management tools, which are tools that help testers to record, monitor, analyze, and resolve defects.
A continuous integration tool is a tool that enables the integration of code changes from multiple developers into a shared repository, and the execution of automated builds and tests, in order to ensure the quality and consistency of the software system. Continuous integration tools are not used by testers to automatically generate test cases from a model, which are test cases that are derived from a representation of the system under test, such as a state diagram, a decision table, a use case, etc. Test cases can be automatically generated by test design tools, which are tools that support the implementation and maintenance of test cases, based on test design specifications or test models.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

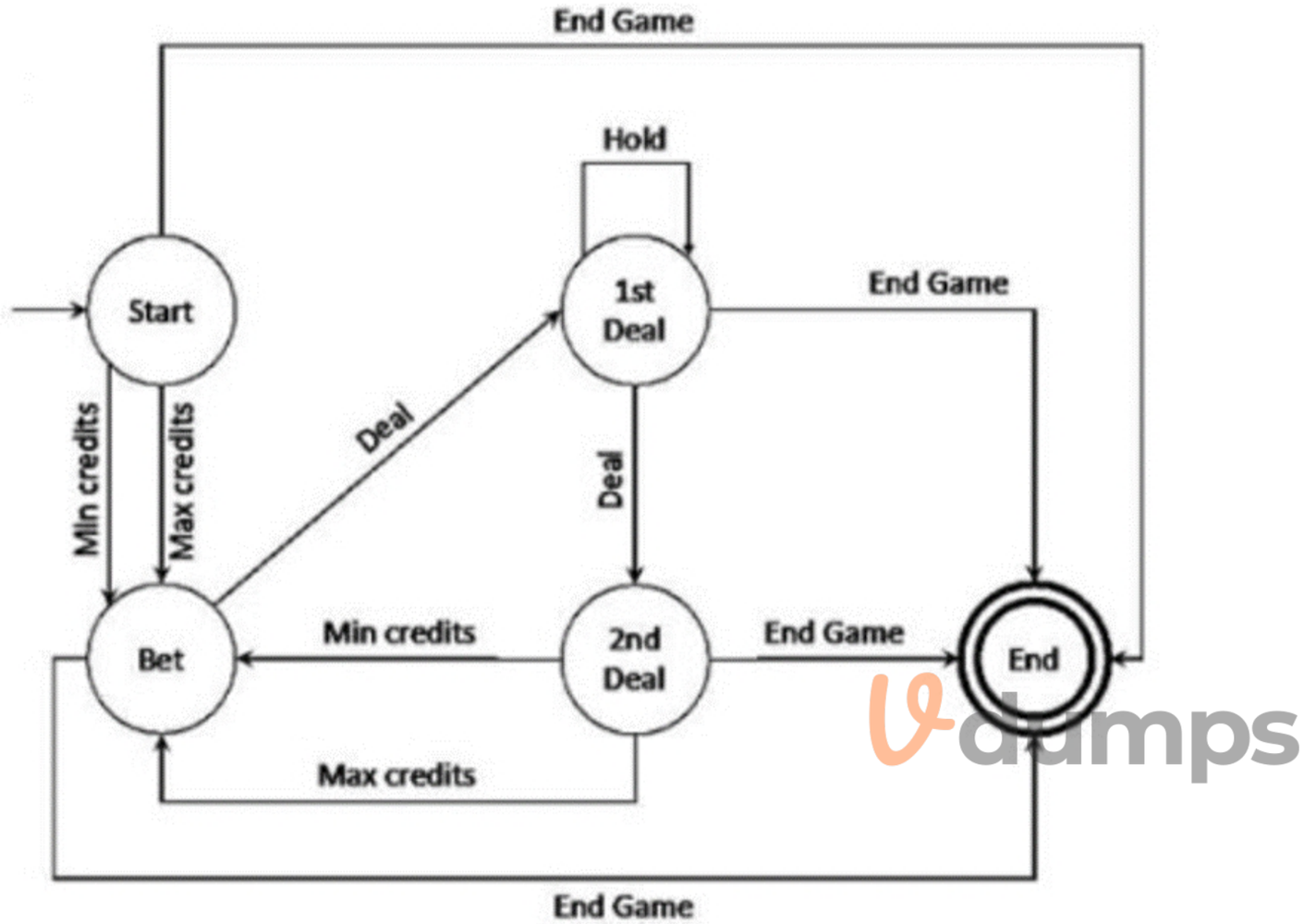
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 3.4.1, Types of Test Tools
ISTQB Glossary of Testing Terms v4.0, Performance Testing Tool, Test Data Preparation Tool, Bug Prediction Tool, Continuous Integration Tool, Test Execution Tool, Defect Management Tool, Test Design Tool

QUESTION 8
Which of the following statements about branch coverage is true?

A. The minimum number of test cases needed to achieve full branch coverage, is usually lower than that needed to achieve full statement coverage
B. If full branch coverage has been achieved, then all unconditional branches within the code have surely been exercised
C. If full branch coverage has been achieved, then all combinations of conditions in a decision table have surely been exercised
D. Exercising at least one of the decision outcomes for all decisions within the code, ensures achieving full branch coverage

