ISTQB

Exam Questions ISTQB-CTFL

ISTQB-Foundation Level Exam

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1
Consider the following testing levels:
1) Component Testing
2) Integration Testing
3) System Testing
4) Acceptance Testing
Which of the following statements is true?

A. Integration and system testing are applicable when V-model is used.Component and acceptance testing are applicable when iterative development models are used.
B. All the testing levels are applicable to V-model for software developmen
C. Only acceptance testing is applicable for iterative models.
D. Acceptance testing is applicable for all software development model
E. Component and system testing are applicable only for the V-model.
F. All testing levels are applicable, independent of which software development life-cycle process (V-mode
G. iterative, incremental) is used.

Answer: D

Explanation:
All testing levels are applicable, independent of which software development life-cycle process (V-model, iterative, incremental) is used. Testing levels are defined based on the scope and objectives of testing, not on the software development model. Component testing, integration testing, system testing and acceptance testing are common testing levels that can be applied to any software development model, as long as they are planned and executed properly. The V-model is a software development model that emphasizes the relationship between each development phase and its corresponding testing phase. Iterative and incremental models are software development models that divide the development process into smaller cycles or iterations, where each iteration produces a working version of the software that can be tested and evaluated. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 18.

NEW QUESTION 2
Who of the following has the best knowledge to decide what tests in a test project should be automated?

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

Answer: D

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

NEW QUESTION 3
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

Answer: B

Explanation:
A correct reason to apply test automation is when there are a lot of repetitive testing tasks. Test automation is the use of software tools or scripts to perform or support testing activities, such as test case execution, test result comparison, test data generation, etc. Test automation can be beneficial when there are a lot of repetitive testing tasks that need to be performed frequently or consistently, such as regression testing, performance testing, load testing, etc. Test automation can help to save time and effort, increase reliability and accuracy, and improve coverage and efficiency of testing. The other options are not correct reasons to apply test automation. When a new test automation tool is launched is not a reason to apply test automation, but rather a factor for choosing a test automation tool. When it is easy to automate is not a reason to apply test automation, but rather a factor for evaluating the feasibility of test automation. When it is cheap to buy test automation tools is not a reason to apply test automation, but rather a factor for estimating the cost and benefit of test automation. Verified References: A Study Guide to the ISTQB®
Foundation Level 2018 Syllabus - Springer, page 10.

NEW QUESTION 4
A software system checks age in order to determine which welcome screen to display. Age groups are:
Group I: 0-12
Group II; 13-18 Group III: over 18
Which of the below represent boundary values?

A. (-1.0.12.13.18,19)
B. (-1.0,11.12.13,14,18.19)
C. (0.12.13.18.19)
D. (4.5.15.20)

Answer: A

Explanation:
A correct list of boundary values for the age input should include the minimum and maximum values of each age group (0, 12, 13, 18), as well as the values just below and above each boundary (-1, 19). Boundary value analysis is a test design technique that involves testing the values at or near the boundaries of an input domain or output range, as these values are more likely to cause errors than values in the middle. Option A satisfies this condition, as it has all six boundary values (-1, 0, 12, 13, 18, 19). Option B has two values from the same equivalence class (12 and 13), option C has only four boundary values (0, 12, 18, 19), and option D has no boundary values at all. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 34.

NEW QUESTION 5
The four test levels used in ISTQB syllabus are:
* 1. Component (unit) testing
* 2. Integration testing
* 3. System testing
* 4. Acceptance testing
An organization wants to do away with integration testing but otherwise follow V-model. Which of the following statements is correct?

A. It is allowed as organizations can decide on men test levels to do depending on the context of the system under test
B. It is allowed because integration testing is not an important test level arc! can be dispensed with.
C. It is not allowed because integration testing is a very important test level and ignoring i: means definite poor product quality
D. It is not allowed as organizations can't change the test levels as these are chosen on the basis of the SDLC (software development life cycle) model

