Number: CTFL_Syll2018
Passing Score: 800
Time Limit: 120
File Version: 11.0

Exam Code: CTFL_Syll2018

Exam Name: ISTQB Certified Tester Foundation Level

Exam A

QUESTION 1
Which of the following is NOT a factor on which test estimation is dependent upon?

A. Defect debugging and resolution
B. The outcome of testing of previous test cycle
C. Characteristics of the development process
D. Characteristics of the product

Correct Answer: B
Section:
Explanation:
Test estimation is the process of predicting the effort, time, and resources required for testing a software system. Test estimation depends on several factors that influence the scope, complexity, and quality of testing. Some of these factors are:
Characteristics of the product: This factor indicates how difficult or challenging it is to test the software system based on its size, functionality, reliability, usability, performance, security, etc.
Characteristics of the development process: This factor indicates how well or poorly the software system is developed based on the development methodology, standards, tools, techniques, etc.
Characteristics of the test process: This factor indicates how effective or efficient the testing process is based on the test strategy, plan, design, execution, evaluation, etc.
Characteristics of the test team: This factor indicates how skilled or experienced the test team is based on their knowledge, competence, motivation, communication, collaboration, etc.
The outcome of testing of previous test cycle is not a factor on which test estimation is dependent upon because it does not affect the effort, time, or resources required for testing the current test cycle. Rather, it is an outcome or consequence of test estimation that can provide feedback and lessons learned for improving future test estimations. You can find more information about test estimation inSoftware Testing Foundations: A Study Guide for the Certified Tester Exam, Chapter 22.

QUESTION 2
Which of the following is a task of the Test Analysis and Design activity of the test process?

A. Measuring the percentage of prepared test cases with what was actually prepared
B. Identifying necessary test data to support the test conditions and test cases
C. Verifying that the test environment has been set up correctly
D. Checking test logs against the exit criteria specified in test planning

Correct Answer: B
Section:
Explanation:
The Test Analysis and Design activity of the test process is the activity where test conditions and test cases are identified and specified based on the test objectives and criteria. The Test Analysis and Design activity also involves identifying and preparing the necessary test data and test environment to support the test execution. Some of the tasks of the Test Analysis and Design activity are:
Reviewing the test basis: This task involves reviewing the requirements, specifications, design documents, or other sources of information that form the basis for testing.
Identifying test conditions: This task involves identifying what needs to be tested based on the test basis and objectives.
Designing and prioritizing test cases: This task involves designing specific scenarios or steps to verify each test condition and assigning priorities to them based on their importance or risk level.
Identifying necessary test data to support the test conditions and test cases: This task involves identifying what data is needed to execute each test case and how to obtain or generate it.
Identifying and preparing the test environment: This task involves identifying what hardware, software, network, data, tools, etc., are needed to execute the test cases and how to set up and maintain them.
Measuring the percentage of prepared test cases with what was actually prepared is not a task of the Test Analysis and Design activity because it does not involve identifying or specifying anything related to testing. Rather, it is a task of the Test Monitoring and Control activity, which involves measuring and reporting the progress and status of testing against the planned activities and criteria. You can find more information about the Test Analysis and Design activity inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 31.

QUESTION 3

Given the following requirement:

Requirement ID: 2.8
Requirement Description: Additional Entrance Fee
Detailed Description:
An additional fee of $3 is charged during the weekend, but
1) Visitors aged under 7 are not charged.
2) Visitors aged 7 to 13 inclusive get a 20% discount off the additional fee.
3) Visitors aged greater than 65 get a 50% discount off the additional fee.
Age should be an integer of 0 or above.
Weekend means Friday to Sunday inclusive.

Which of the following statements is NOT correct?

A. 7 and 13 are boundary values for the equivalence partition including age 10.
B. Thursday is a valid input boundary value
C. A minimum of 6 valid test cases are derived from boundary value analysis based on input age
D. $3.01 is a valid output boundary value

Correct Answer: D
Section:
Explanation:
The requirement given in the image specifies an additional fee of $3 that is charged during the weekend, with some exceptions and discounts based on the age of the visitors. To test this requirement, we can use boundary value analysis, which is a specification-based test technique that involves testing the values at or near the boundaries of an equivalence partition. An equivalence partition is a set of values that are expected to be treated in the same way by the system under test. For example, based on the requirement, we can identify the following equivalence partitions for the input age:
EP1: Age < 0 (invalid)
EP2: Age = 0 (valid, no charge)
EP3: 0 < Age < 7 (valid, no charge)
EP4: Age = 7 (valid, 20% discount)
EP5: 7 < Age < 13 (valid, 20% discount)
EP6: Age = 13 (valid, 20% discount)
EP7: 13 < Age < 65 (valid, full charge)
EP8: Age = 65 (valid, 50% discount)
EP9: Age > 65 (valid, 50% discount)
The boundary values for each equivalence partition are the values at or near the edges of the partition. For example, the boundary values for EP3 are 1 and 6. The boundary values for EP4 are 6 and 7. The boundary values for EP5 are 7 and 12. And so on.
To test this requirement using boundary value analysis, we need to select one value from each boundary and test it with different combinations of weekend and weekday. For example, we can select the following values:
BV1: Age = -1 (from EP1)
BV2: Age = 0 (from EP2 and EP3)
BV3: Age = 6 (from EP3 and EP4)
BV4: Age = 7 (from EP4 and EP5)
BV5: Age = 12 (from EP5 and EP6)
BV6: Age = 13 (from EP6 and EP7)
BV7: Age = 64 (from EP7 and EP8)
BV8: Age = 65 (from EP8 and EP9)
BV9: Age = 66 (from EP9)
We can then create test cases using these values and different combinations of weekend and weekday. For example:
TC1: Age = -1, Weekend = Yes -> Invalid input
TC2: Age = -1, Weekend = No -> Invalid input
TC3: Age = 0, Weekend = Yes -> No charge
TC4: Age = 0, Weekend = No -> No charge
TC5: Age = 6, Weekend = Yes -> No charge
TC6: Age = 6, Weekend = No -> No charge

TC7: Age = 7, Weekend = Yes -> $2.40 ($3 - 20% discount)
TC8: Age = 7, Weekend = No -> No charge
TC9: Age = 12, Weekend = Yes -> $2.40 ($3 - 20% discount)
TC10: Age = 12, Weekend = No -> No charge
TC11: Age = 13, Weekend = Yes -> $2.40 ($3 - 20% discount)
TC12: Age = 13, Weekend = No -> No charge
TC13: Age = 64, Weekend = Yes -> $3
TC14: Age = 64, Weekend = No -> No charge
TC15: Age = 65, Weekend = Yes -> $1.50 ($3 - 50% discount)
TC16: Age = 65, Weekend = No -> No charge
TC17: Age = 66, Weekend = Yes -> $1.50 ($3 - 50% discount)
TC18: Age =
66, Weekend = No -> No charge
Therefore, we need a minimum of18valid test cases to achieve100%boundary value coverage based on input age.
$3.01 is not a valid output boundary value because it is not a possible output value based on the requirement. The output values can only be $0, $1.50, $2.40, or $3 depending on the input age and weekend status.
You can find more information about boundary value analysis in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 4, Section 4.2.

QUESTION 4
Which of the following defect types are LEAST likely to be discovered when using static analysis tools?

A. Variables that are never used
B. Coding standard violations
C. Memory leaks
D. Uncalled functions and procedures

Correct Answer: C
Section:
Explanation:
Static analysis tools are tools that examine the code or design of a software system without executing it. Static analysis tools can be used to find defects, measure complexity, check compliance, or improve quality. Some examples of defect types that can be found by static analysis tools are:
Variables that are never used: This defect type occurs when a variable is declared but not referenced or assigned in the code, which indicates a waste of memory or a logic error.
Coding standard violations: This defect type occurs when the code does not follow the predefined rules or conventions for formatting, naming, commenting, etc., which affects the readability and maintainability of the code.
Uncalled functions and procedures: This defect type occurs when a function or procedure is defined but not called or invoked in the code, which indicates a waste of resources or a missing functionality.
Memory leaks are defect types that are least likely to be found by static analysis tools because they are related to the dynamic behavior and performance of the software system. Memory leaks occur when a program does not release memory that it has allocated, causing the system to run out of memory and slow down or crash. Memory leaks can only be detected by dynamic analysis tools that monitor the memory usage of the program during execution. You can find more information about static analysis tools in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 6, Section 6.3.

QUESTION 5
Out of the following, what is not needed to specify in defect report?

A. How to fix the defect
B. Severity and priority
C. Test environment details
D. How to reproduce the defect

Correct Answer: A
Section:
Explanation:
How to fix the defect is not needed to specify in a defect report because it is not part of the information that is required to describe, reproduce, and prioritize the defect. How to fix the defect is part of the solution or

resolution that is provided by the developers or maintainers who are responsible for correcting the defect. A defect report is a document that records and communicates the details of a defect found during testing. A defect report typically includes the following information:
Defect ID: A unique identifier for the defect
Summary: A brief description of the defect
Severity and priority: An indication of how serious and urgent the defect is
Test environment details: A description of the hardware, software, network, data, tools, etc., that were used when the defect was found
How to reproduce the defect: A step-by-step procedure to recreate the defect
Expected and actual results: A comparison of what should have happened and what actually happened when the defect occurred
Attachments: Any additional information or evidence that can help understand or resolve the defect, such as screenshots, logs, files, etc.
You can find more information about defect reports in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 5.

QUESTION 6
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

Correct Answer: A
Section:
Explanation:
A circumstance where maintenance testing is required is A. Migration of software onto a new platform. Migration of software onto a new platform is a type of change that affects the delivered software product and requires maintenance testing to ensure that the software product still works correctly and consistently on the new platform. A new platform can be a different hardware, operating system, network, database, browser, etc., that supports or interacts with the software product. Migration of software onto a new platform can introduce compatibility issues, performance issues, security issues, etc., that need to be detected and resolved by maintenance testing. A detailed explanation of migration testing can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], page 82.

QUESTION 7
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

Correct Answer: D
Section:
Explanation:
The statement that best defines static techniques is D. Manual examination and automated analysis of code or project documentation. Static techniques are techniques that analyze the code or other software artifacts (such as requirements, specifications, designs, models, test cases, etc.) without executing them, and provide information about their quality, defects, complexity, maintainability, etc. Static techniques can be performed manually or automatically by using tools or methods such as reviews, inspections, walkthroughs, checklists, standards, guidelines, static analysis tools, code metrics tools, etc. Static techniques can help to improve the quality and consistency of code or other software artifacts throughout the software development life cycle and reduce the cost and effort of testing and debugging. A detailed explanation of static techniques can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 27-34.

QUESTION 8
Which of the following is a role of a formal review? [K1]

A. Adjudicator

B. Moderator
C. Governor
D. Corrector

Correct Answer: B
Section:
Explanation:
A role of a formal review is B. Moderator. A formal review is a type of static technique that involves a structured process of examining code or other software artifacts by a team of reviewers who follow predefined roles and rules. A formal review has four main phases: planning, preparation, meeting, and follow-up. A formal review has five main roles: author (the person who created the code or other software artifact under review), moderator (the person who leads and facilitates the review process), reviewer (the person who examines the code or other software artifact under review and provides feedback), scribe (the person who records the issues and outcomes of the review meeting), and manager (the person who monitors and controls the review process). A moderator is a key role in a formal review because he or she is responsible for planning, organizing, conducting, and reporting the review activities and ensuring that the review objectives are met. A detailed explanation of formal reviews can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 35-37.

QUESTION 9
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.

Correct Answer: D
Section:
Explanation:
This is the definition of a data driven approach to test automation design, which allows for reusability and maintainability of test scripts. Option A describes a keyword driven approach, which uses action words to describe the test steps and data. Option B is not related to test automation design, but rather to performance testing. Option C is also not related to test automation design, but rather to a testing technique.

QUESTION 10
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.

Correct Answer: B
Section:
Explanation:
Likelihood of failure = 10%, potential cost of impact = $500,000. The level of risk to the project can be calculated by multiplying the likelihood of failure by the potential cost of impact. This gives us the following values for each option:
A: 1% x $1m = $10,000 B. 10% x $500,000 = $50,000 C. 20% x $150,000 = $30,000 D. 5% x $500,000 = $25,000

QUESTION 11
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 risk profile.

(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

Correct Answer: A
Section:
Explanation:
These are some of the factors that should be considered to determine whether enough testing has been performed. The exit criteria define the specific goals and requirements for testing completion. The budget limits the resources and time available for testing. The product's risk profile indicates the areas of highest priority and impact for testing. Sufficient details of the system status allow decisions on whether further testing is needed or not. The other options are not relevant factors for determining enough testing.

QUESTION 12
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.

Correct Answer: A
Section:
Explanation:
A test condition is an item or event of a component or system that can be verified by one or more test cases. A test condition can be derived from various sources of information, such as requirements, specifications, design documents, use cases, user stories, etc. A test condition can also be based on various aspects of a component or system, such as functionality, usability, performance, reliability, security, etc.
The other statements are not true about test conditions because they describe different concepts related to testing. For example:
B: The grouping of a composite set of test cases which, when tested as a whole, reveal a positive or negative result: This statement describes a test suite, which is a collection of test cases that are intended to be executed together to achieve a specific test objective or goal.
C: A testable component derived from business requirements: This statement describes a test basis, which is the source of information or data that provides the basis for designing and executing test cases.
D: Applies to software testing only: This statement is false because test conditions can apply to any type of testing, not just software testing. Test conditions can also be used for testing hardware, systems, processes, etc.
You can find more information about test conditions in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 4, Section 4.1.

QUESTION 13
Which of the following is a role of a formal review? [K1]

A. Adjudicator
B. Moderator
C. Governor
D. Corrector

Correct Answer: B
Section:
Explanation:
A role of a formal review is B. Moderator. A formal review is a type of static technique that involves a structured process of examining code or other software artifacts by a team of reviewers who follow predefined roles and rules. A formal review has four main phases: planning, preparation, meeting, and follow-up. A formal review has five main roles: author (the person who created the code or other software artifact under review), moderator

(the person who leads and facilitates the review process), reviewer (the person who examines the code or other software artifact under review and provides feedback), scribe (the person who records the issues and outcomes of the review meeting), and manager (the person who monitors and controls the review process). A moderator is a key role in a formal review because he or she is responsible for planning, organizing, conducting, and reporting the review activities and ensuring that the review objectives are met. A detailed explanation of formal reviews can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 35-37.

QUESTION 14
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

Correct Answer: B
Section:
Explanation:
A potential infinite loop is a type of defect that can be typically identified using static analysis by tools1.Static analysis is a technique that examines the source code or other software artifacts without executing them, and can detect defects, vulnerabilities, code smells, and deviations from standards early in the development process1.Static analysis tools are software tools that automate the static analysis technique and provide various features and functionalities to support it1.Static analysis tools can identify a potential infinite loop by analyzing the control flow and data flow of the source code and checking for conditions or statements that may cause an endless repetition or iteration1.A potential infinite loop can cause serious problems in software performance, functionality, reliability, and security1. Therefore, a potential infinite loop is a type of defect that can be typically identified using static analysis by tools.

QUESTION 15
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

Correct Answer: D
Section:
Explanation:
UAT stands for user acceptance testing, which is a type of testing that verifies that the software product meets the acceptance criteria and expectations of the end users or customers1.A UAT specification is a document that defines the scope, objectives, approach, and criteria for UAT1.Testers should be involved in reviewing a UAT specification at any time before UAT begins, as this can help to ensure that the UAT specification is clear, complete, consistent, testable, and aligned with the user requirements1.Testers can also provide feedback and suggestions to improve the UAT specification and avoid potential issues or conflicts during UAT execution1. Therefore, testers should be involved in reviewing a UAT specification at any time before UAT begins.

QUESTION 16
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

Correct Answer: B
Section:
Explanation:
Having the company's independent test team design and execute the tests would achieve the highest level of testing independence for a project's test level, because it would reduce the bias and influence of the developers or other stakeholders who are involved in the software development1.An independent test team can provide a different perspective and focus on testing activities without being affected by development pressures or deadlines1. The other options would not achieve the highest level of testing independence for a project's test level.Option A is not a good option, because training developers to design good tests for the test team to execute would still introduce some bias and influence from the developers' point of view1.Option C is not a good option, because minimizing contact between testers and developers during test design to avoid bias would reduce the communication and collaboration that are essential for effective testing and defect resolution1.Option D is not a good option, because outsourcing test design and execution to a different company would introduce some risks and challenges such as contractual issues, cultural differences, or loss of control over quality

QUESTION 17
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

Correct Answer: C
Section:
Explanation:
Frequency 0, 1, 5, 6; Duration 0, 1, 30, 60, 61 is the correct test set of valid and invalid boundary values applying 2-value boundary value analysis1.Boundary value analysis is a technique that tests the values at and near the boundaries of an input domain or output range1.A 2-value boundary value analysis tests two values for each boundary: one valid value and one invalid value1. For Frequency, the valid boundaries are 1 and 5, so the test set should include one valid value (1 or 5) and one invalid value (0 or 6) for each boundary. For Duration, the valid boundaries are 1 and 60, so the test set should include one valid value (1 or 60) and one invalid value (0 or 61) for each boundary. The other options are incorrect test sets of valid and invalid boundary values applying 2-value boundary value analysis. Option A only includes valid values for each boundary, but not invalid values. Option B includes invalid values that are not adjacent to the boundaries (59 seconds and 60 minutes 1 second). Option D includes more than two values for each boundary (0, 1, 2 for Frequency; 0, 1, 30 for Duration).