Correct Answer: D
Section:
Explanation:
Exercising at least one of the decision outcomes for all decisions within the code, ensures achieving full branch coverage, which is a test coverage criterion that requires that all branches in the control flow of the code are executed at least once by the test cases. A branch is a basic block of code that has a single entry point and a single exit point, and a decision is a point in the code where the control flow can take more than one direction, such as an if-then-else statement, a switch-case statement, a loop statement, etc. The decision outcomes are the possible paths that can be taken from a decision, such as the then branch or the else branch, the case branch or the default branch, the loop body or the loop exit, etc. The other statements are false, because:
The minimum number of test cases needed to achieve full branch coverage, is usually higher than that needed to achieve full statement coverage, which is a test coverage criterion that requires that all executable statements in the code are executed at least once by the test cases. This is because branch coverage is a stronger criterion than statement coverage, as it implies statement coverage, but not vice versa. For example, a single test case can achieve full statement coverage for an if-then-else statement, but two test cases are needed to achieve full branch coverage, as both the then branch and the else branch need to be exercised.
If full branch coverage has been achieved, then all unconditional branches within the code have not necessarily been exercised, as unconditional branches are branches that do not depend on any decision, and are always executed, such as a goto statement, a break statement, a return statement, etc. Unconditional branches are not part of the branch coverage criterion, as they do not represent different paths in the control flow of the code. However, they are part of the statement coverage criterion, as they are executable statements in the code.
If full branch coverage has been achieved, then all combinations of conditions in a decision table have not necessarily been exercised, as a decision table is a test design technique that represents the logical relationships between multiple conditions and their corresponding actions, in a tabular format. A decision table can have more combinations of conditions than the number of decision outcomes in the code, as each condition can have two or more possible values, such as true or false, yes or no, etc. For example, a decision table with four conditions can have 16 combinations of conditions, but the corresponding code may have only two decision outcomes, such as pass or fail. To exercise all combinations of conditions in a decision table, a stronger test coverage criterion is needed, such as condition combination coverage, which requires that all possible combinations of condition outcomes in the code are executed at least once by the test cases.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.3.1, Test Coverage Criteria Based on the Structure of the Software
ISTQB Glossary of Testing Terms v4.0, Branch Coverage, Statement Coverage, Branch, Decision, Decision Outcome, Unconditional Branch, Decision Table, Condition Combination Coverage

QUESTION 9
Consider the following simplified version of a state transition diagram that specifies the behavior of a video poker game:



What Is the minimum number of test cases needed to cover every unique sequence of up to 3 states/2 transitions starting In the 'Start' state and ending In the 'End' state?

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

Correct Answer: D
Section:
Explanation:
The minimum number of test cases needed to cover every unique sequence of up to 3 states/2 transitions starting in the ''Start'' state and ending in the ''End'' state is 4. This is because there are 4 unique sequences of up to 3 states/2 transitions starting in the ''Start'' state and ending in the ''End'' state:
Start -> Bet -> End
Start -> Deal -> End
Start -> 1st Deal -> End
Start -> 2nd Deal -> EndReference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents.

QUESTION 10
Consider a given test plan which, among others, contains the following three sections: 'Test Scope', 'Testing Communication', and 'Stakeholders'. The features of the test object to be tested and those excluded from the testing represent information that is:

A. not usually included in a test plan, and therefore in the given test plan it should not be specified neither within the three sections mentioned, nor within the others
B. usually included in a test plan and, in the given test plan, it is more likely to be specified within 'Test Scope' rather than in the other two sections mentioned
C. usually included in a test plan and, in the given test plan, it is more likely to be specified within 'Testing Communication' rather than in the other two sections mentioned
D. usually included in a test plan and, in the given test plan, it is more likely to be specified within 'Stakeholders' rather than in the other two sections mentioned

Correct Answer: B
Section:
Explanation:
The features of the test object to be tested and those excluded from the testing represent information that is usually included in a test plan and, in the given test plan, it is more likely to be specified within ''Test Scope'' rather than in the other two sections mentioned. The test scope defines the boundaries and limitations of the testing activities, such as the test items, the features to be tested, the features not to be tested, the test objectives, the test environment, the test resources, the test assumptions, the test risks, etc. The test scope helps to establish a common understanding of what is included and excluded from the testing, and to avoid ambiguity, confusion, or misunderstanding among the stakeholders. The other two sections, ''Testing Communication'' and ''Stakeholders'', are also important parts of a test plan, but they do not directly address the features of the test object. The testing communication describes the methods, frequency, and responsibilities for the communication and reporting of the testing progress, status, issues, and results. The stakeholders identify the roles and responsibilities of the people involved in or affected by the testing activities, such as the test manager, the test team, the project manager, the developers, the customers, the users, etc.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1
ISTQB Glossary of Testing Terms v4.0, Test Plan, Test Scope2

QUESTION 11
An alphanumeric password must be between 4 and 7 characters long and must contain at least one numeric character, one capital (uppercase) letter and one lowercase letter of the alphabet.
Which one of the following sets of test cases represents the correct outcome of a two-value boundary value analysis applied to the password length? (Note: test cases are separated by a semicolon)

A. 1xA;aB11;Pq1ZZab;7iDD0a1x
B. aB11;99rSp:5NnN10;7iDD0a1x
C. 1xB: aB11: 99rSp: 5NnN10; 4NnN10T; 44ghWn19
D. 1RhT;rSp53;3N3e10;8sBdby

Correct Answer: D
Section:
Explanation:
The correct outcome of a two-value boundary value analysis applied to the password length is the set of test cases represented by option D. Boundary value analysis is a test design technique that focuses on the values at the boundaries of an equivalence partition, such as the minimum and maximum values, or the values just above and below the boundaries. A two-value boundary value analysis uses two values for each boundary, one representing the valid value and one representing the invalid value. For example, if the valid range of values is from 4 to 7, then the two values for the lower boundary are 3 and 4, and the two values for the upper boundary are 7 and 8. The test cases in option D use these values for the password length, while also satisfying the other requirements of the password, such as containing at least one numeric character, one capital letter, and one lowercase letter. The test cases in option D are:
1RhT: a 4-character password that is valid
rSp53: a 5-character password that is valid
3N3e10: a 6-character password that is valid
8sBdby: an 8-character password that is invalid The test cases in the other options are incorrect, because they either use values that are not at the boundaries of the password length, or they do not meet the other requirements of the password. For example, the test cases in option A are:
1xA: a 3-character password that is invalid, but it does not contain a capital letter
aB11: a 4-character password that is valid
Pq1ZZab: a 7-character password that is valid
7iDD0a1x: an 8-character password that is invalidReference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.1, Black-box Test Design Techniques1

QUESTION 12
A company wants to reward each of its salespeople with an annual bonus that represents the sum of all the bonuses accumulated for every single sale made by that salesperson. The bonus for a single sale can take on the following four values: 3%, 5%, 7% and 10% (the percentage refers to the amount of the single sale). These values are determined on the basis of the type of customer (classified as 'Basic' or 'Premium') to which such sale was made, and on the amount of such sale classified into the following three groups G1, G2 and G3:
* [G1]: less than 300 euros
* [G2]: between 300 and 2000 euros
* [G3]: greater than 2000 euros
Which of the following is the minimum number of test cases needed to cover the full decision table associated with this scenario?