Answer: D

Explanation:
The V-model is a software development life cycle model that defines four test levels that correspond to four development phases: component (unit) testing with component design, integration testing with architectural design, system testing with system requirements, and acceptance testing with user requirements. The V-model emphasizes the importance of verifying and validating each phase of development with a corresponding level of testing, and ensuring that the test objectives, test basis, and test artifacts are aligned and consistent across the test levels. Therefore, an organization that wants to follow the V-model cannot do away with integration testing, as it would break the symmetry and completeness of the V-model, and compromise the quality and reliability of the software or system under test. Integration testing is a test level that aims to test the interactions and interfaces between components or subsystems, and to detect any defects or inconsistencies that may arise from the integration of different parts of the software or system. Integration testing is essential for ensuring the functionality, performance, and compatibility of the software or system as a whole, and for identifying and resolving any integration issues early in the development process. Skipping integration testing would increase the risk of finding serious defects later in the test process, or worse, in the production environment, which would be more costly and difficult to fix, and could damage the reputation and credibility of the organization. Therefore, the correct answer is D.
The other options are incorrect because:
? A. It is not allowed as organizations can decide on the test levels to do depending on the context of the system under test. While it is true that the choice and scope of test levels may vary depending on the context of the system under test, such as the size, complexity, criticality, and risk level of the system, the organization cannot simply ignore or skip a test level that is defined and required by the chosen software development life cycle model. The organization must follow the principles and guidelines of the software development life cycle model, and ensure that the test levels are consistent and coherent with the development phases. If the organization wants to have more flexibility and adaptability in choosing the test levels, it should consider using a different software development life cycle model, such as an agile or iterative model, that allows for more dynamic and incremental testing approaches.
? B. It is not allowed because integration testing is not an important test level and can be dispensed with. This statement is false and misleading, as integration testing is a very important test level that cannot be dispensed with. Integration testing is vital for testing the interactions and interfaces between components or subsystems, and for ensuring the functionality, performance, and compatibility of the software or system as a whole. Integration testing can reveal defects or inconsistencies that may not be detected by component (unit) testing alone, such as interface errors, data flow errors, integration logic errors, or performance degradation. Integration testing can also help to verify and validate the architectural design and the integration strategy of the software or system, and to ensure that the software or system meets the specified and expected quality attributes, such as reliability, usability, security, and maintainability. Integration testing can also provide feedback and confidence to the developers and stakeholders about the progress and quality of the software or system development. Therefore, integration testing is a crucial and indispensable test level that should not be skipped or omitted.
? C. It is not allowed because integration testing is a very important test level and
ignoring it means definite poor product quality. This statement is partially true, as integration testing is a very important test level that should not be ignored, and skipping it could result in poor product quality. However, this statement is too strong and absolute, as it implies that integration testing is the only factor that determines the product quality, and that ignoring it would guarantee a poor product quality. This is not necessarily the case, as there may be other factors that affect the product quality, such as the quality of the requirements, design, code, and other test levels, the effectiveness and efficiency of the test techniques and tools, the competence and experience of the developers and testers, the availability and adequacy of the resources and environment, the management and communication of the project, and the expectations and satisfaction of the customers and users. Therefore, while integration testing is a very important test level that should not be skipped, it is not the only test level that matters, and skipping it does not necessarily mean definite poor product quality, but rather a higher risk and likelihood of poor product quality.
References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.3, pages 16-18; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 38-39; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 104, page 36.

NEW QUESTION 6
Which of the following is a function of a dynamic analysis tool?

A. Provide support for traceability of tests, test results and incidents to source documents
B. Monitor the allocation, use and de-allocation of memory during run-time of a program
C. Execute programs step-by-step in order to reproduce failures and find corresponding defects
D. Provide support for release of baselines consisting of configuration items

Answer: B

Explanation:
A dynamic analysis tool is a tool that performs analysis of a software product based on its behavior during execution. A dynamic analysis tool can monitor various aspects of a program's run-time performance, such as memory usage, CPU load, response time, or resource leaks. A dynamic analysis tool can monitor the allocation, use and de- allocation of memory during run-time of a program, which can help detect defects such as memory leaks, buffer overflows, or memory corruption. A dynamic analysis tool cannot provide support for traceability of tests, test results and incidents to source documents, as this is a function of a test management tool. A dynamic analysis tool cannot execute programs step-by-step in order to reproduce failures and find corresponding defects, as this is a function of a debugging tool. A dynamic analysis tool cannot provide support for release of baselines consisting of configuration items, as this is a function of a configuration management tool. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 6, page 56-57.