QUESTION 18
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

Correct Answer: B
Section:

Explanation:
They should be involved in reviewing requirements or user stories as soon as drafts are available is the option that represents good testing practice for testers, irrespective of the software lifecycle model used, because it allows testers to provide early feedback, identify potential defects, and clarify test objectives and scope1.Early involvement of testers in reviewing requirements or user stories can improve the quality and testability of the software and reduce the cost and effort of testing1. The other options do not represent good testing practice for testers, irrespective of the software lifecycle model used.Option A is not a good option, because they should start test analysis as early as possible, not when the corresponding development level is complete, to avoid delays and inefficiencies in testing1.Option C is not a good option, because they should ensure that different test objectives apply to each test level, not the same test objectives, to avoid redundancy and gaps in testing1.Option D is not a good option, because they should minimize the ratio of development levels to test levels to reduce project costs, but rather balance the development and test levels according to the project needs and risks1.

QUESTION 19
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

Correct Answer: A
Section:
Explanation:
Error, fault, failure is the option that describes the causal chain in the correct sequence1.The causal chain is a model that explains how defects are introduced and manifested in software1.An error is a human action or decision that produces an incorrect result1.A fault is a defect in the software caused by an error1.A failure is an incorrect behaviour of the software caused by a fault1. The other options do not describe the causal chain in the correct sequence. Option B uses synonyms for error (mistake), fault (bug), and failure (defect), but does not follow the correct order. Option C uses synonyms for error (mistake) and fault (failure), but does not follow the correct order and confuses fault with failure. Option D uses synonyms for fault (bug) and error (error), but does not follow the correct order and confuses failure with error.

QUESTION 20
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

Correct Answer: C
Section:
Explanation:
b & c are debugging activities, because they involve locating the cause of failure and fixing the defect in the software . Debugging is the process of finding and removing the causes of software failures . The other options are not debugging activities, but rather testing activities. Option A is a testing activity, because it involves identifying a failure in the software by comparing the expected and actual results . Option D is a testing activity, because it involves checking the fix has resolved the failure by re-testing the software .

QUESTION 21
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

Correct Answer: C
Section:
Explanation:
Code coverage can be used to ensure that all code is exercised by tests, because it measures how much of the code or functionality of the software under test has been exercised by the test cases . Code coverage can help to identify untested or dead code, improve test design, and assess test quality . The other options are not true statements about code coverage. Option A is false, because 100% code coverage does not imply 100% coverage of requirements, as there may be defects or gaps in the requirements or in the mapping between requirements and code . Option B is false, because 100% code coverage does not guarantee that there are no coding errors, as there may be logical or functional errors that are not detected by code coverage . Option D is false, because code coverage does not ensure that all decisions are correctly implemented in the code, as it only measures whether the decisions are executed or not, but not whether they produce the correct results .

QUESTION 22
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

Correct Answer: B
Section:
Explanation:
Assigning resources for the planned activities is an activity that is appropriate to the test planning stage, because it involves estimating and allocating the necessary human and physical resources for testing . Test planning is the process of defining the objectives, scope, approach, and resources for testing . The other options are not activities that are appropriate to the test planning stage, but rather to other stages of testing. Option A is an activity that is appropriate to the test analysis stage, because it involves analysing the test basis (such as requirements, design, or code) to identify test conditions . Option C is an activity that is appropriate to the test implementation stage, because it involves designing and preparing the test environments (such as hardware, software, network, or data) for test execution . Option D is an activity that is appropriate to the test execution stage, because it involves writing a test execution schedule that defines the order and dependencies of test cases or procedures .

QUESTION 23
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

Correct Answer: D
Section:
Explanation:

The correct logical order for the test cases is 3, 1, 4, 2, 6, 5. This order follows the logical sequence of actions that a user would perform to book a flight using a web-based airline booking system. The order also respects the dependencies between test cases, such as logging in before searching for flights or booking a flight before printing a confirmation receipt . The other options are incorrect logical orders for the test cases. Option A does not follow the logical sequence of actions and does not respect the dependencies between test cases. For example, it starts with confirming details and booking a flight before logging in or selecting a flight. Option B does not follow the logical sequence of actions and does not respect the dependencies between test cases. For example, it starts with selecting a flight before logging in or searching for flights. Option C does not follow the logical sequence of actions and does not respect the dependencies between test cases. For example, it starts with viewing selected items in My Flights before selecting any flights.

QUESTION 24
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

Correct Answer: A
Section:
Explanation:
Exploratory testing would be a good test technique to use when under severe time pressure, because it is a type of experience-based technique that allows testers to design and execute tests based on their intuition, knowledge, and skills, without following a predefined test plan or test cases . Exploratory testing can be useful when testing time is very limited, as it can help to find important defects quickly and adapt to changing requirements or situations . The other options are not good test techniques to use when under severe time pressure. Option B is not a good option, because structure-based testing is a type of technique that requires analysing the code or design of the software under test to measure aspects such as code coverage, complexity, or coupling . Structure-based testing can be time-consuming and may not be feasible or effective when testing time is very limited . Option C is not a good option, because specification-based testing is a type of technique that requires analysing the requirements or specifications of the software under test to design test cases that verify the expected functionality or behaviour . Specification-based testing can also be time-consuming and may not be suitable or efficient when testing time is very limited . Option D is not a good option, because use case testing is a type of technique that requires analysing the use cases or scenarios of the software under test to design test cases that verify the end-to-end user interactions and outcomes . Use case testing can also be time-consuming and may not be relevant or effective when testing time is very limited .

QUESTION 25
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

Correct Answer: B
Section:
Explanation:
The test execution schedule would not be expected to see on an incident report from test execution, because it is not related to the incident itself, but rather to the overall plan and progress of testing . An incident report is a document that records any event that deviates from the expected or desired behaviour of the software under test . The other options are information that would be expected to see on an incident report from test execution. Option A is expected, because the version(s) of the software under test can help to identify and reproduce the incident . Option C is expected, because the expected results and actual results can help to describe and demonstrate the incident . Option D is expected, because the precise steps to reproduce the problem can help to analyse and resolve the incident .

QUESTION 26
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

Correct Answer: C
Section:
Explanation:
Event B from S2 is an invalid transition, because it does not exist in the state table. The state table shows the possible transitions between states based on events. For each state, there is a corresponding row that indicates the next state for each event. For example, for state S1, the next state for event A is S2, for event B is S3, and so on.However, for state S2, there is no next state for event B, which means that this transition is invalid

QUESTION 27
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

Correct Answer: D
Section:
Explanation:
Variables that are not used or improperly declared are defects that are more likely to be found by a static analysis tool than by other testing techniques, because they are syntactic or structural errors in the code that can be detected without executing the software13.A static analysis tool can analyse the code or design of the software under test and identify errors such as missing declarations, undefined variables, unused variables, or unreachable code13. The other options are not defects that are more likely to be found by a static analysis tool than by other testing techniques.Option A is not a defect that can be found by a static analysis tool, but rather by a requirements review or specification-based testing technique1.Option B is not a defect that can be found by a static analysis tool, but rather by a structure-based testing technique or a coverage measurement tool1.Option C is not a defect that can be found by a static analysis tool, but rather by a dynamic analysis tool or a monitoring tool1.

QUESTION 28
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

Correct Answer: D
Section:
Explanation:
Checking that at least 50% of decisions have been exercised by a test case suite is not a valid use of decision coverage, because it does not meet the minimum criterion of decision coverage, which is to exercise all possible outcomes of each decision in the software under test1.Decision coverage is a technique that measures how much of the logic or branching of the software under test has been exercised by the test cases1. The other options are valid uses of decision coverage.Option A is a valid use of decision coverage, because it can check that all decisions have been exercised in a single program1.Option B is a valid use of decision coverage, because it can check that all decisions have been exercised in a business process1.Option C is a valid use of decision coverage, because it can check that all calls from one program module to another have been made correctly1.

QUESTION 29
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

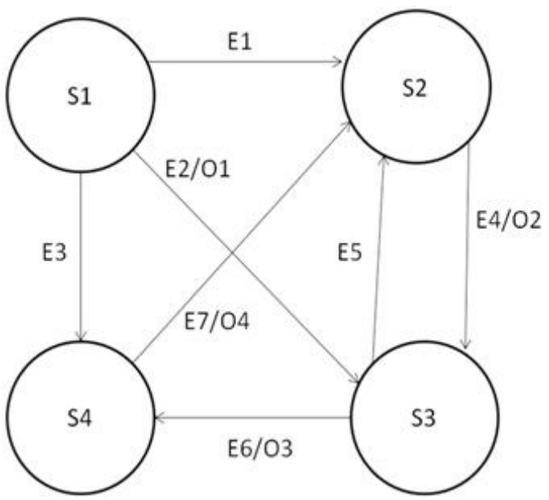
Correct Answer: C
Section:
Explanation:
Decision testing is a white box (structure-based) test case design technique1.White box (structure-based) test case design techniques are techniques that use the structure or logic of the source code as a test basis to derive test cases and measure test coverage1.Decision testing is a technique that uses decisions (or branches) in the source code as a test basis to measure the coverage achieved by a test suite1.A decision (or branch) is a point in the source code where the control flow can take two or more alternative paths based on a condition1.Decision testing requires every decision (or branch) in the source code to take both true and false outcomes at least once by a test suite1. Therefore, decision testing is a white box (structure-based) test case design technique.

QUESTION 30
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.

Correct Answer: D
Section:
Explanation:
Test cases derived from the knowledge of the testers and the knowledge of testers, developers and users are used to drive testing apply to experience-based techniques1.Experience-based techniques are techniques that use the skill, intuition, and experience of testers, developers, and users to derive test cases, using error guessing and exploratory testing1.Error guessing is a technique that uses common sense and previous experience to guess where defects might occur in a system1.Exploratory testing is an approach that involves simultaneous learning, test design, and test execution1.Experience-based techniques are typically used when there is insufficient information or time to apply other more formal techniques1. Therefore, test cases derived from the knowledge of the testers and the knowledge of testers, developers and users are used to drive testing apply to experience-based techniques.

QUESTION 31
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

Correct Answer: D
Section:
Explanation:
S2 and O2 are the finishing state and the output(s) from the test case1. The question asks about a test case that starts at S1 and triggers 4 events in sequence: E1, E4, E5, E7. The question refers to a state diagram that shows the logic of a system with six states (S1, S2, S3, S4, S5, S6) and seven events (E1, E2, E3, E4, E5, E6, E7). The state diagram also shows four outputs (O1, O2, O3, O4) that are produced by some events. The state diagram is shown below:
!state diagram
To answer this question, we need to follow the path of the test case on the state diagram and observe the state transitions and outputs that occur. The path of the test case is:
Start at S1
Trigger E1
Transition to S2
Trigger E4
Transition to S3
Trigger E5
Transition to S2
Output O2
Trigger E7
Transition to S6
Therefore, the finishing state of the test case is S2 and the output(s) from the test case is O2. Hence, D is the correct answer.
Topic 3, Exam Pool C

QUESTION 32
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

Correct Answer: D
Section:
Explanation:
2400, 1000, 2301 are test case inputs that each fall into a different equivalence partition1.Equivalence partitioning is a black box test design technique that divides the input domain of a system into partitions of equivalent data2.Each partition should contain data that is expected to be treated the same by the system2.Therefore, only one test case input from each partition is needed to test the system's behaviour2. In this question, the input domain of the system is the time of travel, which can be divided into three partitions based on the fare rules:
Peak-time tariff: This partition contains all the times between 0600 and 1000 (inclusive), which are subject to a higher fare.
Off-peak tariff: This partition contains all the times between 1001 and 2259 (inclusive), which are subject to a lower fare.
No service: This partition contains all the times between 2300 and 0559 (inclusive), which are not valid for travel.
Therefore, to test the system's behaviour for each partition, we need to choose one test case input from each partition. Among the options given in this question, only D contains one test case input from each partition. 2400 belongs to the no service partition, 1000 belongs to the peak-time tariff partition, and 2301 belongs to the off-peak tariff partition. Therefore, D is the correct answer.

QUESTION 33
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

Correct Answer: D
Section:
Explanation:
Boundary value analysis and decision table testing are the best choice of one black box technique and one white box technique for test case design2.A black box technique is a technique that uses the external behaviour or specification of a system as a test basis to derive test cases without referring to its internal structure or logic2.A white box technique is a technique that uses the internal structure or logic of a system as a test basis to derive test cases and measure test coverage2. In this question, the system has complex business rules and many branches in the structure of its code components. Therefore, we need to choose a black box technique that can handle complex business rules and a white box technique that can cover many branches.
Boundary value analysis is a black box technique that tests the values at or near the boundaries of an equivalence partition2.Boundary value analysis can handle complex business rules by identifying the boundary conditions that may cause errors or exceptions in the system's behaviour2.
Decision table testing is a white box technique that tests all possible combinations of conditions and actions in a decision logic2.Decision table testing can cover many branches by creating a table that shows how each condition affects each action and then deriving test cases from each row of the table2.
Therefore, boundary value analysis and decision table testing are the best choice of one black box technique and one white box technique for test case design.

QUESTION 34
Which of the following is a Black Box test design technique?

A. Decision Coverage
B. Error Guessing
C. Statement Coverage
D. Equivalence Partitioning

Correct Answer: D
Section:
Explanation:
Equivalence partitioning is a black box test design technique2.A black box technique is a technique that uses the external behaviour or specification of a system as a test basis to derive test cases without referring to its internal structure or logic2.Equivalence partitioning is a technique that divides the input domain of a system into partitions of equivalent data2.Each partition should contain data that is expected to be treated the same by the system2.Therefore, only one test case input from each partition is needed to test the system's behaviour2. Therefore, equivalence partitioning is a black box test design technique.

QUESTION 35
Which of the following is a white-box test technique?

A. Decision table testing
B. Exploratory testing
C. Statement testing
D. Error guessing

Correct Answer: C
Section:
Explanation:

Statement testing is a white box test technique2.A white box technique is a technique that uses the internal structure or logic of a system as a test basis to derive test cases and measure test coverage2.Statement testing is a technique that uses executable statements in the source code as a test basis to measure the coverage achieved by a test suite2.A statement is an instruction or command that performs an action or calculation in the source code2.Statement testing requires every statement in the source code to be executed at least once by a test suite2. Therefore, statement testing is a white box test technique.

QUESTION 36
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

Correct Answer: D
Section:
Explanation:
The testability of requirements is evaluated during the Test Analysis and Design stage of the fundamental test process3. The fundamental test process consists of five main activities:
Test Planning and Control: This activity involves defining the scope, objectives, approach, and resources for testing, as well as monitoring and controlling the test progress and results3.
Test Analysis and Design: This activity involves analyzing the test basis (such as requirements, design, etc.) and designing the test cases, procedures, and environment3.The testability of requirements is evaluated during this activity, as it involves checking the clarity, completeness, consistency, and testability of the requirements3.
Test Implementation and Execution: This activity involves implementing the test cases and procedures, preparing the test environment and data, executing the test cases, and logging the test results and incidents3.
Evaluating Exit Criteria and Reporting: This activity involves evaluating the test results against the exit criteria (such as coverage, defect rate, etc.) and reporting the test status and outcomes to the stakeholders3.
Test Closure Activities: This activity involves finalizing the test documentation, archiving the test artifacts, evaluating the test process and lessons learned, and releasing the test resources

QUESTION 37
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

Correct Answer: C
Section:
Explanation:
A test procedure specification is a document that gives the precise steps needed in order to execute a test2.A test procedure specification defines how to run each test case or group of test cases in a specific order, including any preconditions, postconditions, inputs, outputs, actions, verifications, and expected results2.A test procedure specification can also include information about the test environment, tools, data, roles, responsibilities, risks, etc2. Therefore, a test procedure specification is a document that gives the precise steps needed in order to execute a test.

QUESTION 38
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

Correct Answer: A

Section:
Explanation:
Preventing defects from being introduced into the code is not a valid objective of testing, because testing can only detect defects that are already present in the code, not prevent them from happening. Testing can help prevent future defects by providing feedback to developers and other stakeholders, but it cannot guarantee that no defects will be introduced.Preventing defects is more related to quality assurance activities, such as reviews, inspections, and standards12.

QUESTION 39
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

Correct Answer: A
Section:
Explanation:
Having an independent test function in an organisation can improve defect finding during reviews and testing, because independent testers can provide a different perspective and have less bias than developers or other stakeholders.Independent testers can also focus on testing activities without being influenced by development pressures or deadlines12. The other options are not valid reasons for having an independent test function. Option B is more related to code reviews, which can be done by developers themselves or by peers. Option C is not beneficial, because communication between developers and testers is essential for effective testing and defect resolution.Option D is not directly related to the test function, but to the test management and reporting processes12.

QUESTION 40
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

Correct Answer: D
Section:
Explanation:
Testing support tools are software tools that assist testers in performing testing activities, such as test management, test design, test execution, test evaluation, and defect management1.According to the ISTQB syllabus, some typical targets of testing support tools are to automate activities that require significant resources when done manually, automate activities that cannot be executed manually, and automate repetitive tasks1.Automating repetitive inspections is not a typical target of testing support tools, as inspections are static testing techniques that involve human review and analysis of software artifacts1.