A. 12
B. 6
C. 4
D. 3

Correct Answer: B
Section:
Explanation:
The minimum number of test cases needed to cover the full decision table associated with this scenario is 6. This is because the decision table has 4 conditions (type of customer and amount of sale) and 4 actions (bonus percentage). The conditions have 2 possible values each (Basic or Premium, and G1, G2 or G3), so the total number of combinations is 2 x 2 x 2 x 2 = 16. However, not all combinations are valid, as some of them are contradictory or impossible. For example, a sale cannot be both less than 300 euros and greater than 2000 euros at the same time. Therefore, we need to eliminate the invalid combinations and keep only the valid ones. The valid combinations are:
Type of customer
Amount of sale
Bonus percentage
Basic
G1
3%
Basic
G2
5%
Basic
G3
7%
Premium
G1
5%
Premium
G2
7%
Premium
G3
10%
These 6 combinations cover all the possible values of the conditions and actions, and they are the minimum number of test cases needed to cover the full decision table.

QUESTION 13
Which of the following statements is true?

A. Experience-based test techniques rely on the experience of testers to identify the root causes of defects found by black-box test techniques
B. Some of the most common test basis used by white-box test techniques include user stories, use cases and business processes
C. Experience-based test techniques are often useful to detect hidden defects that have not been targeted by black-box test techniques
D. The primary goal of experience-based test techniques is to design test cases that can be easily automated using a GUI-based test automation tool

Correct Answer: C
Section:
Explanation:
Experience-based test techniques are test design techniques that rely on the experience, knowledge, intuition, and creativity of the testers to identify and execute test cases that are likely to find defects in the software system. Experience-based test techniques are often useful to detect hidden defects that have not been targeted by black-box test techniques, which are test design techniques that use the external behavior and specifications of the software system as the test basis, without considering its internal structure or implementation. Experience-based test techniques can complement black-box test techniques by covering aspects that are not explicitly specified, such as usability, security, reliability, performance, etc. The other statements are false, because:
Experience-based test techniques do not rely on the experience of testers to identify the root causes of defects found by black-box test techniques, but rather to identify the potential sources of defects based on their own insights, heuristics, or exploratory testing. The root causes of defects are usually identified by debugging or root cause analysis, which are activities that involve examining the code or the development process to find and fix the errors that led to the defects.
Some of the most common test basis used by white-box test techniques include the source code, the design documents, the architecture diagrams, and the control flow graphs of the software system. White-box test techniques are test design techniques that use the internal structure and implementation of the software system as the test basis, and aim to achieve a certain level of test coverage based on the code elements, such as statements, branches, paths, etc. User stories, use cases, and business processes are examples of test basis used by black-box test techniques, as they describe the functional and non-functional requirements of the software system from the perspective of the users or the stakeholders.
The primary goal of experience-based test techniques is not to design test cases that can be easily automated using a GUI-based test automation tool, but rather to design test cases that can reveal defects that are not easily detected by other test techniques, such as boundary value analysis, equivalence partitioning, state transition testing, etc. Test automation is the use of software tools to execute test cases and compare actual results with expected results, without human intervention. Test automation can be applied to different types of test techniques, depending on the test objectives, the test levels, the test tools, and the test resources. However, test automation is not always feasible or beneficial, especially for test cases that require human judgment, creativity, or exploration, such as those designed by experience-based test techniques.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.1, Black-box Test Design Techniques
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.2, White-box Test Design Techniques
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.3, Experience-based Test Design Techniques
ISTQB Glossary of Testing Terms v4.0, Experience-based Test Technique, Black-box Test Technique, White-box Test Technique, Test Basis, Test Coverage, Test Automation

QUESTION 14
A new web app aims at offering a rich user experience. As a functional tester, you have run some functional tests to verify that, before releasing the app, such app works correctly on several mobile devices, all of which are listed as supported devices within the requirements specification. These tests were performed on stable and isolated test environments where you were the only user interacting with the application. All tests passed, but in some of those tests you observed the following issue: on some mobile devices only, the response time for two web pages containing images was extremely slow.
Based only on the given information, which of the following recommendation would you follow?

A. You should open a defect report providing detailed information on which devices and by running which tests you observed the issue
B. The issue is related to performance efficiency, not functionality. Thus, as a functional tester, you should not open any defect report as all the functional tests passed
C. You should not open any defect report as the problem is most likely due to poor hardware equipment on the devices where you observed the issue
D. You should not open any defect report and inform the test manager that the devices on which you observed the issue should no longer be supported so that they will be removed from the requirements specification

Correct Answer: A
Section:
Explanation:
As a functional tester, you should open a defect report providing detailed information on which devices and by running which tests you observed the issue. A defect report is a document that records the occurrence, nature, and status of a defect detected during testing, and provides information for further investigation and resolution. A defect report should include relevant information such as the defect summary, the defect description, the defect severity, the defect priority, the defect status, the defect origin, the defect category, the defect reproduction steps, the defect screenshots, the defect attachments, etc. Opening a defect report is a good practice for any tester who finds a defect in the software system, regardless of the type or level of testing performed. The other options are not recommended, because:
The issue is related to performance efficiency, not functionality, but that does not mean that as a functional tester, you should not open any defect report as all the functional tests passed. Performance efficiency is a quality characteristic that measures how well the software system performs its functions under stated conditions, such as the response time, the resource utilization, the throughput, etc. Performance efficiency is an important

aspect of the user experience, especially for web applications that run on different devices and networks. Even if the functional tests passed, meaning that the software system met the functional requirements, the performance issue observed on some devices could still affect the user satisfaction, the usability, the reliability, and the security of the software system. Therefore, as a functional tester, you have the responsibility to report the performance issue as a defect, and provide as much information as possible to help the developers or the performance testers to investigate and resolve it.