NEW QUESTION 7

Which of the following coverage criteria results in the highest coverage for state transition based test cases?

A. Can't be determined
B. Covering all transitions at least once
C. Covering only start and end states
D. Covering all states at least once

Answer: B

Explanation:

Covering all transitions at least once is the highest coverage criterion for state transition based test cases, because it ensures that every possible change of state is tested at least once. This means that all the events that trigger the transitions, as well as the actions and outputs that result from the transitions, are verified. Covering all transitions at least once also implies covering all states at least once, but not vice versa. Therefore, option D is not the highest coverage criterion. Option C is the lowest coverage criterion, because it only tests the initial and final states of the system or component, without checking the intermediate states or transitions. Option A is incorrect, because the coverage criteria for state transition based test cases can be determined and compared based on the number of transitions and states covered. References = CTFL 4.0 Syllabus, Section 4.2.3, page 49-50.

NEW QUESTION 8

A test engineer finds a defect while testing. After the developer has fixed the defect, the test engineer decides to re-run a complete section of the tests. Which of the following is correct?

A. The lest engineer should not re-run the tests, as they have already been run, and results recorded.
B. The test engineer should not re-run the tests, they should be part of the developer tests.
C. The test engineer should re-run the tests, in order to ensure that new defects have not been introduced by the fix.
D. The test engineer should re-run the tests, because the defect shows that the test cases need to be updated.

Answer: C

Explanation:

The test engineer should re-run the tests, in order to ensure that new defects have not been introduced by the fix. This is also known as regression testing, which is a type of testing that verifies that previously tested software still performs correctly after a change. Regression testing helps to detect any side effects or unintended consequences of a fix or a modification. The other options are incorrect reasons for re-running the tests. The test engineer should not re-run the tests, as they have already been run, and results recorded, because this ignores the possibility of new defects caused by the fix. The test engineer should not re-run the tests, they should be part of the developer tests, because this assumes that developer tests are sufficient and reliable, which may not be true. The test engineer should not re-run the tests, because the defect shows that the test cases need to be updated, because this does not address the impact of the fix on other test cases or functionalities. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 41.

NEW QUESTION 9

A test score indicator for students produces a performance score based on a combination of the number of consecutive hours studied (below 4 hours, 4 to 8 hours. 9 to 12 hours or above 12 hours) and the average intensity of focus on the material during the study time (low, medium or high).
Given the following test cases: hours intensity score
T1 3 low55
T2 14 high 95
T3 9 low75
What is the minimum number of additional test cases that are needed to ensure full coverage of all valid INPUT equivalence partitions?

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

Answer: C

Explanation:

Considering the various valid input equivalence partitions of hours studied and intensity, three additional test cases are needed to fully cover all valid partitions. This would typically include testing combinations that vary both the number of hours and the intensity levels not covered by the initial test cases (ISTQB Main Web).References:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION 10

The following requirement is given “Set X to be the sum of Y and Z”. All the following four implementations have bugs.
Which one of the following bugs can be caught by Static Analysis?

A. int x = 1. int y = 2. int y = 3.X = y=z;
B. int x = 1. int y = 2. int z = 3.X = z-y
C. int x = 1. Int y = 2. Int z = 3.Z = x +y
D. int y = 2 Int z = 3. Y = z+y

Answer: A

Explanation:

Static analysis is a technique that analyzes the source code or other software artifacts without executing them. Static analysis can detect defects such as syntax errors, coding standards violations, potential security vulnerabilities, or logical flaws. Static analysis can catch the bug in the first implementation, as it contains two syntax errors: the variable y is declared twice, and the assignment statement X = y=z is invalid. Static analysis cannot catch the bugs in the other three implementations, as they are logical errors that do not violate any syntax rules, but produce incorrect results. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 3, page 25-26.

NEW QUESTION 10

Which of the following software development models BEST exemplifies a model that does NOT support the principle of early testing?