QUESTION 41
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

Correct Answer: A
Section:
Explanation:

Screen-dialog flows are sequences of screens and dialogs that represent the user interface and interaction of a system.Use case testing is a technique that uses use cases as a test basis to derive test cases1.A use case is a description of interactions between actors and a system to achieve a goal1. Use case testing is the most effective technique in testing screen-dialog flows, as it can capture the user requirements, scenarios, and expected outcomes of the system.Boundary value testing is a technique that uses boundary values as a test basis to derive test cases1.A boundary value is an input value or output value on the edge of an equivalence partition or at the smallest or largest value of a range1. Boundary value testing is not the most effective technique in testing screen-dialog flows, as it is more suitable for testing numerical inputs and outputs.Statement testing and coverage is a technique that uses statements in the source code as a test basis to measure the coverage achieved by a test suite1.A statement is a minimal executable unit of source code1. Statement testing and coverage is not the most effective technique in testing screen-dialog flows, as it is more suitable for testing the internal logic and structure of the code.State transition testing is a technique that uses state transition diagrams as a test basis to derive test cases1.A state transition diagram shows the states of a system and the transitions between them triggered by events or conditions1. State transition testing is not the most effective technique in testing screen-dialog flows, as it is more suitable for testing systems that have complex or dynamic behavior based on different states.

QUESTION 42
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

Correct Answer: C
Section:
Explanation:
According to the IEEE 829 Standard for Software Test Documentation, an incident report should contain the following information:
Identifier: A unique identifier for the incident report
Summary: A brief summary of the incident
Incident description: A description of the incident, including:
Date: The date when the incident was observed
Author: The name of the person who reported the incident
Source: The software or system lifecycle process in which the incident was observed
Test case: The identification of the test case that caused the incident
Execution phase: The phase of test execution when the incident was observed
Environment: The hardware and software environment in which the incident was observed
Description: A description of the anomaly to enable reproduction of the incident
Expected result: The expected result of the test case
Actual result: The actual result of the test case
Reproducibility: An indication of whether the incident can be reproduced or not
Impact analysis: An analysis of the impact of the incident on other aspects of the software or system
Incident resolution: A description of how the incident was resolved, including:
Resolution date: The date when the incident was resolved
Resolver: The name of the person who resolved the incident
Resolution summary: A brief summary of how the incident was resolved
Status: The current status of the incident (e.g., open, closed, deferred)
Classification information: A classification of the cause and effect of the incident for metrics and reporting purposes
Therefore, among the options given in this question, only (i), (iii), and (iv) are part of an incident report based on IEEE 829.

QUESTION 43
''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.

Correct Answer: C
Section:
Explanation:
Experience-based test design techniques are techniques that use the skill, intuition, and experience of testers to derive test cases, using error guessing and exploratory testing1.Error guessing is a technique that uses common sense and previous experience to guess where defects might occur in a system1.Exploratory testing is an approach that involves simultaneous learning, test design, and test execution1.Experience-based test design techniques are typically used when there is insufficient information or time to apply other more formal techniques1. They do not use decision tables, identify the structure of the system or software, or establish traceability from test conditions back to the specifications and requirements.

QUESTION 44
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

Correct Answer: A
Section:
Explanation:
The effective ways to improve the communication and relationship between testers and others are A. a and b. Communicating factual information in a constructive way and trying to understand how the other person feels and why they react the way they do are both important skills for testers to have. Testers often have to report defects and problems that may not be well received by developers or users. Therefore, testers should communicate in a clear, objective, and respectful manner, avoiding personal attacks or blame. Testers should also empathize with the other person's perspective and emotions, and try to resolve any conflicts or misunderstandings in a positive way. A detailed explanation of communication skills for testers can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 107-108.

QUESTION 45
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

Correct Answer: C
Section:
Explanation:
A valid reason for writing test cases based on experience and intuition is C. Tests based on experience and intuition can supplement formal techniques. Experience and intuition are valuable sources of test ideas that can help

testers to identify potential risks, errors, or scenarios that may not be covered by formal techniques. Formal techniques are systematic methods that use rules or algorithms to derive test cases from the test basis (such as specifications, code, etc.). Formal techniques include black-box techniques (such as equivalence partitioning, boundary value analysis, etc.) and white-box techniques (such as statement coverage, decision coverage, etc.). Formal techniques can provide a high level of coverage and consistency, but they may not be able to capture all possible situations or behaviors of the system or component under test. Therefore, tests based on experience and intuition can supplement formal techniques by adding more diversity and creativity to the test cases. A detailed explanation of experience-based testing techniques can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 74-76.

QUESTION 46
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

Correct Answer: D
Section:
Explanation:
A white box testing design technique is D. Statement testing. Statement testing is a white box testing technique that is based on an analysis of the structure of the component or system. Statement testing aims to cover every executable statement in the code at least once by the test cases. Statement testing can help to detect syntax errors, logic errors, or unreachable code in the system or component under test. Statement testing is usually performed at lower levels of testing, such as unit testing or component testing. A detailed explanation of statement testing can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 68-69.

QUESTION 47
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

Correct Answer: C
Section:
Explanation:
Independent testing is important because C. Independent testers can verify assumptions made during specification and implementation of the system. Independent testing refers to testing performed by testers who are not involved in the development or use of the system or component under test. Independent testers can provide an unbiased and objective view of the quality of the system or component under test. Independent testers can also verify assumptions made by developers or users during specification and implementation of the system or component under test. Assumptions are statements that are believed to be true without proof or verification. Assumptions can lead to errors or defects if they are incorrect or inconsistent with reality. Independent testers can help to identify and validate assumptions by using different sources of information, such as requirements, specifications, standards, regulations, etc., and by applying different testing techniques, such as black-box techniques, white-box techniques, etc. A detailed explanation of independent testing can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 9-10.

QUESTION 48
Which of the following errors CANNOT be found with structure-based testing techniques?

A. Memory is leaking
B. Features are only partially implemented
C. Data structures that are used before initialization
D. Division by zero

Correct Answer: A

Section:
Explanation:
Memory leaks are errors that occur when a program does not release memory that it has allocated, causing the system to run out of memory and slow down or crash. Memory leaks cannot be detected by structure-based testing techniques, which are based on the code structure and logic. Structure-based testing techniques can only find errors that are related to the control flow, data flow, or logic of the program. For example, they can find errors such as features that are only partially implemented, data structures that are used before initialization, or division by zero. To detect memory leaks, you need dynamic analysis tools that monitor the memory usage of the program during execution. You can find more information about structure-based testing techniques and dynamic analysis tools inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Sections 4.2 and 4.31.

QUESTION 49
Which of the following is correct?

A. Intrusive test tools are tools that do not exhibit the probe-effect
B. Testing tools can be used by both developers and testers
C. Use of testing tools is effective only when done as part of a test automation system
D. Testing tools allow developers do testing Use of such tools changes the role of the test team

Correct Answer: B
Section:
Explanation:
Testing tools can be used by both developers and testers for different purposes and at different stages of the software development life cycle. For example, developers can use tools such as unit testing frameworks, code coverage tools, debugging tools, static analysis tools, etc., to improve the quality of their code and find defects early. Testers can use tools such as test management tools, test design tools, test execution tools, test data preparation tools, performance testing tools, etc., to support their testing activities and increase their efficiency and effectiveness. The use of testing tools does not necessarily imply test automation, which is the use of software to perform or support test activities that would otherwise require manual intervention. Test automation is a complex and costly process that requires careful planning, design, implementation, maintenance, and evaluation. The use of testing tools also does not change the role of the test team, which is still responsible for defining the test strategy, designing the test cases, analyzing the test results, reporting the defects, etc. You can find more information about testing tools and test automation inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 61.

QUESTION 50
The following sentences refer to the Standard for Software Test Documentation' specification (IEEE 829). Which sentence is correct?

A. The key to high quality test documentation regimes is strict adherence to this standard
B. Any deviation from this standard should be approved by management, marketing & development
C. This test plan outline is relevant for military projects For consumer market projects there is a different specification with fewer items
D. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization

Correct Answer: D
Section:
Explanation:
The 'Standard for Software Test Documentation' specification (IEEE 829) is a standard that defines a set of documents that can be used to document the test process and its outcomes. The standard provides an outline for each document, specifying its purpose, content, and format. However, the standard does not prescribe how to apply it in different contexts or projects. It is up to each organization or project to decide how to adapt the standard to their specific needs and situation. Therefore, the standard is not a rigid or mandatory requirement that must be followed strictly by all testers. Rather, it is a flexible and adaptable guideline that can be used as a reference or a starting point for creating test documentation regimes. You can find more information about IEEE 829 and test documentation inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 51.

QUESTION 51
A software company adopts the V-model as their development life cycle. Which of the following contains roles of a tester in this company?

A. Decide what should be automated, to what degree, and how.
B. Review test plans and set up test environments.
C. Coordinate the test strategy with the project managers

D. Introduce suitable metrics to measure the testing progress

Correct Answer: C
Section:
Explanation:
The V-model is a development life cycle model that shows the relationship between each phase of development and its corresponding phase of testing. In this model, each level of testing (unit testing, integration testing, system testing, acceptance testing) has a corresponding level of development (component design, component integration, system design, requirements analysis). The model also shows that testing activities should start as early as possible in the development process and that each level of testing should be planned and designed in parallel with its corresponding level of development. Therefore, one of the roles of a tester in a software company that adopts the V-model is to coordinate the test strategy with the project managers who are responsible for planning and managing each phase of development. This role involves defining the scope, objectives, approach, resources, schedule, risks, and deliverables of each level of testing in alignment with the development plan and the project requirements. You can find more information about the V-model and test planning inSoftware Testing Foundations: A Study Guide for the Certified Tester Exam, Chapter 22.

QUESTION 52
Which of the following is an appropriate reason for maintenance testing?

A. Bugs found in the field after upgrading the operation system
B. Bugs found during system testing
C. Bugs found during unit testing
D. Bugs found during integration testing

Correct Answer: A
Section:
Explanation:
Maintenance testing is a type of testing that is performed after a software product has been delivered and deployed to ensure that it still meets its requirements and functions correctly after changes have been made to it or to its environment. Changes can include corrective changes (fixing defects), adaptive changes (adapting to new platforms or environments), perfective changes (improving performance or usability), or preventive changes (avoiding potential problems). One of the appropriate reasons for maintenance testing is to verify that the software product works as expected after upgrading the operating system, which is an example of an adaptive change. Other reasons for maintenance testing can include verifying that the software product works as expected after fixing defects, adding new features, improving performance, or preventing future issues. You can find more information about maintenance testing inSoftware Testing Foundations: A Study Guide for the Certified Tester Exam, Chapter 12.

QUESTION 53
Once a bug is fixed, it should be retested. What is the term used to define this type of testing?

A. Reliability Testing
B. Confirmation Testing
C. Maintainability Testing
D. Regression Testing

Correct Answer: B
Section:
Explanation:
Confirmation testing, also known as re-testing, is the process of testing a defect that has been fixed to verify that it has been resolved and does not affect other parts of the system. Confirmation testing is usually done by the same tester who reported the defect and using the same test case that revealed the defect. You can find more information about confirmation testing inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 3, Section 3.21.

QUESTION 54
What is the difference between system integration testing and acceptance testing?

A. System integration testing is testing non-functional requirements Acceptance testing concentrates on the functionality of the system

B. System integration testing is executed by the developers. Acceptance testing is done by the customer
C. System integration testing verifies that a system interfaces correctly with other systems. Acceptance testing verifies compliance to requirements
D. System integration testing verifies compliance to requirements Acceptance testing verifies correct interaction with other systems existing in the user's environment

Correct Answer: C
Section:
Explanation:
System integration testing and acceptance testing are two different levels of testing that have different objectives and stakeholders. System integration testing is the process of testing the interactions and interfaces between different components or systems that form a larger system. System integration testing verifies that the system meets its functional and non-functional requirements and works as expected in its intended environment. System integration testing is usually done by testers or developers who have access to the system architecture and design specifications. Acceptance testing is the process of testing the system by the end users or customers to determine if it satisfies their needs and expectations and is ready for deployment or delivery. Acceptance testing verifies that the system complies with the user requirements and business processes and provides value to the stakeholders. Acceptance testing is usually done by users or customers who have access to the user requirements and business scenarios. You can find more information about system integration testing and acceptance testing inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 2, Sections 2.2 and 2.31.

QUESTION 55
Which of the following is NOT an example of a common test metric?

A. Percentage of work done in test environment creation
B. Average number of expected defects per requirement
C. Number of test cases run
D. Deviation from test milestone dates

Correct Answer: B
Section:
Explanation:
Test metrics are quantitative measures that are used to monitor, control, and improve the test process and its outcomes. Test metrics can be collected at different levels of testing (test case, test suite, test project, etc.) and can be used for different purposes (planning, estimation, execution, evaluation, etc.). Some examples of common test metrics are:
Percentage of work done in test environment creation: This metric indicates how much effort has been spent on setting up and maintaining the test environment, which includes hardware, software, network, data, tools, etc., that are required for conducting the test activities.
Number of test cases run: This metric indicates how many test cases have been executed during a given period or phase of testing.
Deviation from test milestone dates: This metric indicates how much delay or ahead of schedule the test activities are compared to the planned dates.
Defect density: This metric indicates how many defects have been found per unit of size or functionality of the system under test.
Average number of expected defects per requirement is not a common test metric because it is not easy to estimate or measure how many defects are likely to be found for each requirement. Moreover, this metric does not provide useful information for improving the test process or evaluating the test results. You can find more information about test metrics inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 5, Section 5.41.

QUESTION 56
Which of the following is NOT a deciding factor in determining the extent of testing required?

A. Budget to do testing
B. A particular tester involved in testing
C. Level of risk of the product or features
D. Time available to do testing

Correct Answer: B
Section:
Explanation:
The extent of testing required for a software product or feature depends on several factors that influence the level of quality and risk involved in delivering or deploying it. Some of these factors are:
Budget to do testing: This factor indicates how much money is available or allocated for conducting the test activities, which affects the scope, depth, duration, and resources of testing.

Time available to do testing: This factor indicates how much time is available or allocated for conducting the test activities, which affects the schedule, frequency, speed, and coverage of testing.
Level of risk of the product or feature: This factor indicates how much impact or harm a failure or defect in the product or feature can cause to the users, customers, stakeholders, or environment, which affects the priority, intensity, complexity, and rigor of testing.
Complexity of the product or feature: This factor indicates how difficult or challenging it is to understand, design, implement, or maintain the product or feature, which affects the effort, skill, technique, and tool required for testing.
A particular tester involved in testing is not a deciding factor in determining the extent of testing required because it does not affect the quality or risk level of the product or feature being tested. Rather, it is an outcome or consequence of deciding how much testing is needed based on other factors such as budget, time, risk, and complexity. You can find more information about determining the extent of testing inSoftware Testing Foundations: A Study Guide for the Certified Tester Exam, Chapter 2, Section 2.12.

QUESTION 57
What does the term Pesticide paradox' refer to?

A. The phenomena where a piece of code that has a lot of bugs is likely to have more hidden, yet unfound
B. The decreasing efficiency of debugging when done in code that has many bugs
C. Reduced effectiveness of test cases that are repeated and focused on the same scenarios
D. The redundancy of testing the same objects in both black and white box techniques

Correct Answer: C
Section:
Explanation:
The term 'Pesticide paradox' refers to the phenomenon where the effectiveness of test cases that are repeated and focused on the same scenarios decreases over time because they tend to find the same defects or no defects at all. This is because the system under test becomes more resistant or immune to the existing test cases, just like pests become more resistant or immune to pesticides over time. To overcome the pesticide paradox, test cases should be regularly reviewed and updated to cover new or changed requirements, scenarios, risks, or defects. Test cases should also be designed to cover different aspects and perspectives of the system under test, such as functionality, usability, performance, security, etc. You can find more information about the pesticide paradox inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Section 4.11.

QUESTION 58
Which of the following test techniques is structure-based?

A. Control flow testing
B. Use case testing
C. State transition testing
D. Decision table testing

Correct Answer: A
Section:
Explanation:
Test techniques are methods or procedures that can be used to design, execute, or evaluate test cases. Test techniques can be classified into two categories: specification-based and structure-based. Specification-based test techniques, also known as black-box test techniques, are based on the requirements, specifications, or expectations of the system under test. They do not require any knowledge of the internal structure or implementation of the system. Some examples of specification-based test techniques are use case testing, state transition testing, decision table testing, etc. Structure-based test techniques, also known as white-box test techniques, are based on the code, architecture, or design of the system under test. They require some knowledge of the internal structure or implementation of the system. Some examples of structure-based test techniques are control flow testing, data flow testing, branch testing, statement testing, etc. You can find more information about test techniques inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 41.

QUESTION 59
Which of the following test types is a part of the V-Model?

A. Black-box testing
B. White-box testing
C. Experience-based testing

D. Component testing

Correct Answer: D
Section:
Explanation:
Component testing is a part of the V-Model, which is a development life cycle model that shows the relationship between each phase of development and its corresponding phase of testing. In this model, each level of testing (unit testing, integration testing, system testing, acceptance testing) has a corresponding level of development (component design, component integration, system design, requirements analysis). Component testing is the process of testing individual components or units of software in isolation from other components or systems. Component testing verifies that the components meet their functional and non-functional requirements and work as expected in their intended environment. Component testing is usually done by developers who have access to the component design and code specifications. You can find more information about component testing and the V-Model inSoftware Testing Foundations: A Study Guide for the Certified Tester Exam, Chapter 22.

QUESTION 60
Which of the following statements is correct?

A. Pair programming is done with developer and tester pairing together
B. Pair programming is an alternative term for code inspection.
C. Pair programming is used usually in waterfall model
D. Pair programming is, among other things, an informal review method.