QUESTION 15
For each of the test cases to be executed, the following table specifies the priority order and dependencies on other test cases

Test Case	Priority	Logical Dependencies
TC1	Low	TC5
TC2	High	TC3
TC3	High	TC4
TC4	High	-
TC5	Low	TC2
TC6	Medium	-

Which of the following test execution schedules is compatible with the logical dependencies and allows executing the test cases in priority order?

A. TC4, TC3, TC2, TC6, TC5. TC1
B. TC4, TC6, TC3, TC2, TC5, TC1
C. TC3, TC5, TC6, TC1, TC4, TC3
D. TC4, TC3, TC2, TC6, TC1, TC5

Correct Answer: D
Section:
Explanation:
This answer is correct because it follows the logical dependencies and allows executing the test cases in priority order. TC4, TC3, and TC2 are executed first because they have the highest priority. TC6 is executed next because it has a logical dependency on TC2. TC1 is executed next because it has a logical dependency on TC5. Finally, TC5 is executed last because it has the lowest priority.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents

QUESTION 16
Which of the following statements about error guessing is true?

A. Error guessing is a system that adopts artificial intelligence to predict whether software components are likely to contain defects or not
B. Experienced testers, when applying error guessing, rely on the use of a high-level list of what needs to be tested as a guide to find defects
C. Error guessing refers to the ability of a system or component to continue normal operation despite the presence of erroneous inputs
D. Experienced testers, when applying error guessing technique, can anticipate where errors, defects and failures have occurred and target their tests at those issues

Correct Answer: D
Section:
Explanation:
This answer is correct because error guessing is a test design technique where the experience and intuition of the tester are used to anticipate where errors, defects and failures have occurred or are likely to occur, and to design test cases to expose them. Error guessing can be based on factors such as the complexity of the system or component, the known or suspected weaknesses of the system or component, the previous history of defects, or the common types of errors in the domain or technology. Error guessing can be used as a complementary technique to other more systematic or formal techniques, or when there is insufficient information or time to apply them.Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.5

QUESTION 17

Which of the following is a task the Author is responsible for, as part of a typical formal review?

A. Determining the people who will be involved in the review
B. Recording the anomalies found during the review meeting
C. Identifying potential anomalies in the work product under review
D. Fixing the anomalies found in the work product under review

Correct Answer: C
Section:
Explanation:
This answer is correct because identifying potential anomalies in the work product under review is one of the tasks the Author is responsible for, as part of a typical formal review. The Author is the person who creates the work product to be reviewed, such as a requirement specification, a design document, or a test case. The Author's tasks include preparing the work product for the review, identifying potential anomalies in the work product, and fixing the anomalies found in the work product after the review.Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.4.2.1

QUESTION 18
Which of the following are the phases of the ISTQB fundamental test process?

A. Test planning and control, Test analysis and design, Test implementation and execution, Evaluating ex t criteria and reporting. Test closure activities
B. Test planning, Test analysis and design. Test implementation and control. Checking test coverage and reporting, Test closure activities
C. Test planning and control, Test specification and design. Test implementation and execution, Evaluating test coverage and reporting, Retesting and regression testing, Test closure activities
D. Test planning. Test specification and design. Test implementation and execution. Evaluating exit criteria and reporting. Retesting and test closure activities

Correct Answer: A
Section:
Explanation:
The ISTQB fundamental test process consists of five main phases, as described in the ISTQB Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15:
Test planning and control: This phase involves defining the test objectives, scope, strategy, resources, schedule, risks, and metrics, as well as monitoring and controlling the test activities and results throughout the test process.
Test analysis and design: This phase involves analyzing the test basis (such as requirements, specifications, or user stories) to identify test conditions (such as features, functions, or scenarios) that need to be tested, and designing test cases and test procedures (such as inputs, expected outcomes, and execution steps) to cover the test conditions. This phase also involves evaluating the testability of the test basis and the test items (such as software or system components), and selecting and implementing test techniques (such as equivalence partitioning, boundary value analysis, or state transition testing) to achieve the test objectives and optimize the test coverage and efficiency.
Test implementation and execution: This phase involves preparing the test environment (such as hardware, software, data, or tools) and testware (such as test cases, test procedures, test data, or test scripts) for test execution, and executing the test procedures or scripts according to the test plan and schedule. This phase also involves logging the outcome of test execution, comparing the actual results with the expected results, and reporting any discrepancies as incidents (such as defects, errors, or failures).
Evaluating exit criteria and reporting: This phase involves checking if the planned test activities have been completed and the exit criteria (such as quality, coverage, or risk levels) have been met, and reporting the test results and outcomes to the stakeholders. This phase also involves making recommendations for the release or acceptance decision based on the test results and outcomes, and identifying any residual risks (such as known defects or untested areas) that need to be addressed or mitigated.
Test closure activities: This phase involves finalizing and archiving the testware and test environment for future reuse, and evaluating the test process and the test project against the test objectives and the test plan. This phase also involves identifying any lessons learned and best practices, and communicating the findings and suggestions for improvement to the relevant parties.
Reference= ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 37-38; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 88, page 32.

QUESTION 19
Which of the following issues cannot be identified by static analysis tools?