A. The iterative development model
B. The V-model
C. The Waterfall model
D. The incremental development model

Answer: C

Explanation:
The Waterfall model exemplifies a software development model that does not support the principle of early testing. In the Waterfall model, each phase must be completed before the next begins, which delays testing until after the completion of the earlier phases like requirements gathering and design. This can often result in finding defects later in the development cycle, making them more expensive and time-consuming to fix (ISTQB not-for-profit association) (ISTQB not-for-profit association).References:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf
? ISTQB News Release on CTFL v4.0: https://www.istqb.org/news/posts/istqb-releases-certified-tester-foundation-level-v40-ctfl/

NEW QUESTION 15
A team's test strategy was to invest equal effort in testing each of a system's modules. After running one test cycle, it turned out that most of the critical bugs were detected in one of the system's modules.
Which testing principal suggests a change to the current test strategy for the next test cycle?

A. Pesticide Paradox
B. Early testing
C. Absence-of-errors fallacy
D. Defect clustering

Answer: D

Explanation:
Defect clustering is a testing principle that states that a small number of modules contain most of the defects detected during pre-release testing, or are responsible for most of the operational failures. Defect clustering can be explained by Pareto's principle (also known as the 80-20 rule), which states that approximately 80% of the problems are found in 20% of the modules. Defect clustering suggests a change to the current test strategy for the next test cycle, as it implies that more effort should be allocated to test the modules that have shown high defect density or criticality. Pesticide paradox is another testing principle that states that if the same tests are repeated over and over again, eventually they will no longer find any new defects. Pesticide paradox suggests a change to the current test strategy for the next test cycle, but not based on defect clustering, but rather on test diversity and coverage. Early testing is a testing principle that states that testing activities should start as early as possible in the software development life cycle and should be focused on defined objectives. Early testing does not suggest a change to the current test strategy for the next test cycle, but rather a proactive approach to prevent defects from occurring or propagating. Absence-of-errors fallacy is a testing principle that states that finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations. Absence-of-errors fallacy does not suggest a change to the current test strategy for the next test cycle, but rather a focus on quality attributes and user requirements. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 1, page 9-10.

NEW QUESTION 18
Which of the following definitions is NOT true?

A. Test data preparation tools fill databases, create files or data transmissions to set up test data to be used during the execution of tests.
B. Test execution tools execute test objects using automated test scripts.
C. Test Management tools monitor and report on how a system behaves during the testing activities.
D. Test comparators determine differences between files, databases or test results.

Answer: C

Explanation:
Test Management tools are designed to support the planning, execution, and monitoring of the testing process. They provide features for managing test cases, test runs, tracking defects, and reporting on testing activities. However, the statement in option C describes Test Management tools as monitoring and reporting on the system's behavior during testing activities, which is not accurate. Test Management tools focus on the testing process itself rather than on the behavior of the system under test.
? Test data preparation tools (A) indeed create and manage test data for use during test execution.
? Test execution tools (B) automate the execution of test cases and the comparison of actual outcomes against expected results.
? Test comparators (D) are tools that compare actual outcomes with expected outcomes, highlighting discrepancies.
Therefore, option C is the correct answer as it inaccurately describes the function of Test Management tools.

NEW QUESTION 23
A program is used to control a manufacturing line (turn machines on and off. start and stop conveyer belts, add raw materials to the flow. etc.). Not all actions are possible at all times. For example, there are certain manufacturing stages that cannot be stopped - unless there is an emergency. A tester attempts to evaluate if all such cases (where a specific action is not allowed) are covered by the tests.
Which coverage metric will provide the needed information for this analysis?

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

Answer: D

Explanation:
Branch coverage is a type of structural coverage metric that measures the percentage of branches or decision outcomes that are executed by the test cases. A branch is a point in the code where the control flow can take two or more alternative paths based on a condition. For example, an if-else statement is a branch that can execute either the if-block or the else-block depending on the evaluation of the condition. Branch coverage ensures that each branch is taken at least once by the test cases, and thus reveals the behavior of the software under different scenarios. Branch coverage is also known as decision coverage or all-edges

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

NEW QUESTION 26
Which of the following is the BEST reason for selecting a particular type of software development lifecycle model?