Correct Answer: D
Section:
Explanation:
Pair programming is a software development technique where two programmers work together on the same code at the same workstation. One programmer, called the driver, writes the code while the other programmer, called the navigator, reviews the code and provides feedback and suggestions. Pair programming is not only a coding technique but also an informal review method because it involves constant communication and collaboration between the programmers who check each other's work and share their knowledge and skills. Pair programming can improve the quality and productivity of software development by reducing defects, increasing code readability, enhancing learning opportunities, and fostering creativity and innovation. Pair programming is usually used in agile methods such as Extreme Programming (XP), Scrum, or Kanban. It is not an alternative term for code inspection, which is a formal review method where a group of reviewers examine a piece of code in a structured and systematic way following predefined rules and roles. It is also not done with developer and tester pairing together because pair programming requires both programmers to have similar levels of expertise and familiarity with the code they are working on. You can find more information about pair programming inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 31.

QUESTION 61
Which of the following BEST describes checklist-based testing?

A. An approach to testing whereby the testers dynamically design and execute tests based on their knowledge, exploration of the test item and the results of previous tests.
B. An experience-based test technique whereby the experienced tester uses a high-level list of items to be noted, checked or remembered, or a set of rules or criteria against which a product has to be verified.
C. A procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.
D. A test design technique which ensures that test cases are checked for consistency and completeness against an organisation's list of formatting rules and best practices.

Correct Answer: B
Section:
Explanation:
Checklist-based testing is an experience-based test technique whereby the experienced tester uses a high-level list of items to be noted, checked or remembered, or a set of rules or criteria against which a product has to be verified. Checklist-based testing can help the tester to focus on important aspects of the test object and to ensure that nothing is overlooked or forgotten. Checklist-based testing can also help the tester to communicate and report the test results and coverage more effectively.

QUESTION 62
Which of the following is a key difference between black box and white box test design techniques?

A. Black box techniques use software code to derive test cases, white box techniques do not.
B. White box techniques use functional design specifications to derive test cases, black box techniques do not.
C. White box techniques can measure the extent of code coverage, black box techniques can not.
D. White box techniques derive test cases from models of the software, black box techniques do not.

Correct Answer: C
Section:
Explanation:
A key difference between black box and white box test design techniques is that white box techniques can measure the extent of code coverage, while black box techniques cannot. Code coverage is a measure of how much of the code of a system or component has been executed by a test or a set of tests. White box techniques use software code to derive test cases that cover specific aspects of the code, such as statements, branches, paths, or data flows. White box techniques can use tools or methods to measure the code coverage achieved by the test cases and identify any gaps or redundancies. Black box techniques use functional design specifications to derive test cases that cover specific aspects of the functionality, behavior, or quality of the system or component, such as inputs, outputs, equivalence classes, boundary values, states, transitions, etc. Black box techniques do not use or access the code of the system or component, and therefore cannot measure the code coverage.

QUESTION 63
A car insurance policy has 3 rates of insurance depending on the age of the driver. For drivers aged between 17 and 25 inclusive they are charged at rate A, drivers aged between 26 and 50 inclusive are charged at rate B and those drivers aged over 50 are charged at rate C.
You are designing test cases, which of the following three ages would test all valid equivalence partitions and therefore test rate A, B and C?

A. 26, 45, 50.
B. 10, 21, 55.
C. 20, 35, 65.
D. 17, 25, 50.

Correct Answer: C
Section:
Explanation:
The three ages that would test all valid equivalence partitions and therefore test rate A, B and C are 20, 35 and 65. Equivalence partitioning is a technique to divide a set of possible inputs or outputs into classes that are expected to behave similarly or produce similar results. For each equivalence class, only one test case is required to represent the whole class. In this case, we can identify the following equivalence classes for the age of the driver:
Valid age between 17 and 25 inclusive (rate A)
Valid age between 26 and 50 inclusive (rate B)
Valid age over 50 (rate C)
Invalid age below 17
Invalid age above 100
Therefore, three test cases are required to test all valid equivalence partitions, and any value within each class can be used as a representative value. For example, 20 for rate A, 35 for rate B, and 65 for rate C.

QUESTION 64
A student needs to score at least 50 points to pass. If they score at least 100 points they will achieve a merit and if they score at least 150 points they will achieve a distinction.
Which two values are in the same partition?

A. 45 and 55.
B. 55 and 120.
C. 50 and 60.
D. 45 and 170.

Correct Answer: C
Section:

Explanation:
According to the syllabus, equivalence partitioning is a technique that divides the input data into partitions that are expected to be processed in the same way by the system under test. The partitions should be disjoint, meaning that they do not overlap. The answer C is correct because 50 and 60 are in the same partition of valid inputs that will pass the test. The other answers are incorrect because they contain values from different partitions.

QUESTION 65
In foundation level syllabus you will find the main basic principles of testing, Which of the following sentences describes one of these basic principles?

A. Complete testing of software is attainable if you have enough resources and test tools
B. For a software system, it is not possible under normal conditions, to test all input and output combinations.
C. A goal of testing is to show that the software is defect free
D. With automated testing you can make statements with more confidence about the quality of a product than with manual testing.

Correct Answer: B
Section:
Explanation:
One of the basic principles of testing is that for a software system, it is not possible under normal conditions to test all input and output combinations because they are too many and too complex to cover within a reasonable time and budget. Therefore, testers have to select a subset of inputs and outputs that are representative and relevant for testing based on criteria such as requirements, risks, priorities, etc. This principle implies that testing cannot prove that a software system is defect-free or has a certain level of quality because there might be some untested inputs or outputs that could reveal defects or failures. Testing can only provide information about the quality and risk level of a software system based on the observed behavior and results under certain conditions and assumptions. You can find more information about the basic principles of testing inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 11.

QUESTION 66
Which of the following is an example of black-box dynamic testing?

A. Code inspection
B. Checking memory leaks for a program by executing it
C. Functional Testing
D. Coverage analysis

Correct Answer: C
Section:
Explanation:
Functional testing is an example of black-box dynamic testing. Functional testing is the process of testing the functionality or behavior of a software system based on its requirements or specifications. Functional testing does not require any knowledge of the internal structure or implementation of the system under test; it only focuses on what the system does rather than how it does it. Functional testing is also an example of dynamic testing because it involves executing the software system with selected inputs and observing its outputs and effects on the environment. Dynamic testing is different from static testing, which involves examining the software system without executing it, such as code inspection, static analysis, etc. You can find more information about functional testing and dynamic testing inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 21.

QUESTION 67
Where and by whom is Beta testing normally performed?

A. By customers or potential customers at their own locations
B. By an independent test team at the developing organization's location
C. At the developing organization's site, but not by the developing team
D. By customers or potential customers at the developing organization's site

Correct Answer: A
Section:

Explanation:
Beta testing is a type of acceptance testing that is normally performed by customers or potential customers at their own locations. Beta testing is done after the software product has passed the internal testing and quality assurance processes and is ready for release or deployment. Beta testing allows the customers or users to evaluate the software product in their real environment and provide feedback and suggestions for improvement. Beta testing can also help identify defects, compatibility issues, usability problems, or performance bottlenecks that might not have been detected during the internal testing. You can find more information about beta testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 2, Section 2.3.

QUESTION 68
Which of the following is NOT an objective of testing?

A. Finding defects
B. Providing information for decision-making
C. Analyzing and removing the cause of failures
D. Gaining confidence about the level of quality of the software

Correct Answer: C
Section:
Explanation:
Testing has several objectives that aim to provide information and confidence about the quality and risk level of the software product and to support decision-making and improvement processes. Some of these objectives are:
Finding defects: Testing is the process of finding defects or failures in the software product that do not meet its requirements or expectations.
Providing information for decision-making: Testing is the process of providing information about the quality and risk level of the software product based on the observed behavior and results under certain conditions and assumptions. This information can help stakeholders make informed decisions about releasing, deploying, accepting, or improving the software product.
Gaining confidence about the level of quality of the software: Testing is the process of gaining confidence that the software product meets its requirements and expectations and provides value to the users and customers.
Analyzing and removing the cause of failures: Testing is not only the process of finding defects but also the process of analyzing and removing the cause of failures that lead to defects. This objective is related to debugging, which is a development activity that involves locating, diagnosing, and fixing errors in the code or design of the software product.
You can find more information about the objectives of testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 1, Section 1.2.

QUESTION 69
Which statement about use case testing is true?

A. The test cases are designed to find defects in the data flow.
B. The test cases are designed to find defects in the process flow
C. The test cases are designed to be used by real users, not by professional testers
D. The test cases are always designed by customers or end users

Correct Answer: B
Section:
Explanation:
Use case testing is a specification-based test technique that is based on the use cases or scenarios that describe how users interact with the system under test. Use case testing aims to find defects in the process flow or functionality of the system under test by verifying that it behaves as expected according to the use cases or scenarios. Use case testing does not require any knowledge of the data flow or structure of the system under test; it only focuses on what the system does rather than how it does it. Use case testing can be done by professional testers or real users, depending on the level and purpose of testing. Use case testing can also be designed by customers or end users, developers, testers, analysts, or anyone who has access to the use case specifications or user requirements. You can find more information about use case testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 4, Section 4.2.

QUESTION 70
When testing a mission critical system a high coverage should be achieved. Which of the following techniques should be implemented as a structural based coverage technique in order to achieve highest coverage?

A. multiple condition coverage

B. decision table
C. use case testing
D. statement coverage

Correct Answer: A
Section:
Explanation:
Multiple condition coverage is a structure-based test technique that aims to achieve a high level of coverage by testing all possible combinations of conditions and outcomes in each decision point of the code or design of the system under test. Multiple condition coverage verifies that each condition in a decision point has an effect on the outcome and that there are no hidden defects or logical errors in the code or design. Multiple condition coverage requires some knowledge of the internal structure or implementation of the system under test; it focuses on how the system does what it does rather than what it does. Multiple condition coverage is one of the most rigorous and complex structure-based test techniques because it can generate a large number of test cases for each decision point, especially if there are many conditions involved. You can find more information about multiple condition coverage in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 4, Section 4.3.

QUESTION 71
Which of the following is NOT an example of a typical risk-based testing activity?

A. The evaluation of a risk-management tools to decide which tool to use for future projects
B. The focus of testing is shifted to an area in the system where tests find with more defects than expected
C. Brainstorming sessions are held with a wide variety of stakeholders to identify possible failures in the system
D. Tests are prioritized to ensure that those associated with critical parts of the system are executed earlier

Correct Answer: A
Section:
Explanation:
Risk-based testing is an approach to testing that prioritizes and focuses on testing activities based on the level of risk associated with each feature or function of the system under test. Risk-based testing aims to optimize the use of time, resources, and techniques for testing by identifying and addressing the most critical and likely sources of failure or harm in the system under test. Some examples of typical risk-based testing activities are:
Brainstorming sessions are held with a wide variety of stakeholders to identify possible failures in the system: This activity is part of the risk identification process, which involves gathering information and opinions from different perspectives and sources to discover potential risks in the system under test.
Tests are prioritized to ensure that those associated with critical parts of the system are executed earlier: This activity is part of the risk analysis and evaluation process, which involves assessing the probability and impact of each risk and ranking them according to their severity and importance.
The focus of testing is shifted to an area in the system where tests find more defects than expected: This activity is part of the risk mitigation and monitoring process, which involves taking actions to reduce or eliminate the risks and tracking their status and progress.
The evaluation of a risk-management tool to decide which tool to use for future projects is not an example of a typical risk-based testing activity because it is not directly related to testing the system under test based on its risks. Rather, it is an example of a tool selection or evaluation activity, which involves comparing and choosing a tool that can support or enhance the testing process based on criteria such as functionality, usability, reliability, compatibility, cost, etc. You can find more information about risk-based testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 3, Section 3.4.

QUESTION 72
The following program part is given:
IF (condition A)
then DO B
END IF
How many test cases are necessary in order to achieve 100% statement coverage?

A. 1
B. 2
C. 4
D. a very high number

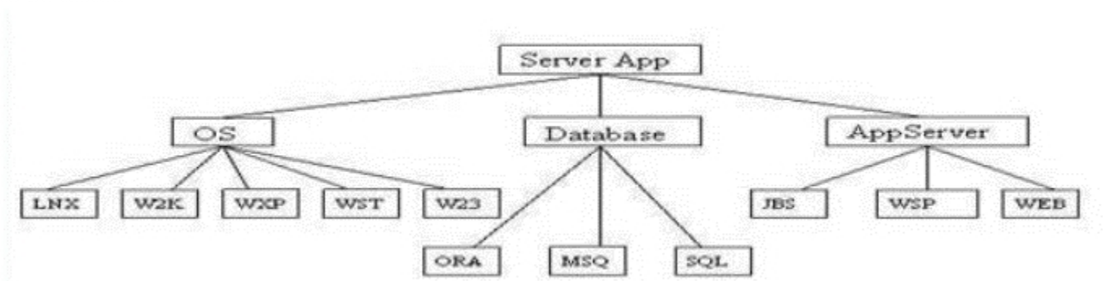
Correct Answer: B
Section:
Explanation:
To achieve 100% statement coverage, you need to execute every statement in the code at least once. In this case, you need two test cases: one where condition A is true and one where condition A is false. This way, you can cover both the DO B statement and the END IF statement. You can find more information about statement coverage inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Section 4.31.

QUESTION 73
Which of the following tool types is the most useful one for a test manager?

A. Modeling tool
B. Static analysis tool
C. Coverage measurement tool
D. Defect tracking tool

Correct Answer: D
Section:
Explanation:
A tool type is a category of tools that have similar features and functions and can be used for similar purposes. A test manager is a person who is responsible for planning, monitoring, controlling, and reporting the test activities and results. A test manager needs to use tools that can help them with these tasks and provide them with useful information and insights. Some examples of tool types that are useful for a test manager are:
Defect tracking tool: This tool type allows the test manager to record, track, manage, and report the defects found during testing. It also helps the test manager to communicate with developers, testers, and other stakeholders about the defect status and resolution.
Test management tool: This tool type allows the test manager to organize, manage, and control the test process and its outcomes. It also helps the test manager to define the test strategy, plan the test activities, allocate the test resources, schedule the test execution, monitor the test progress, and evaluate the test results.
Requirements management tool: This tool type allows the test manager to manage and trace the requirements of the system under test. It also helps the test manager to ensure that the test objectives are aligned with the user needs and expectations and that the test coverage is adequate and complete.
The other tool types mentioned in the question are not as useful for a test manager as they are for other roles or purposes. For example:
Modeling tool: This tool type allows the user to create graphical or textual representations of software systems or processes. It can be used for design, analysis, simulation, or documentation purposes.
Static analysis tool: This tool type allows the user to examine the code or design of a software system without executing it. It can be used for finding defects, measuring complexity, checking compliance, or improving quality.
Coverage measurement tool: This tool type allows the user to measure how much of the code or design of a software system has been exercised by a set of test cases. It can be used for evaluating the effectiveness or completeness of testing.
You can find more information about tool types inA Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 61.

QUESTION 74
The following diagram lists various types of operating systems, databases and application servers supported by the application under test. For complete coverage of all combinations, how many combinations of the above are to be tested?



A. 11
B. 5
C. 45

D. 3

Correct Answer: C
Section:
Explanation:
The diagram lists various types of operating systems (LNX, W2K, WSP), databases (ORA, MSQ, SQL), and application servers (JBS, WSP) supported by the application under test. To test all possible combinations of these types, we need to multiply the number of options in each category. In this case, we have:
3 options for operating systems
3 options for databases
2 options for application servers
Therefore, we have 3 x 3 x 2 = 18 possible combinations to test.
However, if we look closely at the diagram, we can see that some combinations are not valid or feasible because they are not connected by lines. For example, we cannot test LNX with WSP as an application server because there is no line between them. Similarly, we cannot test W2K with JBS as an application server because there is no line between them. Therefore, we need to exclude these invalid combinations from our calculation.
If we count only the valid combinations that are connected by lines in the diagram, we get:
5 combinations for LNX (LNX-ORA-JBS, LNX-ORA-WSP, LNX-MSQ-JBS, LNX-MSQ-WSP, LNX-SQL-JBS)
5 combinations for W2K (W2K-ORA-WSP, W2K-MSQ-WSP, W2K-SQL-WSP)
5 combinations for WSP (WSP-ORA-JBS, WSP-ORA-WSP, WSP-MSQ-JBS, WSP-MSQ-WSP)
Therefore, we have 5 + 5 + 5 = 15 valid combinations to test.
You can find more information about testing combinations inSoftware Testing Foundations: A Study Guide for the Certified Tester Exam, Chapter 4, Section 4.22.

QUESTION 75
Which ONE of the following statements does NOT describe how testing contributes to higher quality?

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

Correct Answer: B
Section:
Explanation:
The statement that does not describe how testing contributes to higher quality is ''The testing of software demonstrates the absence of defects''. This statement is false because testing cannot prove that a software system is defect-free or has a certain level of quality because there might be some untested inputs or outputs that could reveal defects or failures. Testing can only provide information about the quality and risk level of a software system based on the observed behavior and results under certain conditions and assumptions. The other statements describe how testing contributes to higher quality by:
Performing a review of the requirement specifications before implementing the system can enhance quality: This statement is true because reviewing the requirement specifications can help identify and prevent defects, ambiguities, inconsistencies, or incompleteness in the requirements before they are implemented in the system, which can save time and effort and improve customer satisfaction.
Properly designed tests that pass reduce the level of risk in a system: This statement is true because passing tests can increase confidence that the system meets its requirements and expectations and provides value to the users and customers. Passing tests can also reduce the probability or impact of failures or defects in the system that could cause harm or loss.
Software testing identifies defects, which can be used to improve development activities: This statement is true because identifying defects can help analyze and remove their causes and prevent them from recurring in future development activities. Identifying defects can also help improve development processes, methods, standards, tools, techniques, etc., by providing feedback and lessons learned.
You can find more information about how testing contributes to higher quality in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 1.