A. Very low MTBF (Mean Time Between failure)
B. Potentially endless loops

C. Referencing a variable with an undefined value
D. Security vulnerabilities

Correct Answer: A
Section:
Explanation:
Static analysis tools are software tools that examine the source code of a program without executing it.They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential bugs12.However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)3. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures.MTBF can only be estimated by dynamic testing, which involves executing the program under various conditions and collecting data on its failures4. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools. The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can be identified by static analysis tools.Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false5.Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables6. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption.Reference=What Is Static Analysis? Static Code Analysis Tools - Perforce Software,How Static Code Analysis Works | Perforce,Static Code Analysis: Techniques, Top 5 Benefits & 3 Challenges,What is MTBF? Mean Time Between Failures Explained | Perforce,Static analysis tools - Software Testing MCQs - CareerRide,ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

QUESTION 20
A company runs a pilot project for evaluation of a test automation tool. Which of the following is NOT a valid object of this pilot project?

A. Get familiar with the functionality and options of the tool
B. Check haw the tool fits to the existing test processes
C. Train all testers on using the tool
D. Decide upon standards for tool implementation

Correct Answer: C
Section:
Explanation:
A pilot project is a small-scale experiment or trial that is conducted to evaluate the feasibility, effectiveness, and suitability of a test automation tool before implementing it on a larger scale1.
The objectives of a pilot project may vary depending on the context and scope of the test automation initiative, but some common ones are2:
To get familiar with the functionality and options of the tool
To check how the tool fits to the existing test processes and environment
To assess the benefits and challenges of using the tool
To decide upon standards and guidelines for tool implementation and usage
To estimate the costs and resources required for tool deployment and maintenance
Therefore, option C is not a valid objective of a pilot project, as it is not necessary to train all testers on using the tool at this stage. Training all testers on using the tool would be more appropriate after the tool has been selected and approved for full-scale implementation, and after the standards and guidelines have been established. Training all testers on using the tool during the pilot project would be inefficient, costly, and premature, as the tool may not be suitable or effective for the intended purpose, or may be replaced by another tool later.
1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82
2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 83
: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 84
: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 85

QUESTION 21
Which of the following applications will be the MOST suitable for testing by Use Cases

A. Accuracy and usability of a new Navigation system compared with previous system
B. A billing system used to calculate monthly charge based or large number of subscribers parameters

C. The ability of an Anti virus package to detect and quarantine a new threat
D. Suitability and performance of a Multi media (audio video based) system to a new operating system

Correct Answer: A
Section:
Explanation:
A new navigation system compared with a previous system is the most suitable application for testing by use cases, because it involves a high level of interaction between the user and the system, and the expected behavior and outcomes of the system are based on the user's needs and goals. Use cases can help to specify the functional requirements of the new navigation system, such as the ability to enter a destination, select a route, follow the directions, receive alerts, etc. Use cases can also help to compare the accuracy and usability of the new system with the previous system, by defining the success and failure scenarios, the preconditions and postconditions, and the alternative flows of each use case. Use cases can also help to design and execute test cases that cover the main and exceptional paths of each use case, and to verify the satisfaction of the user's expectations.
The other options are not the most suitable applications for testing by use cases, because they do not involve a high level of interaction between the user and the system, or the expected behavior and outcomes of the system are not based on the user's needs and goals. A billing system used to calculate monthly charge based on a large number of subscriber parameters is more suitable for testing by data-driven testing, which is a technique for testing the functionality and performance of a system or component by using a large set of input and output data. The ability of an antivirus package to detect and quarantine a new threat is more suitable for testing by exploratory testing, which is a technique for testing the functionality and security of a system or component by using an informal and flexible approach, based on the tester's experience and intuition. The suitability and performance of a multimedia (audio video based) system to a new operating system is more suitable for testing by compatibility testing, which is a technique for testing the functionality and performance of a system or component by using different hardware, software, or network environments.Reference= CTFL 4.0 Syllabus, Section 3.1.1, page 28-29; Section 4.1.1, page 44-45; Section 4.2.1, page 47-48.

QUESTION 22
A Test Manager conducts risk assessment for a project. One of the identified risks is: The sub-contractor may fail to meet his commitment'. If this risk materializes. it will lead to delay in completion of testing required for the current cycle.
Which of the following sentences correctly describes the risk?

A. It is a product risk since any risk associated with development timeline is a product risk.
B. It is no longer a risk for the Test Manager since an independent party (the sub-contractor) is now managing it
C. It is a object risk since successful completion of the object depends on successful and timely completion of the tests
D. It is a product risk since default on part of the sub-contractor may lead to delay in release of the product

Correct Answer: D
Section:
Explanation:
A product risk is a risk that affects the quality or timeliness of the software product being developed or tested1.Product risks are related to the requirements, design, implementation, verification, and maintenance of the software product2.
The risk of the sub-contractor failing to meet his commitment is a product risk, as it could cause a delay in the completion of the testing required for the current cycle, which in turn could affect the release date of the product.The release date is an important aspect of the product quality, as it reflects the customer satisfaction and the market competitiveness of the product3.
The other options are not correct because:
A . It is not true that any risk associated with development timeline is a product risk.Some risks could be project risks, which are risks that affect the management or control of the software project, such as budget, resources, schedule, or communication1. For example, a risk of losing a key project stakeholder is a project risk, not a product risk.
B . It is not true that the risk is no longer a risk for the Test Manager since an independent party is managing it.The Test Manager is still responsible for ensuring that the testing activities are completed according to the test plan and the quality objectives4.The Test Manager should monitor and control the sub-contractor's performance and communicate with him regularly to identify and mitigate any potential issues or deviations5.
C .It is not clear what is meant by ''object'' in this option, but it could be interpreted as the software system under test or the test object6. In any case, the risk is not an object risk, as it does not affect the successful completion of the object, but rather the successful completion of the testing of the object.An object risk could be a risk that affects the functionality, reliability, usability, efficiency, maintainability, or portability of the software system under test2. For example, a risk of the software system having a high complexity or a low testability is an object risk, not a product risk.
Reference=
1ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 97
2ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 98
3ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 99
4ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 100
5ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 101

6ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 102

QUESTION 23
Who of the following has the best knowledge to decide what tests in a test project should be automated?