A. The project manager's preference
B. Tester skill level with the software development lifecycle model
C. The project team's overall familiarity with the model
D. The type of product being developed

Answer: D

Explanation:
The choice of a software development lifecycle (SDLC) model is primarily influenced by the type of product being developed. Different products and project requirements may demand different SDLC models to address specific challenges and needs efficiently. For instance, a complex, safety-critical product might best be served by a Waterfall model due to its structured nature and phase dependencies, while a more iterative and incremental model might be suited for projects requiring frequent feedback and changes.References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 2.1 "Software Development Lifecycles".

NEW QUESTION 28
In the newest version of payroll system number of changes were made. As a tester you got a task to perform regression and confirmation tests. Which of the following project activities are related to confirmation testing?

A. Testing due to the application of a new version of the interface
B. Testing that fixes resolved the defects in the search function
C. Testing if a system still works after update of an operating system
D. Testing to ensure the adding of a new functionalities haven't broken existing functions

Answer: B

Explanation:
Confirmation testing, also known as re-testing, is performed to verify that specific defects have been successfully fixed.
Option A: "Testing due to the application of a new version of the interface" would typically involve regression testing, not confirmation testing.
Option B: "Testing that fixes resolved the defects in the search function" fits the description of confirmation testing as it focuses on ensuring that specific issues have been addressed. Option C: "Testing if a system still works after update of an operating system" is an example of regression testing, as it checks the overall system behavior after an update. Option D: "Testing to ensure the adding of new functionalities haven't broken existing functions" is another example of re (ISTQB not-for-profit association) (Udemy) it checks for unintended consequences of new changes.
Therefore, the correct answer is B6†source9†source. References:
? Certified Tester Foundation Level v4.0
? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 29
Which of the following project scenario gives the BEST example where maintenance testing should be triggered?

A. Completion of architecture of the bank system
B. Release of the early draft of the low level project design of an loT application
C. Defect was found in a pre-released version of the customer service application
D. Delivery of the hot fix to mobile operating system and ensuring that it still works

Answer: D

Explanation:
Maintenance testing is triggered by changes such as bug fixes, enhancements, or environmental changes.
Option A: "Completion of architecture of the bank system" is not a typical scenario for maintenance testing, as it describes a design phase rather than an operational change. Option B: "Release of the early draft of the low level project design of an IoT application" is again not suitable for maintenance testing, as it refers to the design phase.
Option C: "Defect was found in a pre-released version of the customer service application" is closer but not quite accurate, as maintenance testing focuses on changes mad (ISTQB not-for-profit association)system is released.
Option D: "Delivery of the hot fix to mobile operating system and ensuring that it still works" is the best example as it directly involves testing after a fix has been implemented. Therefore, the correct answer is D6†source.
References:
? Certified Tester Foundation Level v4.0
? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 30
A test manager defined the following test levels in her test plan; Component, System and Acceptance.
Which Software Development lifecycle is the Test Manager most likely following?

A. V-Model
B. Agile
C. Waterfall
D. Prototyping

Answer: A

Explanation:
The test manager is most likely following the V-model for software development. The V-model is a software development model that defines four testing levels that correspond to four development phases: component testing corresponds to component design, integration testing corresponds to architectural design, system testing corresponds to system requirements specification, and acceptance testing corresponds to user requirements specification. The V-model also defines the test planning and test execution activities for each testing level. Agile is a software development model that follows an iterative and incremental approach, where testing is integrated into each iteration and adapts to changing requirements and feedback. Waterfall is a software development model that follows a sequential and linear approach, where testing is performed after the development phase is completed. Prototyping is a software development model that involves creating a simplified version of the software to elicit user feedback and validate requirements before developing the final product. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 18.

NEW QUESTION 34
Which of the following statements contradicts the general principles of testing?

A. Most defects are found in a small subset of a system's modules.
B. If new defects are to be found we should run the same test set more often.
C. Testing is better if it starts at the beginning of a project.
D. How testing is done, is based on the situation in a particular project.

Answer: B

Explanation:
Statement B contradicts the general principles of testing, because running the same test set more often will not increase the chances of finding new defects, unless there are some changes in the system or environment that affect the test results. Running different test sets with different inputs, outputs or conditions would be more effective in finding new defects. Statements A, C and D are consistent with the general principles of testing. Statement A states that most defects are found in a small subset of a system’s modules, which is true according to the defect clustering principle. Statement C states that testing is better if it starts at the beginning of a project, which is true according to the early testing principle. Statement D states that how testing is done, is based on the situation in a particular project, which is true according to the context-dependent testing principle. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, pages 4-6.