QUESTION 76
Testing should provide sufficient information to stakeholders to make informed decisions about the release of the software or system being tested. At which of the following fundamental test process activity the sufficiency of the testing and the resulting information are assessed?

A. Implementation and execution
B. Requirements specification
C. Evaluating exit criteria and reporting.

D. Analysis and design

Correct Answer: C
Section:
Explanation:
The fundamental test process activity where the sufficiency of testing and resulting information are assessed is Evaluating exit criteria and reporting. This activity involves checking whether the test objectives have been met and whether there are any unresolved issues or risks that could affect the release or deployment decision. This activity also involves preparing and communicating a test summary report that summarizes the test activities and results and provides recommendations and feedback for improvement. You can find more information about Evaluating exit criteria and reporting in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 3, Section 3.5.

QUESTION 77
A Software was re-deployed because the backend database was changed from one vendor to another The Test Manager decided to perform some functional tests on the redeployed system. This is an example of test of which test type?

A. Regression tests
B. Non-functional tests
C. Structural tests
D. Unit tests

Correct Answer: A
Section:
Explanation:
Regression testing is a type of testing that is performed after a software system has been changed or modified to verify that the changes have not introduced any new defects or adversely affected the existing functionality of the system. Regression testing can be performed when the software system undergoes different types of changes, such as:
Corrective changes: Changes that fix defects or errors in the software system
Adaptive changes: Changes that adapt the software system to new platforms or environments
Perfective changes: Changes that improve the performance or usability of the software system
Preventive changes: Changes that avoid potential problems or issues in the software system
In this case, the software system was re-deployed because the backend database was changed from one vendor to another, which is an example of an adaptive change. Therefore, the test manager decided to perform some functional tests on the re-deployed system to ensure that the change did not affect the functionality of the system. This is an example of regression testing.
The other types of testing mentioned in the question are not relevant for this scenario. For example:
Non-functional testing: This type of testing verifies the non-functional aspects of the software system, such as reliability, performance, security, usability, etc.
Structural testing: This type of testing verifies the internal structure or implementation of the software system, such as code, architecture, design, etc.
Unit testing: This type of testing verifies individual components or units of software in isolation from other components or systems.
You can find more information about regression testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 2, Section 2.4.

QUESTION 78
Which of the following statements is true?
I) Test planning activities include selecting the test object
II,) Test strategy implementation is NOT a part of test planning
II,I) Choosing appropriate test techniques is part of test design
IV) Test schedule and exit criteria modification are part of test planning

A. I, II,
B. II, II,I
C. I,IV
D. II,I, IV

Correct Answer: C

Section:
Explanation:
Statement I is true, as test planning activities include defining the test object, which is the component or system to be tested. Statement II, is false, as test strategy implementation is part of test planning and involves defining the test approach and test levels to be applied. Statement II,I is true, as test design activities include choosing appropriate test techniques to derive test cases from test conditions. Statement IV is true, as test planning activities include defining and updating the test schedule and exit criteria throughout the project lifecycle.
Reference: ISTQB Certified Tester Foundation Level Syllabus 2018, Section 2.1

QUESTION 79
Which test level is concerned with testing the smallest bodies of software?

A. Component test
B. Feature test
C. Functional test
D. Subsystem test

Correct Answer: A
Section:
Explanation:
Component testing is concerned with testing the smallest bodies of software, such as functions, classes, procedures, or modules. Component testing is also known as unit testing or module testing. Feature testing is a type of functional testing that focuses on verifying a specific feature or functionality of a system. Functional testing is a type of black-box testing that verifies that the system meets its functional requirements and specifications. Subsystem testing is a type of integration testing that verifies that a group of components work together as expected.
Reference: ISTQB Certified Tester Foundation Level Syllabus 2018, Section 2.2

QUESTION 80
Which of the following would be the LEAST likely to be used as the basis fcr a test exit criteria?

A. Cost of testing performed so far
B. Number of unfixed defects
C. Confidence of testers in tested code
D. Test schedules

Correct Answer: A
Section:
Explanation:
The cost of testing performed so far is not a test exit criteria, as it does not reflect the quality or completeness of the testing process.Test exit criteria are usually based on factors such as the number of unfixed defects, the confidence of testers in tested code, the test coverage achieved, and the test schedules

QUESTION 81
Which of the following metrics is from the test design phase?

A. Number of test cases run / not run
B. Number of defects found and fix
C. Percentage of test conditions covered by test cases
D. Subjective confidence of testes in the system under test

Correct Answer: C
Section:
Explanation:

The percentage of test conditions covered by test cases is a metric from the test design phase, as it measures how well the test cases cover the test basis13. The other metrics are from the test execution phase or the test closure phase.

QUESTION 82
Which of the following BEST matches the attributes with a level of testing?
I, Stubs and drivers are often used
II, The test environment should correspond to the production environment
II,I, Finding defects is not the main focus
IV Testing can be based on use cases
V, Testing is normally performed by testers
VI, Testing for functional and non-functional characteristics

A. Component-V Integration - II, System - IV Acceptance-VI
B. Component -1 Integration - V System - II, Acceptance - IV
C. Component - IV Integration -1 System-VI Acceptance-V
D. Component-VI Integration - IV System -1 Acceptance-II,I

Correct Answer: B
Section:
Explanation:
Explanation: The attributes that best match each level of testing are as follows14: Component testing: Stubs and drivers are often used (I), as they simulate the behavior of missing or incomplete components. Integration testing: Testing for functional and non-functional characteristics (VI), as integration testing verifies that components interact correctly and meet their specified requirements. System testing: The test environment should correspond to the production environment (II,), as system testing evaluates the system's compliance with its specified requirements under realistic conditions. Acceptance testing: Testing can be based on use cases (IV), as acceptance testing validates that the system meets the user's needs and expectations

QUESTION 83
A system sets new users' password to a temporary password
The temporary password is a random number based on the first sft: characters of the username.
If the username is shorter than 6 characters, the system displays an error message.
Which of the following is a possible representation of equivalence classes for the username string?

A. {Random number} {Error message}
B. {Error message displayed} {Error message not displayed}
C. {Username shorter than 6 characters} {Username equal to or longer than 6 characters}
D. {Username with a permanent password} {Username with a temporary password}

Correct Answer: C
Section:
Explanation:
The equivalence classes for the username string are based on the length of the string, not on the type of password or the error message. Therefore, option C is correct, as it divides the username string into two equivalence classes: shorter than 6 characters and equal to or longer than 6 characters. Option A is incorrect, as it does not consider the length of the username string. Option B is incorrect, as it does not specify the criteria for displaying or not displaying the error message. Option D is incorrect, as it confuses the username string with the password string.

QUESTION 84
What is a 'test harness'?

A. One of the topics that needs to be covered in the final Test Report.
B. A general name for test data that is used by the test cases.

C. A detailed description of a test case to enable its execution by non-expert tester.
D. A test environment comprised of stubs and drivers needed to execute a test.

Correct Answer: D
Section:
Explanation:
A test harness is a test environment comprised of stubs and drivers needed to execute a test, especially when testing components or interfaces. Therefore, option D is correct. Option A is incorrect, as a test harness is not a topic for the final Test Report. Option B is incorrect, as a test harness is not a general name for test data. Option C is incorrect, as a test harness is not a detailed description of a test case.

QUESTION 85
For a given set of test-cases, which of the following is a benefit of running these tests with a test automation tool?

A. Test coverage is increased.
B. The number of found bugs is reduced.
C. The total cost of the test project always decreases
D. The time spent on repetitive tasks is reduced

Correct Answer:
Section:
Explanation:
Explanation of Correct Answer: A benefit of running tests with a test automation tool is that the time

QUESTION 86
Which of the following is a key characteristic of informal reviews?

A. Metrics analysis
B. Kick-off meeting
C. Individual preparation
D. Low cost

Correct Answer:
Section:
Explanation:
Explanation of Correct Answer: A key characteristic of informal reviews is that they are low cost, as

QUESTION 87
Which of the following statements correctly describe Black-Box Testing''

A. Test on an individual software component in isolation from other components, to avoid external influence
B. Tests that investigate the input vs. trfe output behavior of a software system.
C. Test derived from the ability of the testers and their intuition and experience with similar applications and technologies (sometimes used to strengthen systematic techniques)
D. Test of the interfaces between components and of the interactions with different parts of a system

Correct Answer: B
Section:
Explanation:
Black-box testing is a test technique that investigates the input vs. output behavior of a software system, without considering its internal structure or implementation. Therefore, option B is correct. Option A is incorrect, as it describes white-box testing, which focuses on the internal structure or implementation of a software system. Option C is incorrect, as it describes exploratory testing, which relies on the testers' intuition and experience.

Option D is incorrect, as it describes integration testing, which tests the interfaces and interactions between components or systems.

QUESTION 88
Which of the following are correct tasks during 'Test analysis and design'?
I, Designing and prioritizing test cases
II, Identifying any required infrastructure and tools
II,I, Reviewing the test basis
IV Creating test data and preparing test harnesses
V, Writing automated test scripts

A. II, II,I, IV, V
B. l, II,I IV
C. I, II, lll
D. I, ll

Correct Answer: C
Section:
Explanation:
Test analysis and design is the phase of the test process where test cases and test data are designed and prioritized, based on the test basis and test objectives. Therefore, option C is correct, as it includes tasks I, II, and II,I, Option A is incorrect, as it includes tasks IV and V, which are part of test implementation and execution phase. Option B is incorrect, as it includes task IV, which is part of test implementation and execution phase. Option D is incorrect, as it does not include task II,I, which is part of test analysis and design phase.

QUESTION 89
While reporting a defect, which of the following indicates the level of business importance assigned to the defect?

A. Urgency
B. Priority
C. Difficulty
D. Severity

Correct Answer:
Section:
Explanation:
Explanation of Correct Answer: Priority indicates the level of business importance assigned to the

QUESTION 90
Which of the following should be considered when purchasing a test execution tool?

A. The ability of the tool to trace tests to requirements and report coverage level
B. The amount of effort required to achieve positive Return on Investment (ROI)
C. The ability of the tool to track the output and productivity of each individual tester
D. The ability of the tool to run the unit-level tests for the developers

Correct Answer: D
Section:
Explanation:
A test execution tool is a tool that executes tests on a software system and logs the outcomes of each test step. The ability of the tool to run the unit-level tests for the developers is one of the factors that should be considered when purchasing a test execution tool, as it can improve the efficiency and effectiveness of testing at this level. The other options are not relevant to test execution tools, but to other types of tools such as test

management tools, test measurement tools, and requirements management tools.
Reference:Section 6.1.1,Section 6.3.2

QUESTION 91
Which of the following factors is LEAST likely to be used as a basis for estimating testing effort?

A. Requirements for documentation
B. Quality of the test basis Work
C. Breakdown Structure
D. Skills of the test team

Correct Answer: C
Section:
Explanation:
Work Breakdown Structure (WBS) is a project management technique that decomposes a project into smaller and manageable tasks. It is not a factor that is used as a basis for estimating testing effort, but rather an outcome of the estimation process. The other options are factors that can influence the testing effort, such as the quality of the test basis, the requirements for documentation, and the skills of the test team.

QUESTION 92
Can 'cost' be regarded as Exit criteria'?

A. No The financial value of product quality cannot be estimated so it is incorrect to use cost as an exit criteria
B. Yes. Spending too much money on testing will result in an unprofitable product, and having cost as an exit criteria helps avoid this
C. Yes Going by cost as an exit criteria constrains the testing project which will help achieve the desired quality level defined for the project
D. No The cost of testing cannot be measured effectively, so it is incorrect to use cost as an exit criteria

Correct Answer: B
Section:
Explanation:
Cost can be regarded as an exit criterion, as it is one of the factors that can influence the decision to stop testing. Spending too much money on testing will result in an unprofitable product, and having cost as an exit criterion helps avoid this. The other options are incorrect, as the financial value of product quality can be estimated, the cost of testing can be measured effectively, and having cost as an exit criterion does not necessarily constrain the testing project or help achieve the desired quality level.

QUESTION 93
Which of the following is a correct reason to apply test automation?

A. When a new test automation tool is launched
B. When there are a lot of repetitive testing tasks
C. When it is easy to automate
D. When it is cheap to buy test automation tools

Correct Answer: B
Section:
Explanation:
A correct reason to apply test automation is when there are a lot of repetitive testing tasks, as test automation can reduce the effort and time required to execute them, and improve the consistency and accuracy of the results. The other options are not valid reasons to apply test automation, as they do not consider the benefits and risks of test automation, the suitability of the test object for automation, or the availability of resources and skills for automation.

QUESTION 94

Which of the following processes is related to ensuring the integrity of the testware?

A. Configuration management
B. Test automation
C. Build release
D. Project management

Correct Answer: A
Section:
Explanation:
Configuration management is related to ensuring the integrity of the testware, as it is a process that controls and documents changes to test items, test cases, test data, test scripts, test results, and test documentation throughout the testing lifecycle. The other options are not related to ensuring the integrity of the testware, but to other aspects of testing such as test execution, test environment setup, and test planning.

QUESTION 95
Considering the following pseudo-code, calculate the MINIMUM number of test cases required to achieve 100% statement coverage and 100% decision coverage.

```
IF user age > 14 THEN
 IF user name = 'blank' THEN
  Error message
 ELSE
  IF sex=null or age>120 THEN
   Error message
  ELSE
   Correct entry
  ENDIF
ELSE
 Error message
ENDIF
```

A. Statement coverage: 4, Decision coverage: 5
B. Statement coverage. 3. Decision coverage 4
C. Statement coverage 4. Decision coverage 4
D. Statement coverage: 4. Decision coverage: 3

Correct Answer: C
Section:
Explanation:
To achieve 100% statement coverage, we need to execute every statement in the code at least once. To achieve 100% decision coverage, we need to execute every possible outcome of every decision in the code at least once. In this case, we can use the following test cases to cover both statement and decision coverage:
A = 5, B = 4, C = 3, D = 2, E = 1
A = 1, B = 2, C = 3, D = 4, E = 5
A = 3, B = 4, C = 5, D = 2, E = 1
A = 3, B = 2, C = 1, D = 4, E = 5
These four test cases will cover all the statements and all the possible outcomes of the decisions in the code. Therefore, the minimum number of test cases required to achieve both statement and decision coverage is four.

QUESTION 96
Consider the following Pseudo code:

```
If (A>B)
   THEN: If (A>C)
      THEN: If (A>D)
         THEN: If (A>E)
            THEN: Print "A is huge"
         End IF
      End IF
   End IF
End IF
```

How many minimum lea cases are required to cover 100% Statement coverage and Decision coverage?

A. 1 for Statement. 5 for Decision
B. 1 for Statement. 2 for Decision
C. 5 for Statement. 1 for Decision
D. 2 for Statement, 5 for Decision

Correct Answer: B
Section:
Explanation:
To achieve 100% statement coverage, we need to execute every statement in the code at least once. To achieve 100% decision coverage, we need to execute every possible outcome of every decision in the code at least once. In this case, we can use the following test cases to cover both statement and decision coverage:
A = true, B = true
A = false, B = false
These two test cases will cover all the statements and all the possible outcomes of the decisions in the code. Therefore, the minimum number of test cases required to achieve both statement and decision coverage is two.

QUESTION 97
What is a typical benefit of use case testing?

A. Finding failures in the possible states transitions
B. Clearer identification of system equivalence partitioning
C. Finding of failures in the business process flows, which highlight system use in the real world
D. Identification of bugs in the system components.

Correct Answer: C
Section:
Explanation:
Use case testing is a technique that uses scenarios based on use cases to test the functionality and usability of a system from the user's perspective. A use case is a description of how a system interacts with one or more actors (users or other systems) to achieve a specific goal. A typical benefit of use case testing is finding failures in the business process flows, which highlight system use in the real world. This is because use case testing focuses on how the system is used by different actors in different situations, rather than on individual components or features of the system.

QUESTION 98
Which of the following statements is the best explanation why software failures can be caused by environmental conditions?

A. Factors like magnetism, radiation and even pollution can affect electronic devices and the performance of their embedded real time software
B. Environmental conditions only affect the hardware - not the software
C. If the hardware on which the software application is running under ambient temperature and humidity, no failures can be linked to environmental conditions
D. Extreme heat and vibrations exerted on storage media can cause errors in algorithms and program flows

Correct Answer: B

Section:
Explanation:
The statement that environmental conditions only affect the hardware - not the software is not a good explanation why software failures can be caused by environmental conditions. Environmental conditions can affect both the hardware and the software, as they can influence the performance, reliability, and functionality of the system. For example, factors like magnetism, radiation, pollution, heat, humidity, vibration, etc. can affect electronic devices and the embedded software that runs on them. Software failures caused by environmental conditions can have serious consequences, especially for safety-critical systems.
Reference:Certified Tester Foundation Level Syllabus, Section 1.2.2

QUESTION 99
Which of the following test types are non-functional tests?
I) Acceptance test
II,) Regression test
II,I) Stress test
IV) Component test
V) Reliability test

A. I, II,I and V
B. I, II, and IV
C. II, Ill and V
D. II,I and V

Correct Answer: A
Section:
Explanation:
Non-functional testing is a type of testing that focuses on the quality attributes of the system or its components, such as performance, reliability, usability, security, etc. Non-functional testing can be done at any test level, depending on the test objectives and scope. In this case, the test types that are non-functional tests are:
Acceptance test: A test level that focuses on verifying whether the system meets the user's needs and expectations, and whether it is acceptable for delivery or deployment.
Stress test: A type of performance testing that evaluates the behavior of the system under extreme or abnormal conditions, such as high load, limited resources, or concurrent access.
Reliability test: A type of testing that evaluates the ability of the system or its components to perform their required functions under stated conditions for a specified period of time.
Therefore, statements I, II,I, and V are correct.