A. The developer
B. The customer
C. The development manager
D. The test leader

Correct Answer: D
Section:
Explanation:
The test leader is the person who is responsible for planning, monitoring, and controlling the test activities and resources in a test project. The test leader should have the best knowledge of the test objectives, scope, risks, resources, schedule, and quality criteria.The test leader should also be aware of the test automation criteria, such as the execution frequency, the test support, the team education, the roles and responsibilities, and the devs and testers collaboration1. Based on these factors, the test leader can decide which tests are suitable for automation and which are not, and prioritize them accordingly.The test leader can also coordinate with the test automation engineers, the developers, and the stakeholders to ensure the alignment of the test automation strategy with the test project goals and expectations.Reference= ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 2, Section 2.3.1, Page 152; ISTQB Glossary of Testing Terms v4.0, Page 403; ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 6, Section 6.1.1, Page 514; Top 8 Test Automation Criteria You Need To Fulfill - QAMIND1

QUESTION 24
A requirement specifies that a certain identifier (ID) must be between 5 and 10 characters long, must contain only alphanumenc characters, and its first character must be a letter As a tester, you want to apply one-dimensional equivalence partitioning to test this ID. This means that you have to apply equivalence partitioning individually: to the length of the ID, the type of characters contained within the ID, and the type of the first character of the ID. What is the number of partitions to cover?

A. 7.
B. 6.
C. 5.
D. 3.

Correct Answer: A
Section:
Explanation:
To apply one-dimensional equivalence partitioning to the ID requirement, we need to consider each condition individually:
Length of the ID: Valid partitions (5-10 characters), Invalid partitions (less than 5, more than 10) = 3 partitions.
Type of characters: Valid partitions (alphanumeric), Invalid partitions (non-alphanumeric) = 2 partitions.
First character: Valid partitions (letter), Invalid partitions (non-letter) = 2 partitions. Adding these partitions, we get a total of 3 (length) + 2 (character type) + 2 (first character) = 7 partitions. Thus, the correct answer is A.

QUESTION 25
A requirement specifies that if the total amount of sales (TAS) made during the year by a corporate seller is 300,000 or more, the bonus that must be paid to the seller is 100% of a certain amount agreed upon at the beginning of the year. The software contains a fault as it implements this requirement with the decision 'IF (TAS = 300,000)' instead of 'IF (TAS >= 300.000)'. The application of the 3-value boundary value analysis to this problem consists of the following three test cases (TAS is an integer variable):
TC1 = 299,999 TC2=300,000 TC=300,001
Which of the following statements is TRUE?

A. TC1 would highlight the fault
B. TC2 would highlight the fault
C. TC3 would highlight the fault

D. None of the three test cases would highlight the fault.

Correct Answer: B
Section:
Explanation:
The requirement specifies that a bonus should be paid if the total amount of sales (TAS) made during the year is 300,000 or more. The software incorrectly implements this requirement with 'IF (TAS = 300,000)' instead of 'IF (TAS >= 300,000)'. Using boundary value analysis (BVA), which is a common technique in software testing, the three test cases provided (TC1 = 299,999, TC2 = 300,000, and TC3 = 300,001) cover the critical boundary values around the condition.
TC1 tests just below the boundary (299,999),
TC2 tests exactly at the boundary (300,000),
TC3 tests just above the boundary (300,001).
Since the software incorrectly checks for TAS equal to 300,000, only TC2 will fail because the condition is exactly met and highlights the incorrect implementation of the decision logic.

QUESTION 26
What type of testing measures its effectiveness by tracking which lines of code were executed by the tests?

A. Acceptance testing
B. Structural testing
C. Integration testing
D. Exploratory testing

Correct Answer: B
Section:
Explanation:
Structural testing is a type of testing that measures its effectiveness by tracking which lines of code were executed by the tests. Structural testing, also known as white-box testing or glass-box testing, is based on the internal structure, design, or implementation of the software. Structural testing aims to verify that the software meets the specified quality attributes, such as performance, security, reliability, or maintainability, by exercising the code paths, branches, statements, conditions, or data flows. Structural testing uses various coverage metrics, such as function coverage, line coverage, branch coverage, or statement coverage, to determine how much of the code has been tested and to identify any untested or unreachable parts of the code. Structural testing can be applied at any level of testing, such as unit testing, integration testing, system testing, or acceptance testing, but it is more commonly used at lower levels, where the testers have access to the source code.
The other options are not correct because they are not types of testing that measure their effectiveness by tracking which lines of code were executed by the tests. Acceptance testing is a type of testing that verifies that the software meets the acceptance criteria and the user requirements. Acceptance testing is usually performed by the end-users or customers, who may not have access to the source code or the technical details of the software. Acceptance testing is more concerned with the functionality, usability, or suitability of the software, rather than its internal structure or implementation. Integration testing is a type of testing that verifies that the software components or subsystems work together as expected. Integration testing is usually performed by the developers or testers, who may use both structural and functional testing techniques to check the interfaces, interactions, or dependencies between the components or subsystems. Integration testing is more concerned with the integration logic, data flow, or communication of the software, rather than its individual lines of code. Exploratory testing is a type of testing that involves simultaneous learning, test design, and test execution. Exploratory testing is usually performed by the testers, who use their creativity, intuition, or experience to explore the software and discover any defects, risks, or opportunities for improvement. Exploratory testing is more concerned with the behavior, quality, or value of the software, rather than its internal structure or implementation.Reference= ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54; Chapter 1: Fundamentals of Testing, Section 1.4: Testing Throughout the Software Development Lifecycle, Pages 11-13; Chapter 3: Static Testing, Section 3.4: Exploratory Testing, Pages 40-41.