NEW QUESTION 35
Which are the MAIN goals of risk management in a software project?

A. To increase the success probability for the project regardless of costs.
B. To increase focus on preventative processes and to increase satisfaction for the testers.
C. To control contractual problems and minimize the impacts of company policies.
D. To reduce the probability of undesired situations and to reduce the effect of potential impact.

Answer: D

Explanation:
Risk management is a process that identifies, analyzes, evaluates and mitigates risks in a software project. Risks are factors that may negatively affect the quality, schedule, budget or scope of a software project. The main goals of risk management in a software project are to reduce the probability of undesired situations and to reduce the effect of potential impact. This can be achieved by applying various strategies, such as avoidance, transfer, reduction or acceptance. Risk management does not aim to increase the success probability for the project regardless of costs, as this may not be feasible or realistic. Risk management does not aim to increase focus on preventative processes and to increase satisfaction for the testers, as these are secondary or indirect outcomes. Risk management does not aim to control contractual problems and minimize the impacts of company policies, as these are specific types of risks that may not apply to all projects. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 2, page 14-15.

NEW QUESTION 38
Why should you choose a test technique?

A. Because you need to match the way you test to the content of the product under test
B. Because of the time constraints that usually accompany a test project
C. Because this way you cover the full scope of the product's functionality
D. Because choosing a test technique is a common practice in software testing

Answer: A

Explanation:
You should choose a test technique because you need to match the way you test to the content of the product under test. A test technique is a method or process for deriving and selecting test cases based on some criteria or rules. Different test techniques are suitable for different types of software products, depending on their characteristics, functionalities, requirements, specifications, risks, etc. Choosing a test technique helps to ensure that the test cases are relevant, effective, and efficient for the product under test. The other options are not correct reasons to choose a test technique. Time constraints are not a factor for choosing a test technique, but rather for prioritizing or optimizing testing activities. Covering the full scope of the product’s functionality is not a guarantee of choosing a test technique, but rather a goal of testing. Choosing a test technique is not a common practice in software testing, but rather a professional skill and responsibility. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 31.

NEW QUESTION 42
While repotting a defect, which attribute indicates the degree of impact that the defect has on the system?

A. Priority
B. Severity

C. Status
D. Description

Answer: B

Explanation:
In defect reporting, the attribute that indicates the degree of impact that the defect has on the system is the severity. Severity reflects the seriousness of the defect in terms of its impact on the operation of the system, ranging from minor issues that do not significantly affect the system's functionality to critical defects that can cause system failure. Therefore, option B is the correct answer.

NEW QUESTION 46
Which test approach will best fit a new project, with little documentation and high probability for bugs?

A. Exploratory testing
B. Requirements based testing
C. Metric based approach
D. Regression testing

Answer: A

Explanation:
Exploratory testing is an approach to testing that emphasizes learning, test design and test execution at the same time. Exploratory testing relies on the tester's skills, creativity and intuition to explore the software under test and discover defects. Exploratory testing is suitable for a new project with little documentation and high probability for bugs, as it can help uncover unknown requirements, assumptions and risks. Exploratory testing is not requirements based testing, which is an approach to testing that derives test cases from documented requirements or specifications. Requirements based testing is not feasible for a new project with little documentation, as it requires clear and complete requirements to be available. Exploratory testing is not metric based approach, which is an approach to testing that uses quantitative measures to monitor and control the testing process and evaluate the quality of the software product. Metric based approach is not effective for a new project with high probability for bugs, as it may not capture all aspects of quality and may lead to false confidence or unrealistic expectations. Exploratory testing is not regression testing, which is an approach to testing that verifies that previously tested software still performs correctly after changes. Regression testing is not relevant for a new project with no previous versions or baselines. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 5, page 47-48.

NEW QUESTION 47
Given the following state model of sales order software: SEE ATTACHMENT
Which of the following sequences of tran-sitions provides the highest level of tran-sition coverage for the model (assuming you can start in any state)?