QUESTION 100
Which of the following is NOT a test control activity?

A. Re-prioritize tests because of time pressure
B. Change the test schedule due to viability of a test environment
C. Writing test suspension and resumption criteria in the test plan
D. Set an entry criterion requiring fixes to be retested by a developer before accepting them into a build

Correct Answer: C
Section:
Explanation:
Test control is a test management activity that involves making decisions based on information from test monitoring and test progress reporting. Test control activities include:
Re-prioritizing tests because of time pressure
Changing the test schedule due to availability of a test environment
Setting entry and exit criteria for test levels or test activities
Initiating corrective actions to resolve deviations from the test plan
Writing test suspension and resumption criteria in the test plan is not a test control activity, but rather a test planning activity. Test suspension and resumption criteria are used to define when to stop and resume testing in case of unforeseen events or issues that affect the test execution.

QUESTION 101
Which of the following statements about use cases and use case testing is NOT TRUE?

A. Use cases can be used as test basis for acceptance testing
B. Use case testing can find defects in the process flow.
C. Use cases can be described at the abstract level or at the system level
D. Use cases are normally derived from decision tables

Correct Answer: D
Section:
Explanation:
The statement that use cases are normally derived from decision tables is not true. Decision tables are a technique to represent complex logical conditions in a tabular form, where each column represents a possible combination of conditions and actions. Use cases are a technique to describe how a system interacts with one or more actors (users or other systems) to achieve a specific goal. Use cases are not derived from decision tables, but from the requirements and the user's perspective of the system.

QUESTION 102
What is a defect density?

A. The percentage of defects identified over the total number of test cases
B. The deviation rate of the system from its expected behavior
C. The ratio of the defects identified in these system over the expected number of total defects
D. The number of defects identified in the system under test divided by the size of the system

Correct Answer: D
Section:
Explanation:
Defect density is a metric that measures the number of defects identified in the system under test divided by the size of the system. The size of the system can be measured in different ways, such as lines of code, function points, or modules. Defect density can be used to compare the quality of different systems or different versions of the same system.

QUESTION 103
What is the order in which the specifications for test cases, test conditions and test procedures are developed as a part of the test development process?

A. Test Procedure --> Test Condition -> Test Case
B. Test Condition --> Test Case --> Test Procedure
C. Test Procedure -> Test Case -> Test Condition
D. Test Condition --> Test Procedure --> Test Case

Correct Answer: B
Section:
Explanation:
The order in which the specifications for test cases, test conditions and test procedures are developed as a part of the test development process is as follows:
Test Condition: A test condition is an item or event of a component or system that could be verified by one or more test cases, such as a function, transaction, feature, quality attribute, or structural element.
Test Case: A test case is a set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement.
Test Procedure: A test procedure is a document providing detailed instructions for the execution of one or more test cases.
Therefore, the correct order is test condition --> test case --> test procedure.

QUESTION 104

When should component integration tests be carried out?

A. Integration tests should always be done after system tests
B. Integration tests should be done at the customer's site, after acceptance tests
C. Integration tests can be done before or after system tests
D. Integration tests should always be done before system tests

Correct Answer: D
Section:
Explanation:
Component integration testing is the process of testing the interactions between different components of a system. It should always be done before system testing, which is the process of testing the system as a whole. Integration testing can reveal defects in the interfaces and interactions between components, which can affect the functionality and performance of the system. System testing can reveal defects in the system requirements, design, and behavior, which can affect the quality and usability of the system. Therefore, it is logical to perform integration testing before system testing, to ensure that the components work together correctly before testing the system as a whole.

QUESTION 105
Which of the following activities do NOT belong to test implementation and execution?

A. Checking if the preconditions of test execution have been met
B. Logging of test results
C. Test data generation
D. Prioritizing test conditions

Correct Answer: D
Section:
Explanation:
Test implementation and execution is the phase of the test process where test cases are executed and test results are logged. It includes activities such as checking if the preconditions of test execution have been met, executing test cases according to a test procedure, comparing actual results with expected results, logging test results and reporting defects, and repeating test activities as needed. Prioritizing test conditions is not an activity that belongs to test implementation and execution, but rather to test analysis and design, which is the phase where test conditions are identified and test cases are designed.
Reference:Certified Tester Foundation Level Syllabus, Section 4.2.3 and 4.2.4

QUESTION 106
Choose below the best characterization of the difference between black-box testing and white-box testing

A. For black-box testing, the testers do not need to know the test object in advance
B. For white-box testing, the testers have to consult specifications and developers before running any tests
C. Black box testing uses static analysis. White box testing uses dynamic analysis
D. Black-box testing is based on the test basis documentation. White-box testing is based on an analysis of the code itself.
E. Black-box testing is based on an analysis of the code itself White-box testing is based on the test basis.

Correct Answer: D
Section:
Explanation:
Black-box testing and white-box testing are two approaches to test design that differ in the way they derive test cases from the test basis. Black-box testing is based on an analysis of the test basis documentation, such as requirements, specifications, user stories, or use cases, without looking at the internal structure or code of the system under test. White-box testing is based on an analysis of the code itself, such as statements, branches, paths, or data flows, to design test cases that cover specific aspects of the code. Black-box testing focuses on the external behavior and functionality of the system, while white-box testing focuses on the internal logic and structure of the system.

QUESTION 107
For a mandatory input field 'ZIP code' the following rules are given:
1 - The valid ZIP code format is 5 numeric digits
2 - The code has to exist in the pest office's official ZIP code list
Using equivalence classes partitioning, how many test cases are required to test this field?

A. 6
B. 4
C. 8
D. 3

Correct Answer: B
Section:
Explanation:
Equivalence partitioning is a technique to divide a set of possible inputs or outputs into classes that are expected to behave similarly or produce similar results. For each equivalence class, only one test case is required to represent the whole class. In this case, we can identify the following equivalence classes for the input field 'ZIP code':
Valid ZIP code format and valid ZIP code value (e.g., 12345)
Valid ZIP code format and invalid ZIP code value (e.g., 99999)
Invalid ZIP code format and valid ZIP code value (e.g., 1234)
Invalid ZIP code format and invalid ZIP code value (e.g., ABCDE)
Therefore, four test cases are required to test this field using equivalence partitioning.

QUESTION 108
Which of the following statements about independent testing is WRONG?

A. Independent testing is necessary because developers don't know any testing
B. A certain degree of independence makes the tester more effective at finding defects
C. Independent testing is best suited for the system test level
D. Independent test teams may find other types of defects than developers who are familiar with the system's structure.

Correct Answer: A
Section:
Explanation:
The statement that independent testing is necessary because developers don't know any testing is wrong. Developers do know some testing, as they perform unit testing and component testing on their own code. However, independent testing is beneficial because it can provide a different perspective and a higher degree of objectivity than testing done by developers. Independent testers may have different skills, knowledge, experience, and expectations than developers, which can help them find different types of defects and improve the quality of the system.

QUESTION 109
Integration testing has following characteristics.
I, It can be done in incremental manner
II, It is always done after system testing
II,I, It includes functional tests
IV It includes non-functional tests

A. I, I and IV are correct
B. I Ill and IV are correct
C. TI is correct
D. II, and II,I are correct

Correct Answer: B
Section:
Explanation:
Integration testing is the process of testing the interactions between different components or subsystems of a system. It has the following characteristics:
It can be done in an incremental manner, meaning that components or subsystems are integrated and tested one by one until the whole system is integrated and tested.
It is usually done before system testing, which is the process of testing the system as a whole against its requirements and specifications.
It includes functional tests, which are tests that verify the functionality and behavior of the system or its components.
It includes non-functional tests, which are tests that verify the quality attributes of the system or its components, such as performance, reliability, security, etc.
Therefore, statements I, II,I, and IV are correct.

QUESTION 110
Which of the following is NOT a common objective of testing?

A. Providing information on the status of the system
B. Preventing defects
C. Finding defects in the software
D. Debugging the software to find the reason for defects

Correct Answer: D
Section:
Explanation:
Debugging the software to find the reason for defects is not a common objective of testing. Debugging is the process of finding, analyzing, and removing the causes of failures in software. Debugging is usually done by developers after testers have reported defects found during testing. Testing is the process of evaluating software by applying test cases to find defects and provide information on its quality. Testing has several common objectives, such as:
Providing information on the status of the system, such as its readiness for release, its compliance with requirements, its risks and issues, etc.
Preventing defects, by applying good practices throughout the software development lifecycle, such as reviews, inspections, static analysis, etc.
Finding defects in the software, by applying test cases that cover different aspects of the software functionality, quality, performance, security, etc.
Therefore, statement D is not correct.

QUESTION 111
Which of the following DOES NOT describe 'component testing'?

A. Component testing tests interfaces between modules and interactions of different parts of a system.
B. Component testing occurs with access to the code being tested and with the support of a development environment, such as a unit test framework or debugging tool
C. Component testing verifies the functioning of. software modules, programs, objects, classes, etc. that are separately testable.
D. In component testing stubs, drivers and simulators may be usefully utilized to facilitate tester activity

Correct Answer: A
Section:
Explanation:
The statement that component testing tests interfaces between modules and interactions of different parts of a system does not describe component testing. Component testing is the process of testing individual software components that are separately testable, such as modules, programs, objects, classes, etc. Component testing verifies the functionality and quality of the components in isolation from other components. Component testing does not test the interfaces or interactions between components, as this is done in integration testing.

QUESTION 112
Which type of software development product can undergo static testing?

A. Any software development product can undergo static testing, including requirements specifications, design specifications and code

B. Static testing is done only on the requirements. You need to execute the software in order to find defects in the code
C. Static testing is done only on the code as part of the 'code review' sessions Other documents are reviewed, but not by static testing Static tests
D. should be performed on the installation and user guide documents as these documents are used by the end user

Correct Answer: A
Section:
Explanation:
Any software development product can undergo static testing, including requirements specifications, design specifications, code, test cases, test plans, user manuals, etc. Static testing is a type of testing that does not involve executing the software or system under test, but rather analyzing it using various techniques, such as reviews, inspections, walkthroughs, checklists, static analysis tools, etc. Static testing can help find defects and improve the quality of any software development product at any stage of the software development lifecycle.

QUESTION 113
You need to test a vending machine for light drinks The machine has a button for each of the drinks it contains
Pressing a button before inserting coins, will display the cost of the drink
Pressing the same button after inserting enough coins will dispense the drink and provide change if needed)
If the machine is out of the specific drink, and the button for this drink is pressed, the machine displays 'Sold Out' (regardless if coins were inserted or not).
Which test technique is most suitable for this situation?

A. State transition testing
B. Equivalence class testing
C. Boundary value testing
D. Use-case testing

Correct Answer: A
Section:
Explanation:
State transition testing is the most suitable test technique for this situation. State transition testing is a technique that uses scenarios based on state diagrams to test the behavior of a system or component that can change its state in response to events or inputs. A state diagram is a graphical representation of the states that a system or component can be in, the events or inputs that can trigger a change of state (transitions), and the actions or outputs that can occur as a result of a change of state. In this case, the vending machine for light drinks can be modeled as a state diagram with states such as ''idle'', ''coins inserted'', ''drink selected'', ''drink dispensed'', ''change returned'', ''sold out'', etc., and transitions triggered by events such as ''button pressed'', ''coins inserted'', ''drink dispensed'', ''change returned'', etc. State transition testing can help design test cases that cover all the possible states and transitions of the vending machine and verify its functionality and usability.

QUESTION 114
A tester thinks of a likely cause for a specific bug Should the tester make a comment about this in the bug report?

A. No. A bug r^&ort must only include factual information, and not unsupported hypothesis
B. No. Such addition may bias the developers' attitude when they attempt to fix the bug
C. Yes It will reduce the risk that the bug fix will cause a regression
D. Yes Observations that may help correct a bug should be included in the bug report

Correct Answer: D
Section:
Explanation:
A bug report is a document that records the details of a defect or failure found during testing, such as the steps to reproduce, the expected and actual results, the severity and priority, etc. A bug report should also include any observations or suggestions that may help correct the bug, such as the possible cause, the affected components, the related requirements, etc. This information can help the developers to locate and fix the bug more efficiently and effectively. Therefore, if a tester thinks of a likely cause for a specific bug, they should make a comment about it in the bug report.

QUESTION 115

Which of the following sentences about testing and debugging is correct?

A. Re-testing checks that debugging has found and analyzed the failure
B. Dynamic testing finds defects, while debugging removes failures
C. Dynamic testing reveals failures, while debugging removes defects
D. Like most development activities, debugging is usually done before testing starts

Correct Answer: C
Section:
Explanation:
Testing and debugging are two different activities that are related to finding and removing defects and failures in software. Testing is the process of evaluating software by applying test cases to find failures and provide information on its quality. Debugging is the process of finding, analyzing, and removing the causes of failures in software. Testing reveals failures, which are deviations of the actual behavior of the software from its expected behavior. Debugging removes defects, which are flaws in the software that cause failures.

QUESTION 116
What is integration testing?

A. Looking for faults in larger components or subsystems
B. Another term for testing system integrity
C. Specifying which components to integrate in which order
D. Testing that the interfaces work correctly

Correct Answer: D
Section:
Explanation:
Integration testing is the process of testing the interactions between different components or subsystems of a system. Integration testing verifies that the interfaces work correctly, meaning that they pass data and control correctly between components or subsystems, and that they handle errors and exceptions properly. Integration testing can also verify the functionality and quality of the integrated system or subsystem.

QUESTION 117
Which of the following BEST describes a Test Case?

A. A statement about ''what to test'' in terms of measurable coverage criteria from analysis of the test basis.
B. A set of preconditions, inputs, actions, expected results and postconditions developed based on test conditions.
C. A description of the test objectives to be achieved and the means and the schedule for achieving them.
D. A source to determine expected results to compare with the actual result of the system under test.

Correct Answer: B
Section:
Explanation:
A test case is a set of preconditions, inputs, actions, expected results and postconditions developed based on test conditions. A test condition is a statement about what to test in terms of measurable coverage criteria from analysis of the test basis. A test plan is a description of the test objectives to be achieved and the means and the schedule for achieving them. A test oracle is a source to determine expected results to compare with the actual result of the system under test.

QUESTION 118
Testing and Debugging are key activities in the software development lifecycle.
Which of the following are DEBUGGING activities?
a) Designing tests to find failures.
b) Locating the cause of failures.

c) Analysing and fixing the defects.
d) Executing tests to show failures.

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

Correct Answer: C
Section:
Explanation:
Testing and debugging are two different activities that are related to finding and removing defects and failures in software. Testing is the process of evaluating software by applying test cases to find failures and provide information on its quality. Debugging is the process of finding, analyzing, and removing the causes of failures in software. Designing tests to find failures and executing tests to show failures are testing activities. Locating the cause of failures and analyzing and fixing the defects are debugging activities.

QUESTION 119
During which stage of the fundamental test process is the testability of requirements evaluated?

A. Test Execution.
B. Test Planning.
C. Test Design.
D. Test Analysis.

Correct Answer: D
Section:
Explanation:
The fundamental test process consists of five main activities: test planning and control, test analysis, test design, test implementation and execution, and test evaluation and reporting. During the test analysis activity, the testability of requirements is evaluated, meaning that the requirements are checked for clarity, completeness, consistency, verifiability, etc. Test analysis also involves identifying test conditions based on the test basis, such as requirements, specifications, user stories, or use cases.

QUESTION 120
Which ONE of the following is the BEST way to take advantage of the different mindsets of testers and developers?

A. Insist on independent testing at all stages in the lifecycle.
B. Have all developers undergo ISTQB training.
C. Keep developers and testers in separate teams.
D. Bring the two mindsets together.

Correct Answer: D
Section:
Explanation:
The best way to take advantage of the different mindsets of testers and developers is to bring the two mindsets together. Testers and developers have different perspectives and skills that can complement each other and improve the quality of the software. Testers tend to focus on finding defects and verifying the software against the requirements and specifications, while developers tend to focus on creating and implementing the software according to the design and architecture. By collaborating and communicating with each other, testers and developers can share their knowledge, feedback, and ideas, and resolve issues more efficiently and effectively.

QUESTION 121
In which development life cycle model is regression testing an increasingly important activity as the project progresses?

A. V-model.
B. Waterfall.
C. Scrum.
D. Progressive.

Correct Answer: C
Section:
Explanation:
In a Scrum development life cycle model, regression testing is an increasingly important activity as the project progresses. Scrum is an agile framework that follows an iterative and incremental approach to software development, where the software is delivered in small units called sprints. Regression testing is a type of testing that verifies that previously tested software still performs as expected after changes or modifications. Regression testing is essential in Scrum, as new features or functionalities are added or modified in each sprint, which can introduce new defects or affect existing ones. Therefore, regression testing should be performed at the end of each sprint, as well as before releasing the software, to ensure that the software meets the quality standards and user expectations.

QUESTION 122
Which of the following apply to System Testing?
a) May satisfy legal requirements.
b) Can use system specifications as a test basis.
c) Often the responsibility of business users.
d) Main goal is to establish confidence.
e) Should focus on the communication between systems.