QUESTION 27
A program is used to control a manufacturing line (turn machines on and off. start and stop conveyer belts, add raw materials to the flow. etc.). Not all actions are possible at all times. For example, there are certain manufacturing stages that cannot be stopped - unless there is an emergency. A tester attempts to evaluate if all such cases (where a specific action is not allowed) are covered by the tests.
Which coverage metric will provide the needed information for this analysis?

A. Code coverage
B. Data flow coverage
C. Statement coverage
D. Branch Coverage

Correct Answer: D
Section:
Explanation:
Branch coverage is a type of structural coverage metric that measures the percentage of branches or decision outcomes that are executed by the test cases. A branch is a point in the code where the control flow can take two or more alternative paths based on a condition. For example, an if-else statement is a branch that can execute either the if-block or the else-block depending on the evaluation of the condition. Branch coverage ensures that each branch is taken at least once by the test cases, and thus reveals the behavior of the software under different scenarios. Branch coverage is also known as decision coverage or all-edges coverage.
Branch coverage is suitable for testing the cases where a specific action is not allowed, because it can verify that the test cases cover all the possible outcomes of the conditions that determine the action. For example, if the program has a condition that checks if the manufacturing stage can be stopped, then branch coverage can ensure that the test cases cover both the cases where the stage can be stopped and where it cannot be stopped. This way, branch coverage can help identify any missing or incorrect branches that may lead to undesired or unsafe actions.
The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types of coverage metrics, such as statement coverage, branch coverage, data flow coverage, etc. Code coverage does not specify which type of coverage metric is used for the analysis. Data flow coverage is a type of structural coverage metric that measures the percentage of data flow paths that are executed by the test cases. A data flow path is a sequence of statements that define, use, or kill a variable. Data flow coverage is useful for testing the correctness and completeness of the data manipulation in the software, but not for testing the conditions that determine the actions. Statement coverage is a type of structural coverage metric that measures the percentage of statements or lines of code that are executed by the test cases. Statement coverage ensures that each statement is executed at least once by the test cases, but it does not reveal the behavior of the software under different scenarios. Statement coverage is a weaker criterion than branch coverage, because it does not account for the branches or decision outcomes in the code.Reference= ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

QUESTION 28
What is test oracle?

A. The source of lest objectives
B. The source for the actual results
C. The source of expected results
D. The source of input conditions

Correct Answer: C
Section:
Explanation:
A test oracle is a mechanism or principle that can be used to determine whether the observed behavior or output of a system under test is correct or not1.A test oracle can be based on various sources of expected results, such as specifications, user expectations, previous versions, comparable systems, etc2.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.2.1, Page 91; ISTQB Glossary of Testing Terms, Version 4.0, Page 332.

QUESTION 29
Which ONE of the following statements does NOT describe how testing contributes to higher quality?

A. Properly designed tests that pass reduce the level of risk in a system.
B. The testing of software demonstrates the absence of defects.
C. Software testing identifies defects, which can be used to improve development activities.
D. Performing a review of the requirement specifications before implementing the system can enhance quality.

Correct Answer: B
Section:
Explanation:
The testing of software does not demonstrate the absence of defects, but rather the presence of defects or the conformance of the software to the specified requirements1.Testing can never prove that the software is defect-free, as it is impossible to test all possible scenarios, inputs, outputs, and behaviors of the software2.Testing can only provide a level of confidence in the quality of the software, based on the coverage, effectiveness, and efficiency of the testing activities3.
The other options are correct because:
A .Properly designed tests that pass reduce the level of risk in a system, as they verify that the system meets the expected quality attributes and satisfies the needs and expectations of the users and clients4.Risk is the potential for loss or harm due to the occurrence of an undesirable event5.Testing can help to identify, analyze, prioritize, and mitigate the risks associated with the software product and project6.

C .Software testing identifies defects, which can be used to improve development activities, as they provide feedback on the quality of the software and the effectiveness of the development processes7. Defects are flaws or errors in the software that cause it to deviate from the expected or required results or behavior. Testing can help to detect, report, track, and resolve the defects, and prevent them from recurring in the future.
D . Performing a review of the requirement specifications before implementing the system can enhance quality, as it can ensure that the requirements are clear, complete, consistent, testable, and aligned with the needs and expectations of the users and clients. Requirements are the specifications of what the software should do and how it should do it. Testing can help to validate that the requirements are met by the software, and verify that the software is implemented according to the requirements.
Reference=
1ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 10
2ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 11
3ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 12
4ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 13
5ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 97
6ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 98
7ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 14
[8] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 15
[9] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16
[10] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 17
[11] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 18
[12] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 19

QUESTION 30
Which of the following statements best describes how configuration management supports testing?

A. Configuration management helps reduce testing effort by identifying a manageable number of test environment configurations in which to test the software, out of all possible configurations of the environment in which the software will be released
B. Configuration management is an administrative discipline that includes change control, which is the process of controlling the changes to identified items referred to as Configuration Items'
C. Configuration management is an approach to interoperability testing where tests are executed in the cloud, as the cloud can provide cost-effective access to multiple configurations of the test environments
D. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation

Correct Answer: D
Section:
Explanation:
This answer is correct because configuration management is a process of establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation. This supports testing by providing traceability, consistency, and control over the test artifacts and the software under test.Reference: : ISTQB Glossary of Testing Terms v4.0, : ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.2

QUESTION 31
Which of the following statements about exploratory testing is true?