A. IN PRODUCTION -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED -> PLACED
B. IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION
C. PLACED -> IN PRODUCTION -> SHIPPED -> CANCELLED -> PLACED
D. PLACED -> CANCELLED -> PLACED -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED

Answer: B

Explanation:
State transition testing is a black-box testing technique where test cases are designed to cover states and transitions of a state machine.
Given the state model with the following transitions:
? PLACED -> IN PRODUCTION
? IN PRODUCTION -> CANCELLED
? IN PRODUCTION -> SHIPPED
? SHIPPED -> INVOICED
? INVOICED -> CANCELLED
? CANCELLED -> PLACED
To cover all transitions at least once, we need to create a sequence that covers all six transitions.
Option A: IN PRODUCTION -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED -> PLACED- Misses SHIPPED -> INVOICED and INVOICED -> CANCELLED transitions.
Option B: IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION- Covers all transitions.
Option C: PLACED -> IN PRODUCTION -> SHIPPED -> CANCELLED (ISTQB not-for-profit association) (Udemy)sses INVOICED -> CANCELLED transition.
Option D: PLACED -> CANCELLED -> PLACED -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED- Misses SHIPPED -> INVOICED and INVOICED ->
CANCELLED transitions.
Given these, Option B covers all the transitions6†source9†source. References:
? Certified Tester Foundation Level v4.0
? 10 Sample Exams ISTQB Foundation Level (CTFL) v4.0

NEW QUESTION 49
In maintenance testing, what is the relationship between impact analysis and regression testing?

A. Impact analysis requires a regression testing for only the tests that have detected faults in previous SW release
B. There is no relationship between impact analysis and regression testing.
C. Impact analysis requires a regression testing for all program elements which were newly integrated (new functionalities).
D. The impact analysis is used to evaluate the amount of regression testing to be performed.

Answer: D

Explanation:
In maintenance testing, the relationship between impact analysis and regression testing is that the impact analysis is used to evaluate the amount of regression testing to be performed. Maintenance testing is a type of testing that is performed on an existing software product after it has been delivered or deployed, in order

to ensure that it still meets its requirements and functions correctly after a change or a modification. Maintenance testing can be triggered by various reasons, such as corrective maintenance (fixing defects), adaptive maintenance (adapting to new environments), perfective maintenance (improving performance), preventive maintenance (avoiding future problems), etc. Impact analysis is a technique that is used to assess the extent and nature of changes introduced by maintenance activities on the software product or project. Impact analysis helps to identify which parts of the software product are affected by the changes, which parts need to be modified or updated accordingly, which parts need to be retested or verified for correctness or compatibility, etc. Regression testing is a type of testing that verifies that previously tested software still performs correctly after a change or a modification. Regression testing helps to detect any side effects or unintended consequences of maintenance activities on the software product's functionality or quality. Regression testing can be performed at various levels and scopes depending on the impact analysis results. Therefore, in maintenance testing, impact analysis is used to evaluate the amount of regression testing to be performed. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 20.

NEW QUESTION 54
During system testing phase of a word processor, a tester finds that on opening a file from a particular set of files, which are part of a critical workflow, the word processor crashes. Which of the following is the next step the tester should take poor to recording the deviation?

A. Try to recreate the incident before reporting
B. Try to identify the code fragment causing the problem
C. Send an email to the developer and not report the bug
D. Report the incident as is without any further action

Answer: A

Explanation:
An incident is any event that occurs during testing that requires investigation. An incident report is a document that records the details of an incident. The next step the tester should take prior to recording the deviation is to try to recreate the incident before reporting. This can help confirm that the incident is reproducible and not caused by a random or external factor. This can also help gather more information about the incident, such as the steps to reproduce it, the expected and actual results, the severity and priority of the incident, or any screenshots or logs that can illustrate the incident. Trying to identify the code fragment causing the problem is not the next step the tester should take prior to recording the deviation, as this is a debugging activity that is usually performed by developers after receiving the incident report. Sending an email to the developer and not reporting the bug is not the next step the tester should take prior to recording the deviation, as this is an informal and unstructured way of communicating incidents that can lead to confusion, inconsistency or loss of information. Reporting the incident as is without any further action is not the next step the tester should take prior to recording the deviation, as this can result in incomplete or inaccurate incident reports that can hamper the investigation and resolution of incidents. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33.