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

Correct Answer: B
Section:
Explanation:
System testing is a test level that focuses on testing the system as a whole against its requirements and specifications. System testing applies to both functional and non-functional aspects of the system, such as functionality, performance, reliability, usability, security, etc. System testing has several objectives, such as:
Satisfying legal requirements, such as compliance with standards or regulations.
Using system specifications as a test basis, such as system requirements, system design, user stories, or use cases.
Establishing confidence in the system, such as its readiness for release or deployment, its suitability for use, its quality and reliability, etc.
Therefore, statements a and b apply to system testing.

QUESTION 123
After a record of poor-quality software releases (incorrect menu selection options, new features that do not work, users allowed to change security levels without administrator rights), you have been asked to review the test capability in your company.
You have limited time to do the review before the next project, which type of testing would be MOST appropriate to review first?

A. Functional testing.
B. Non-functional testing.
C. Performance testing.
D. Structural testing.

Correct Answer: A
Section:

Explanation:
Functional testing is the most appropriate type of testing to review first, after a record of poor-quality software releases. Functional testing is a type of testing that verifies the functionality and behavior of the software against its requirements and specifications. Functional testing can help detect defects such as incorrect menu selection options, new features that do not work, users allowed to change security levels without administrator rights, etc. Functional testing can also help improve the user satisfaction and confidence in the software. Therefore, functional testing should be reviewed first to ensure that it is done effectively and efficiently, and that it covers all the relevant aspects of the software functionality.

QUESTION 124
Which of the following are triggers for Maintenance testing?
a) System migration from one platform to another.
b) Retirement of a system.
c) Preparation for an audit of a system.
d) Modifications to a system.
e) Development of a whole new system.

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

Correct Answer: A
Section:
Explanation:
Maintenance testing is a type of testing that is done on an existing system after modifications or migration, or to prevent deterioration or obsolescence. Maintenance testing can be triggered by various events or situations, such as:
System migration from one platform to another, which can affect the functionality, performance, or compatibility of the system.
Preparation for an audit of a system, which can require verifying the compliance of the system with standards or regulations.
Modifications to a system, which can introduce new defects or affect existing ones.
Therefore, statements a, c, and d are correct.

QUESTION 125
Which statement about Static Testing is TRUE?

A. Static testing can be applied to any work product that participants know how to read and understand.
B. Static testing must only be applied to final work products that have been signed off.
C. Static testing must be conducted by users of the product being tested.
D. Static testing executes the code to verify the functionality is as expected.

Correct Answer: A
Section:
Explanation:
Static testing is a type of testing that does not involve executing the software or system under test, but rather analyzing it using various techniques, such as reviews, inspections, walkthroughs, checklists, static analysis tools, etc. Static testing can be applied to any work product that participants know how to read and understand, such as requirements specifications, design specifications, code, test cases, test plans, user manuals, etc. Static testing can help find defects and improve the quality of any work product at any stage of the software development lifecycle.

QUESTION 126
When comparing Static and Dynamic test techniques, which of the following statements is TRUE?

A. Static Testing finds failures, whilst Dynamic Testing only finds the cause of failures.

B. Static Testing techniques, such as reviews, can be undertaken before Dynamic Testing, making defects cheaper to remove.
C. Static Testing is based on the execution of code, whist Dynamic Testing relies on examination and analysis.
D. Only Static Testing has the objective of identifying defects.

Correct Answer: B
Section:
Explanation:
Static testing and dynamic testing are two types of testing that differ in the way they evaluate the software or system under test. Static testing is a type of testing that does not involve executing the software or system under test, but rather analyzing it using various techniques, such as reviews, inspections, walkthroughs, checklists, static analysis tools, etc. Dynamic testing is a type of testing that involves executing the software or system under test using test cases and comparing the actual results with the expected results. Static testing techniques, such as reviews, can be undertaken before dynamic testing, making defects cheaper to remove. This is because static testing can help find defects and improve the quality of any work product at any stage of the software development lifecycle, before they become more costly and difficult to fix during dynamic testing or after deployment.

QUESTION 127
Which defect below is MOST likely to be found by a review during static testing?

A. Incorrect interface specifications.
B. Old versions of software in use.
C. Performance bottlenecks.
D. Broken links to web addresses.

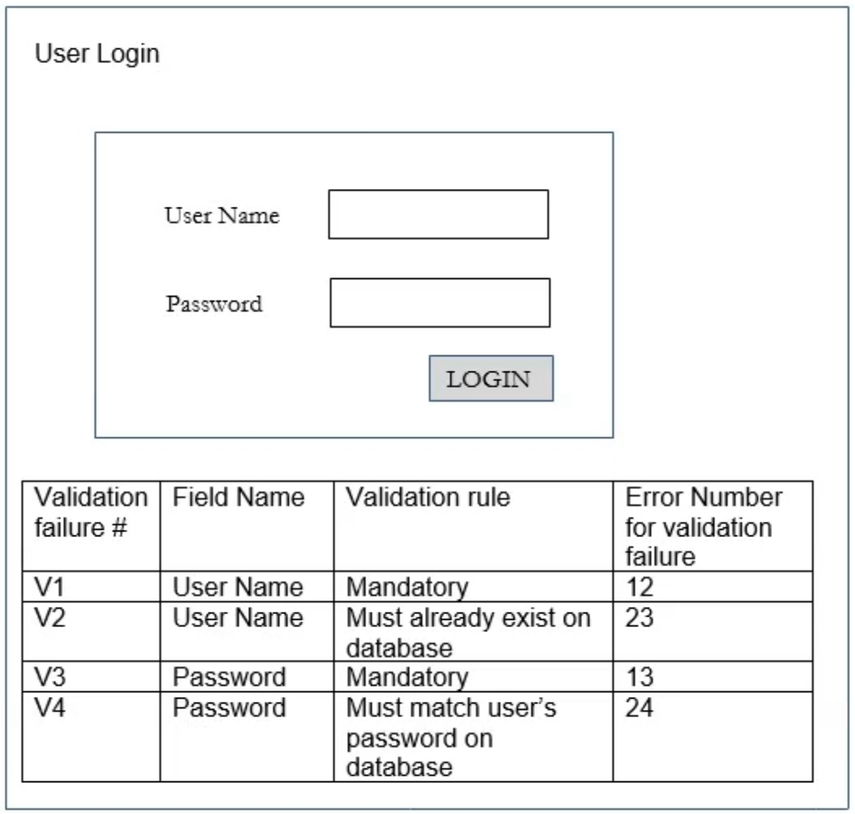
Correct Answer: A
Section:
Explanation:
Incorrect interface specifications are most likely to be found by a review during static testing. A review is a type of static testing technique that involves a manual examination of a work product by one or more individuals who follow a defined process. A review can be applied to any work product, such as requirements specifications, design specifications, code, test cases, test plans, user manuals, etc. A review can help find defects and improve the quality of any work product at any stage of the software development lifecycle. Incorrect interface specifications are defects that affect the definition and communication of the interfaces between components or systems. These defects can be detected by reviewing the interface specifications document and comparing it with the requirements and design specifications.

QUESTION 128
Which of the following is a typical characteristic of the WALKTHROUGH review type?

A. The meeting is led by the author.
B. Metrics are gathered throughout.
C. Attendees must prepare before the meeting.
D. Entry and exit criteria are defined.

Correct Answer: A
Section:
Explanation:
A typical characteristic of the walkthrough review type is that the meeting is led by the author. A walkthrough is a type of informal review that involves a presentation of a work product by the author to peers and other stakeholders for feedback and comments. The author leads the meeting and explains the work product to the participants, who can ask questions and make suggestions for improvement. The author may also record any issues or action items during the meeting. A walkthrough does not require formal preparation or follow-up by the participants, nor does it have defined entry and exit criteria.

QUESTION 129
You are performing a review of your colleague's test cases based on the following test basis document:



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

The Test Cases are as follows:
TC1. Success -- valid 'User Name' and 'Password'; Customer Menu displayed
TC2. Failure -- 'User Name' field has blank entry; Error Number 12 displayed
TC3. Failure -- 'User Id' entered does not exist on database (i.e. unregistered user); Error Number 23 displayed
TC4. Failure -- 'Password' entered does not match user's password on database; Error Number 24 displayed
You are guided by the following checklist in your review:
C1. There must be one test case to cover success
C2. There must be one test case for each error path (e.g. validation failure)
C3. Each test case must use terminology consistent with the test basis document (field names, error numbering, etc.)
Record a separate defect for each missing test case (checklist items C1 and C2) and for each test case that does not meet checklist item C3.
How many defects should you record?

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

Correct Answer: C
Section:
Explanation:
Defect 1: Missing test case for validation failure of 'Password' field being blank. According to checklist item C2, there must be one test case for each error path. The test basis document specifies that the 'Password' field must be mandatory, and that if it is blank, Error Number 13 should be displayed. However, none of the test cases cover this scenario. Therefore, a test case for this validation failure is missing and should be added.
Defect 2: Incorrect field name in TC3. According to checklist item C3, each test case must use terminology consistent with the test basis document. The test basis document uses the term 'User Name' to refer to the field that

identifies the user. However, TC3 uses the term 'User Id', which is inconsistent and confusing. Therefore, TC3 should be corrected to use the term 'User Name' instead of 'User Id'.
Defect 3: Incorrect error number in TC3. According to checklist item C3, each test case must use terminology consistent with the test basis document. The test basis document specifies that if the 'User Name' entered does not exist on the database, Error Number 12 should be displayed. However, TC3 states that Error Number 23 should be displayed, which is incorrect and does not match the test basis document. Therefore, TC3 should be corrected to use Error Number 12 instead of Error Number 23.

QUESTION 130
A holiday club restricts those booking the holiday, to people between the ages of 18 and 30 inclusive.
Using three-point boundary values, what ages would be required to test the lower and upper boundary?

A. 17, 18, 19, 29, 30, 31.
B. 17, 18, 19, 30, 31, 32.
C. 18, 19, 20, 28, 29, 30.
D. 16, 17, 18, 30, 31, 32.

Correct Answer: A
Section:
Explanation:
According to the syllabus, boundary value analysis is a technique that tests the values on and near the boundaries of an equivalence partition. The three-point boundary values are the minimum, just below the minimum, and just above the minimum for the lower boundary, and the maximum, just below the maximum, and just above the maximum for the upper boundary. The answer A is correct because it contains the three-point boundary values for both the lower and upper boundaries of the age range. The other answers are incorrect because they either miss some of the boundary values or include values that are too far from the boundaries.

QUESTION 131
Given the following decision tables, what is the expected result for the test case listed below?

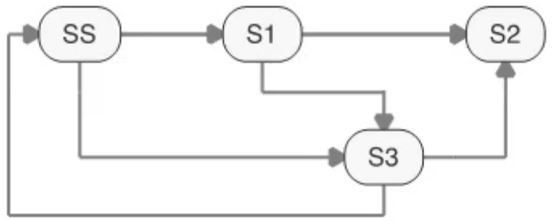
	Rule 1	Rule 2	Rule 3	Rule 4
Condition				
< 10 kg	True	True	False	False
< £10	True	False	True	False
Action				
Must pay in cash only	True	False	True	False
Free delivery	False	False	True	True

Test Case: Purchase a Toaster weighing 9kg for 10.

A. No need to pay in cash, no free delivery.
B. Must pay in cash, no free delivery.
C. No need to pay in cash, free delivery.
D. Must pay in cash, free delivery.

Correct Answer: B
Section:
Explanation:
According to the syllabus, decision tables are a technique that shows combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects). The answer B is correct because it matches the rules in the decision table for the given test case. The other answers are incorrect because they do not match the rules in the decision table for the given test case.

QUESTION 132
Given the following state transition diagram where SS is the start state:



Which of the following answers describes a test case that only uses valid transitions to exercise all states, using the minimum number of transitions?

A. SS-S3-SS-S1-S3-S2.
B. SS-S3-S2-S1.
C. SS-S1-S2-S3.
D. SS-S1-S3-S2.

Correct Answer: D
Section:
Explanation:
According to the syllabus, state transition testing is a technique that tests the behaviour of a system or component based on its transitions between states. A state transition diagram is a graphical representation of the states and transitions of a system or component. A test case that uses valid transitions to exercise all states should cover all the nodes and edges in the diagram. The answer D is correct because it covers all the four states (SS, S1, S2, S3) and all the five transitions (SS-S1, S1-S3, S3-S2, S2-SS, SS-S3) in the diagram. The other answers are incorrect because they either miss some states or transitions or use more transitions than necessary.

QUESTION 133
Which statement about deriving test cases from a use case is TRUE?

A. Test cases are designed to exercise the defined behaviours.
B. There should only be one test case for each use case.
C. Test cases can only be derived if there is an activity diagram.
D. It is not possible to measure use case test coverage.

Correct Answer: A
Section:
Explanation:
According to the syllabus, use case testing is a technique that tests the functionality of a system using scenarios based on use cases. A use case is a description of how an actor interacts with a system to achieve a goal. A use case defines the expected behaviour and possible variations of the system under test. The answer A is correct because test cases are designed to exercise the defined behaviours of the use case and its variations. The other answers are incorrect because they are not true statements about use case testing. There can be multiple test cases for each use case, depending on the complexity and variations of the use case. Test cases can be derived from other sources besides activity diagrams, such as textual descriptions or flow charts. It is possible to measure use case test coverage by comparing the number of executed test cases with the number of defined test cases for each use case.

QUESTION 134
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 decisions are based on a numeric value.
D. Checking that at least 100% decision coverage has been achieved, as this guarantees 100% statement coverage.

Correct Answer: C

Section:
Explanation:
According to the syllabus, decision coverage is a technique that measures the percentage of decision outcomes that have been exercised by a test suite. A decision outcome is the result of evaluating a boolean expression in a control structure, such as an if statement or a switch statement. The goal of decision coverage testing is to cover and validate all the accessible source code by checking and ensuring that each branch of every possible decision point is executed at least once. The answer C is incorrect because it is not a valid use of decision coverage testing. Decision coverage testing does not check that all decisions are based on a numeric value, as there can be other types of values involved in boolean expressions, such as strings or booleans. The other answers are correct because they are valid uses of decision coverage testing. Checking that all decisions have been exercised in a single program or a business process can help to identify defects or missing functionality in the code or process logic. Checking that at least 100% decision coverage has been achieved can help to ensure that all reachable code has been executed and that 100% statement coverage has been achieved as well.

QUESTION 135
Which of the following is an approach that can be used for exploratory testing?

A. Time-boxed test sessions are created, during which a tester uses a test charter containing test objectives to guide the testing.
B. A tester methodically executes tests from a list of possible failures, based on experience, defect and failure data.
C. A tester analyses, designs and implements tests based on external rules and standards.
D. Tests are designed based on the guidance of stakeholders and experts outside the test team.

Correct Answer: A
Section:
Explanation:
According to the syllabus, exploratory testing is a technique that involves simultaneous learning, test design and test execution. A tester dynamically designs and executes tests based on their knowledge, exploration of the test item and the results of previous tests. A time-boxed test session is a fixed period of time during which testing is performed. A test charter is a document that contains test objectives, scope, risks and other information to guide the testing. The answer A is correct because it describes an approach that can be used for exploratory testing. The other answers are incorrect because they describe other techniques, such as error guessing (B), compliance testing and user acceptance testing (D).

QUESTION 136
You are testing a mobile app that displays a person's status in respect of Covid-19. There are five possibilities: Fully Vaccinated, Partly Vaccinated, Infected & Recovered. Last Tested Positive or Last Tested Negative. You have found that, after receiving data about successful administration of a second injection, the person's status has not changed from Partly Vaccinated to Fully Vaccinated, although it should have done. The project uses a popular proprietary defect management tool where you have drafted an incident report with the following information:
* Test id., test environment used and date/time of run
* Expected and actual results with steps to reproduce.
* Severity level 4 (Critical - an entire functional area is unusable)
* Version data for the application under test and the testware that was used
Which one of the following important items of information is missing?

A. Recommendations
B. Name of Tester
C. Priority
D. Change History

Correct Answer: C
Section:
Explanation:
Priority is an important item of information that is missing from the incident report. Priority indicates the importance or urgency of resolving the incident, based on the business needs and the impact on the stakeholders. Priority is usually assigned by the project manager or the customer, and it helps to determine the order in which incidents should be addressed. Priority may differ from severity, which indicates the degree of impact of the incident on the system or the component under test. Severity is usually assigned by the tester or the developer, and it helps to assess the risk of the incident. For example, an incident may have a high severity but a low priority, if it affects a critical function but only occurs in a rare situation. Conversely, an incident may have a low severity but a high priority, if it affects a minor function but occurs frequently or affects many users. Therefore, both priority and severity are useful information for incident management and resolution.
The other options are not essential information for the incident report, although they may be helpful or desirable in some cases. Recommendations are suggestions or proposals for resolving the incident, which may be

provided by the tester, the developer, or other stakeholders. However, recommendations are not mandatory, and they may not be feasible or acceptable in some situations. Name of tester is the identifier of the person who reported the incident, which may be useful for communication or accountability purposes. However, name of tester is not critical, and it may be replaced by other identifiers, such as email address, employee number, or role. Change history is the record of the changes made to the incident report, such as status, resolution, or comments. Change history is valuable for tracking and auditing purposes, but it is not part of the initial incident report, rather it is updated as the incident progresses.

QUESTION 137
What one of the following would be a typical objective of running a pilot project when introducing a new tool into an organisation?

A. To establish whether the tool is available for a free trial period (and for how long).
B. To provide training, coaching, and mentoring for users of the tool.
C. To develop a clear set of requirements and objective criteria against which the tool can be evaluated.
D. To evaluate how the tool fits with existing processes and practices, and determining what would need to change.