A. Exploratory testing is an experience-based test technique in which testers explore the requirements specification to detect non testable requirements
B. When exploratory testing is conducted following a session-based approach, the issues detected by the testers can be documented in session sheets
C. Exploratory testing is an experience-based test technique used by testers during informal code reviews to find defects by exploring the source code
D. In exploratory testing, testers usually produce scripted tests and establish bidirectional traceability between these tests and the items of the test basis

Correct Answer: B
Section:
Explanation:
Exploratory testing is an experience-based test technique in which testers dynamically design and execute tests based on their knowledge, intuition, and learning of the software system, without following predefined test scripts or test cases. Exploratory testing can be conducted following a session-based approach, which is a structured way of managing and measuring exploratory testing. In a session-based approach, the testers perform

uninterrupted test sessions, usually lasting between 60 and 120 minutes, with a specific charter or goal, and document the issues detected, the test coverage achieved, and the time spent in session sheets. Session sheets are records of the test activities, results, and observations during a test session, which can be used for reporting, debriefing, and learning purposes. The other statements are false, because:
Exploratory testing is not a test technique in which testers explore the requirements specification to detect non testable requirements, but rather a test technique in which testers explore the software system to detect functional and non-functional defects, as well as to learn new information, risks, or opportunities. Non testable requirements are requirements that are ambiguous, incomplete, inconsistent, or not verifiable, which can affect the quality and effectiveness of the testing process. Non testable requirements can be detected by applying static testing techniques, such as reviews or inspections, to the requirements specification, before the software system is developed or tested.
Exploratory testing is not a test technique used by testers during informal code reviews to find defects by exploring the source code, but rather a test technique used by testers during dynamic testing to find defects by exploring the behavior and performance of the software system, without examining the source code. Informal code reviews are static testing techniques, in which the source code is analyzed by one or more reviewers, without following a formal process or using a checklist, to identify defects, violations, or improvements. Informal code reviews are usually performed by developers or peers, not by testers.
In exploratory testing, testers usually do not produce scripted tests and establish bidirectional traceability between these tests and the items of the test basis, but rather produce unscripted tests and adapt them based on the feedback and the findings of the testing process. Scripted tests are tests that are designed and documented in advance, with predefined inputs, outputs, and expected results, and are executed according to a test plan or a test procedure. Bidirectional traceability is the ability to trace both forward and backward the relationships between the items of the test basis, such as the requirements, the design, the risks, etc., and the test artifacts, such as the test cases, the test results, the defects, etc. Scripted tests and bidirectional traceability are usually associated with more formal and structured testing approaches, such as specification-based or structure-based test techniques, not with exploratory testing.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.3, Experience-based Test Design Techniques1
ISTQB Glossary of Testing Terms v4.0, Exploratory Testing, Session-based Testing, Session Sheet, Non Testable Requirement, Static Testing, Informal Review, Dynamic Testing, Scripted Testing, Bidirectional Traceability2

QUESTION 32
In a two-hour uninterrupted test session, performed as part of an iteration on an Agile project, a heuristic checklist was used to help the tester focus on some specific usability issues of a web application.
The unscripted tests produced by the tester's experience during such session belong to which one of the following testing quadrants?

A. Q1
B. Q2
C. Q3
D. Q4

Correct Answer: C
Section:
Explanation:

The unscripted tests produced by the tester's experience during the two-hour test session belong to the testing quadrant Q3. The testing quadrants are a classification of testing types based on two dimensions: the test objectives (whether the testing is focused on supporting the team or critiquing the product) and the test basis (whether the testing is based on the technology or the business). The testing quadrants are labeled as Q1, Q2, Q3, and Q4, and each quadrant represents a different testing perspective, such as unit testing, acceptance testing, usability testing, or performance testing. The testing quadrant Q3 corresponds to the testing types that have the objective of critiquing the product from the business perspective, such as exploratory testing, usability testing, user acceptance testing, alpha testing, beta testing, etc. The unscripted tests performed by the tester in the given scenario are examples of exploratory testing and usability testing, as they are based on the tester's experience, intuition, and learning of the web application, and they focus on some specific usability issues, such as the user interface, the user satisfaction, the user feedback, etc. The other options are incorrect, because:

The testing quadrant Q1 corresponds to the testing types that have the objective of supporting the team from the technology perspective, such as unit testing, component testing, integration testing, system testing, etc. These testing types are usually performed by developers or testers who have access to the source code, the design, the architecture, or the configuration of the software system, and they aim to verify the functionality, the quality, and the reliability of the software system at different levels of integration.

The testing quadrant Q2 corresponds to the testing types that have the objective of supporting the team from the business perspective, such as functional testing, acceptance testing, story testing, scenario testing, etc. These testing types are usually performed by testers or customers who have access to the requirements, the specifications, the user stories, or the business processes of the software system, and they aim to validate that the software system meets the expectations and the needs of the users and the stakeholders.

The testing quadrant Q4 corresponds to the testing types that have the objective of critiquing the product from the technology perspective, such as performance testing, security testing, reliability testing, compatibility testing, etc. These testing types are usually performed by testers or specialists who have access to the tools, the metrics, the standards, or the benchmarks of the software system, and they aim to evaluate the non-functional aspects of the software system, such as the efficiency, the security, the reliability, or the compatibility of the software system under different conditions or environments.Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles

ISTQB Glossary of Testing Terms v4.0, Testing Quadrant, Exploratory Testing, Usability Testing, Unit Testing, Component Testing, Integration Testing, System Testing, Functional Testing, Acceptance Testing, Story Testing,

Scenario Testing, Performance Testing, Security Testing, Reliability Testing, Compatibility Testing

QUESTION 33
Which of the following is not an example of a typical content of a test completion report for a test project?

A. The additional effort spent on test execution compared to what was planned
B. The unexpected test environment downtime that resulted in slower test execution
C. The residual risk level if a risk-based test approach was adopted
D. The test procedures of all test cases that have been executed

Correct Answer: D
Section:
Explanation:
This answer is correct because the test procedures of all test cases that have been executed are not a typical content of a test completion report for a test project. A test completion report is a document that summarizes the test activities and results at the end of a test project. It usually includes information such as the test objectives, scope, approach, resources, schedule, results, deviations, issues, risks, lessons learned, and recommendations for improvement. The test procedures of all test cases that have been executed are part of the test documentation, but they are not relevant for the test completion report, as they do not provide a high-level overview of the test project outcomes and performance.Reference: ISTQB Foundation Level Syllabus v4.0, Section 2.5.3.2