Correct Answer: D
Section:
Explanation:
After completing the tool selection and a successful proof-of-concept, introducing the selected tool into an organization generally starts with a pilot project, which has the following objectives12:
Gaining in-depth knowledge about the tool, understanding both its strengths and weaknesses
Evaluating how the tool fits with existing processes and practices, and determining what would need to change
Assessing whether the benefits will be achieved at reasonable cost
Identifying and resolving any technical and organizational issues before deploying the tool on a larger scale
Providing training, coaching, and mentoring for users of the tool
Defining metrics to measure the success of the tool deployment
Therefore, the option D is the most typical objective of running a pilot project, as it is essential to ensure the compatibility and suitability of the tool with the current way of working, and to identify any potential risks or challenges that may arise from the tool introduction12.
The other options are not typical objectives of running a pilot project, because they are either part of the tool selection process, or the outcomes of the pilot project. They are:
To establish whether the tool is available for a free trial period (and for how long) (A).This is part of the tool selection process, which involves gathering information about the tool features, costs, support, and availability12.This should be done before running a pilot project, as it helps to narrow down the list of potential tools and to conduct a proof-of-concept12.
To provide training, coaching, and mentoring for users of the tool (B).This is an outcome of the pilot project, which aims to increase the competence and confidence of the tool users and to facilitate the tool adoption12.This should be done during and after the pilot project, as it helps to ensure the effective and efficient use of the tool and to address any issues or feedback from the users12.
To develop a clear set of requirements and objective criteria against which the tool can be evaluated .This is also part of the tool selection process, which involves defining the tool requirements based on the needs and expectations of the stakeholders, and establishing the evaluation criteria based on the benefits and risks of the tool12.This should also be done before running a pilot project, as it helps to compare and rank the potential tools and to select the most suitable one12.
ISTQB Foundation Level 2018 syllabus
Chapter 6 - Tool Support for Testing | PPT

QUESTION 138
Which of the following statements about test estimation approaches is CORRECT?

A. The Wideband Delphi estimation technique is an example of the expert-based approach
B. The Wideband Delphi estimation technique is an example of the metrics-based approach
C. Burndown chads used in Agile development is an example of the expert-based approach
D. Burndown charts used in Agile development is an example of the risk-based approach

Correct Answer: A
Section:
Explanation:

The correct answer is A, as it states that the Wideband Delphi estimation technique is an example of the expert-based approach.The Wideband Delphi estimation technique is a method of estimating testing effort or duration by using a structured group process that involves multiple experts2.The experts provide their estimates independently and anonymously, then compare and discuss them until they reach a consensus2.This technique is an example of the expert-based approach, which is an approach that relies on the knowledge and experience of experts to estimate testing activities2. Option B is incorrect, as it states that the Wideband Delphi estimation technique is an example of the metrics-based approach.The metrics-based approach is an approach that uses historical data and mathematical formulas to estimate testing activities2. This approach does not involve experts or group processes. Option C is incorrect, as it states that burndown charts used in Agile development is an example of the expert-based approach.Burndown charts are graphical tools that show the amount of work remaining versus time in an Agile project2. They are used to monitor and control testing progress and quality. They are not examples of the expert-based approach, as they do not rely on experts' opinions or estimates. Option D is incorrect, as it states that burndown charts used in Agile development is an example of the risk-based approach.The risk-based approach is an approach that uses risk analysis to prioritize and estimate testing activities2. This approach involves identifying and assessing risks based on their likelihood and impact.It does not involve burndown charts or Agile development.
Reference:2, Section 2.6

QUESTION 139
Which of the following BEST describes a test summary report for executive-level employees?

A. The report Is detailed and Includes specific information on defects and trends
B. The report is detailed and includes a status summary of defects by priority or budget
C. The report is high-level and includes specific information on defects and trends
D. The report is high-level and includes a status summary of detects by priority or budget

Correct Answer: D
Section:
Explanation:
The correct answer is D, as it describes a test summary report for executive-level employees.A test summary report is a document that summarizes the results and evaluation of testing activities for a specific activity or phase3.It may have different levels of detail and content depending on the intended audience and purpose3.A test summary report for executive-level employees is typically high-level and includes a status summary of defects by priority or budget3.This type of report provides a concise overview of the quality and progress of testing without going into too much detail or technical information3. Option A is incorrect, as it describes a test summary report for technical-level employees.A test summary report for technical-level employees is typically detailed and includes specific information on defects and trends3.This type of report provides a comprehensive analysis of the quality and progress of testing with relevant data and metrics3. Option B is incorrect, as it describes a test summary report that is neither suitable for executive-level nor technical-level employees.A test summary report that is detailed and includes a status summary of defects by priority or budget is too detailed for executive-level employees and too vague for technical-level employees3. Option C is incorrect, as it describes a test summary report that is neither suitable for executive-level nor technical-level employees.A test summary report that is high-level and includes specific information on defects and trends is too high-level for technical-level employees and too specific for executive-level employees3.
Reference:3, Section 2.7

QUESTION 140
Which of the following BEST defines risk level?

A. Risk level Is calculated by adding the probabilities of all planned risks to a project
B. Risk level Is determined by the likelihood of an event happening and the impact or harm from that event
C. Risk level Is calculated by dividing the sum of all known risks by the sum of all unknown risks
D. Risk level is determined by calculating the absolute value of the sum of all potential issues that may occur on the project

Correct Answer: B
Section:
Explanation:
The correct answer is B, as it defines risk level correctly.Risk level is determined by the likelihood of an event happening and the impact or harm from that event4.Likelihood is the probability or frequency of the event occurring4.Impact is the severity or consequence of the event occurring4.Risk level can be calculated by multiplying likelihood and impact, or by using a risk matrix that assigns risk levels based on different combinations of likelihood and impact4. Option A is incorrect, as it does not define risk level correctly.Risk level is not calculated by adding the probabilities of all planned risks to a project, but by assessing the likelihood and impact of each individual risk4. Option C is incorrect, as it does not define risk level correctly.Risk level is not calculated by dividing the sum of all known risks by the sum of all unknown risks, but by assessing the likelihood and impact of each individual risk4. Option D is incorrect, as it does not define risk level correctly.Risk level is not determined by calculating the absolute value of the sum of all potential issues that may occur on the project, but by assessing the likelihood and impact of each individual risk4.
Reference:4, Section 2.8

QUESTION 141
Which of the following is MOST likely to be an example of a PROJECT risk?

A. A system architecture may not support some non-functional requirements
B. A computation is not always performed correctly in some situations
C. Team members' skills may not be sufficient for the assigned work
D. Specific modules do not adequately meet their intended functions according to the user

Correct Answer: C
Section:
Explanation:
The correct answer is C, as it is an example of a project risk.A project risk is a risk that affects the management and control of the testing project, such as planning, budget, schedule, resources, quality, or scope1.Team members' skills may not be sufficient for the assigned work is a project risk, as it affects the quality and efficiency of the testing activities1. Option A is incorrect, as it is an example of a product risk.A product risk is a risk that affects the quality of the software product under test, such as functionality, reliability, usability, security, or performance1.A system architecture may not support some non-functional requirements is a product risk, as it affects the reliability and performance of the software product1. Option B is incorrect, as it is an example of a defect.A defect is a flaw in the software product that causes it to produce incorrect or unexpected results or behavior1.A computation is not always performed correctly in some situations is a defect, as it causes the software product to produce incorrect results1. Option D is incorrect, as it is an example of a defect.A defect is a flaw in the software product that causes it to produce incorrect or unexpected results or behavior1.Specific modules do not adequately meet their intended functions according to the user is a defect, as it causes the software product to produce unexpected behavior1.
Reference:1, Section 2.8

QUESTION 142
You are testing an e-commerce system which sporadically fails to properly manage customers' shopping carts. You have stressed the urgency of this situation to the development manager and development team and they recognize the priority in getting this resolved. The development team is waiting for more information in your defect report in order to help resolve this failure.
Given the following items of information:
1. The expected results
2. The actual results
3. The urgency and priority to fix this
4. The date and author of the defect report
5. A description of the defect in order to reproduce, including screenshots and database dumps
Which of the items would be MOST useful to include in the defect report?

A. 1, 2, 5
B. 1,2, 3, 4, 5
C. 1, 2, 4
D. 3, 4

Correct Answer: A
Section:
Explanation:
The correct answer is A, as it includes the items that would be most useful to include in the defect report.A defect report is a document that describes and records any anomaly found during testing2. A defect report typically includes information such as:
The expected results
The actual results
A description of the defect in order to reproduce, including screenshots and database dumps
These items are useful to include in the defect report, as they help the developers to understand and fix the defect2. Option B is incorrect, as it includes an item that would not be useful to include in the defect report.The urgency and priority to fix this is not useful to include in the defect report, as it is determined by the test manager or other stakeholders based on various factors such as severity, impact, risk, and business value2. Option C is incorrect, as it omits an item that would be useful to include in the defect report.The actual results are useful to include in the defect report, as they show what happened when the defect occurred2. Option D is incorrect, as it omits two items that would be useful to include in the defect report.The expected results and a description of the defect in order to reproduce are useful to include in the defect report, as they show what should have happened and how to replicate the defect2.

Reference:2, Section 2.9

QUESTION 143
What type of testing is important after Migration, retirement or enhancement of an existing system?

A. Regression testing
B. Operational acceptance testing
C. System Testing
D. Maintenance testing

Correct Answer: D
Section:
Explanation:
Maintenance testing is important after migration, retirement or enhancement of an existing system. defines maintenance testing as follows:
Maintenance testing is a type of software testing that occurs after changes have been made to a system in order to ensure that no new defects have been introduced and that the system still meets its original requirements and specifications.
Regression testing (A) is important after any change in the system that may affect existing functionality. Operational acceptance testing (B) is important before deploying a system into production environment. System testing is important after integrating all components of a system into a complete system.

QUESTION 144
You are introducing a new test tool into your organization and planning a pilot project.
What is a MAIN objective of this pilot project?

A. To immediately save cost for current projects in your organisation
B. To show competitors that your organisation is improving its test process
C. To motivate the test team and make testers feel valued
D. To learn more detail about the tool and how it fits with existing processes

Correct Answer: D
Section:
Explanation:
A main objective of a pilot project for introducing a new test tool into an organization is to learn more detail about the tool and how it fits with existing processes. states this as follows:
A pilot project should be conducted before introducing a new test tool into an organization in order to learn more about how to use it effectively and efficiently in your context and how it will interact with other tools and processes.
A pilot project is not meant to immediately save cost for current projects in your organisation (A), as it may require additional resources and time to set up and evaluate the tool. A pilot project is not meant to show competitors that your organisation is improving its test process (B), as it is an internal activity that may not be visible or relevant to external parties. A pilot project is not meant to motivate the test team and make testers feel valued , as it may cause resistance or frustration if the tool is not suitable or easy to use.

QUESTION 145
Which of the following statements is true?

A. 100% branch coverage means 100% statement coverage.
B. 100% statement coverage means 100% branch coverage.
C. 100% branch coverage means 100% statement coverage and vice-versa.
D. It is impossible to achieve 100% statement coverage

Correct Answer: A
Section:

Explanation:
100% branch coverage means 100% statement coverage, but not vice-versa. Branch coverage is a stronger criterion than statement coverage, as it requires that every possible outcome of each decision point is executed at least once.Statement coverage only requires that every executable statement is executed at least once1explains this as follows:
Branch Coverage (also known as Decision Coverage) measures which possible branches in flow control structures are followed. Branch Coverage is a testing method, which aims to ensure that each one of the possible branch from each decision point is executed at least once and thereby ensuring that all reachable code is executed.
Statement Coverage measures the number of statements in the source code that are executed during the test. Statement Coverage is a metric, which is used to calculate and measure the number of statements in the source code which have been executed during testing.
B, C, and D are false statements. 100% statement coverage does not mean 100% branch coverage (B), as there may be some branches that are not covered by the test cases. 100% branch coverage does not mean 100% statement coverage and vice-versa , as branch coverage implies statement coverage but not the other way around. It is possible to achieve 100% statement coverage (D), but it may not be feasible or cost-effective for some systems.

QUESTION 146
Which activity in the fundamental test process includes test script generation?

A. Test closure activities
B. Test analysis and design
C. Test planning and control
D. Test implementation and execution

Correct Answer: D
Section:
Explanation:
Test implementation and execution is the activity in the fundamental test process that includes test script generation.Test script generation is the process of creating executable test cases from test conditions and test data2defines this activity as follows:
Test implementation and execution has the following major tasks:
Develop and prioritize test cases, creating test data and writing test procedures.
Check test environment has been set up correctly.
Execute test cases, evaluate results and document deviations from expected results.
Use exit criteria to report on suitability of system under test.
Test closure activities (A) include finalizing and archiving test results, evaluating the test process, identifying areas for improvement, and celebrating achievements. Test analysis and design (B) include reviewing test basis, identifying test conditions, designing high-level test cases, and defining exit criteria. Test planning and control include defining test objectives, scope, strategy, resources, schedule, risks, and metrics.

QUESTION 147
The ISTQB fundamental test process consists of 5 main activities To which of these belongs the task 'Identifying necessary test data'?

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

Correct Answer: D
Section:
Explanation:
Test analysis and design is the activity in the fundamental test process that includes identifying necessary test data.Test data are the inputs that are used to execute the test cases and verify the expected results2defines this activity as follows:
Test analysis and design has the following major tasks:
Reviewing the test basis (such as requirements, risk analysis reports, design documents or code).
Identifying test conditions based on analysis of test items, specifications, behavior and structure of the software.
Designing high-level test cases based on test conditions and designing techniques.

Evaluating testability of requirements and system under test.
Defining exit criteria.
Evaluating exit criteria and reporting (A) is part of the test closure activities, where the results of testing are evaluated against the defined objectives. Test implementation and execution (B) is where the test cases are executed using the identified test data and deviations from expected results are documented. Test planning and control is where the overall approach and resources for testing are defined and monitored.

QUESTION 148
Which of the following BEST explains a drawback of Independent testing?

A. An independent test team may be isolated from the rest of the development and project team
B. Due to their differing backgrounds and perspectives, an independent lest team may discover defects which the developers did not uncover
C. An independent test team may possess specializations in specific test types such as usability or security which detract from the overall effectiveness of the test team
D. Having the business organization participate as an independent test team can hurt the overall testing effort since business participants are often not trained nor experienced in testing

Correct Answer: A
Section:
Explanation:
The correct answer is A, as it explains a drawback of independent testing.Independent testing is a type of testing that is performed by a person or team that is not involved in the development of the software1.Independent testing can have many benefits, such as providing an objective view of the software quality, reducing conflicts of interest, and increasing the effectiveness and efficiency of testing1.However, independent testing can also have some drawbacks, such as being isolated from the rest of the development and project team1.This can lead to communication problems, lack of collaboration, and misunderstanding of requirements and expectations1. Option B is incorrect, as it does not explain a drawback of independent testing, but a benefit.Due to their differing backgrounds and perspectives, an independent test team may discover defects that the developers did not uncover1.This can improve the quality of the software and reduce the risk of failures in operation1. Option C is incorrect, as it does not explain a drawback of independent testing, but a strength.An independent test team may possess specializations in specific test types such as usability or security that enhance the overall effectiveness of the test team1.These test types may require specialized skills, tools, or environments that are not available to the developers or other testers1. Option D is incorrect, as it does not explain a drawback of independent testing, but a misconception.Having the business organization participate as an independent test team can benefit the overall testing effort since business participants can provide valuable feedback on the user requirements, expectations, and satisfaction1.However, business participants are often not trained nor experienced in testing, so they should not be the only source of independent testing1.They should be supported by professional testers who can apply appropriate test techniques and methods1.
Reference:1, Section 2.2

QUESTION 149
Which of the following tasks Is MOST LIKELY to be performed by the tester?

A. Develop a test strategy and test policy for the organization
B. Introduce suitable metrics for measuring test progress
C. Promote and advocate the test team within the organization
D. Create the detailed test execution schedule

Correct Answer: D
Section:
Explanation:
The correct answer is D, as it describes a task that is most likely to be performed by the tester.A tester is a person who performs testing activities such as planning, designing, executing, and evaluating tests2.Creating the detailed test execution schedule is one of these activities, as it involves defining the order and timing of test cases to be executed based on various factors such as dependencies, risks, priorities, and resources2. Option A is incorrect, as it describes a task that is most likely to be performed by the test manager.A test manager is a person who leads and manages testing activities such as establishing test policies and strategies, managing test teams and stakeholders, monitoring and controlling test progress and quality, and reporting on test results2.Developing a test strategy and test policy for the organization is one of these activities, as it involves defining the overall approach and objectives for testing at an organizational level2. Option B is incorrect, as it describes a task that is most likely to be performed by the test analyst.A test analyst is a person who supports testing activities such as analyzing requirements and risks, designing and prioritizing tests, evaluating test coverage and traceability, and reviewing work products2.Introducing suitable metrics for measuring test progress is one of these activities, as it involves defining and collecting quantitative data to monitor and control various aspects of testing such as efficiency, effectiveness, quality, and maturity2. Option C is incorrect, as it describes a task that is most likely to be performed by the test leader.A test leader is a person who coordinates testing activities such as planning and estimating tests, allocating tasks and resources, supervising test execution and evaluation, communicating with stakeholders, and resolving issues2.Promoting and advocating the test team within the organization is one of these activities, as it involves demonstrating the value and benefits of testing to other parties such as management, developers, customers, and users2.
Reference:2, Section 2.3