NEW QUESTION 55
As the last stage of a test cycle of an embedded device, you are performing exploratory testing. You observed that some character. (A, X and Z) sent via a serial port to the device do not get registered on the device whereas they should be. You suspect that this could be due to a wrong configuration of the "bit parity" parameter.
Which of the following items of an incident report would you be UNABLE to write down based on this information?

A. Expected result
B. Test case identifier
C. Test setup details
D. Actual result

Answer: B

Explanation:
An incident report is a document that records the details of an incident. An incident report typically contains the following items:
? Identifier: A unique identifier for the incident report
? Summary: A concise summary of the incident
? Description: A detailed description of the incident, including the steps to reproduce it, the expected and actual results, and any relevant screenshots or logs
? Severity: The degree of impact that the incident has on the system
? Priority: The level of urgency for resolving the incident
? Status: The current state of the incident, such as new, open, resolved, closed, etc.
? Resolution: The action taken to resolve the incident, such as fix, workaround, reject, etc. Based on the information given in the question, the tester would be able to write down all of these items except for the test case identifier. A test case identifier is a unique identifier for a test case that is used to link it to other test artifacts, such as test plans, test scripts, test results or incident reports. However, since the tester is performing exploratory testing, there is no predefined test case that can be associated with the incident. Exploratory testing is an approach to testing that emphasizes learning, test design and test execution at the same time. Exploratory testing relies on the tester's skills, creativity and intuition to explore the software under test and discover defects. Exploratory testing does not use formal test cases or scripts, but rather uses test charters or missions that guide the tester's actions and objectives. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33; Chapter 5, page 47-48.

NEW QUESTION 58
Which of the following issues cannot be identified by static analysis tools?

A. Very low MTBF (Mean Time Between failure)
B. Potentially endless loops
C. Referencing a variable with an undefined value
D. Security vulnerabilities

Answer: A

Explanation:
Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential bugs12. However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)3. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures. MTBF can only be estimated by dynamic testing, which involves executing the program under various conditions and collecting data on its failures4. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools. The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can

be identified by static analysis tools. Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false5. Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables6. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption. References = What Is Static Analysis? Static Code Analysis Tools - Perforce Software, How Static Code Analysis Works | Perforce, Static Code Analysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs -
CareerRide, ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

NEW QUESTION 63
Which of the following BEST matches the attributes with a level of testing?

A. Stubs and drivers are often usedII The lest environment should correspond to the production environment III Finding defects is not the main focusIV Testing can be based on use casesV Testing is normally performed by testersVI Testing for functional and non-functional characteristics
B. Component - VI Integration - IV System -1 Acceptance - 111
C. Component - IV Integration -1 System - VI Acceptance - V
D. Component-I Integration - V System - II Acceptance - IV
E. Component - V Integration - II System - IV Acceptance - VI

Answer: D

Explanation:
The relationship between impact analysis and regression testing in maintenance testing is that impact analysis is used to evaluate the amount of regression testing to be performed. Maintenance testing is a type of testing that is performed on an existing software product after it has been delivered or deployed, in order to ensure that it still meets its requirements and functions correctly after a change or a modification. Maintenance testing can be triggered by various reasons, such as corrective maintenance (fixing defects), adaptive maintenance (adapting to new environments), perfective maintenance (improving performance), preventive maintenance (avoiding future problems), etc. Impact analysis is a technique that is used to assess the extent and nature of changes introduced by maintenance activities on the software product or project. Impact analysis helps to identify which parts of the software product are affected by the changes, which parts need to be modified or updated accordingly, which parts need to be retested or verified for correctness or compatibility, etc. Regression testing is a type of testing that verifies that previously tested software still performs correctly after a change or a modification. Regression testing helps to detect any side effects or unintended consequences of maintenance activities on the software product's functionality or quality. Regression testing can be performed at various levels and scopes depending on the impact analysis results. Therefore, in maintenance testing, impact analysis is used to evaluate the amount of regression testing to be performed. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 20.

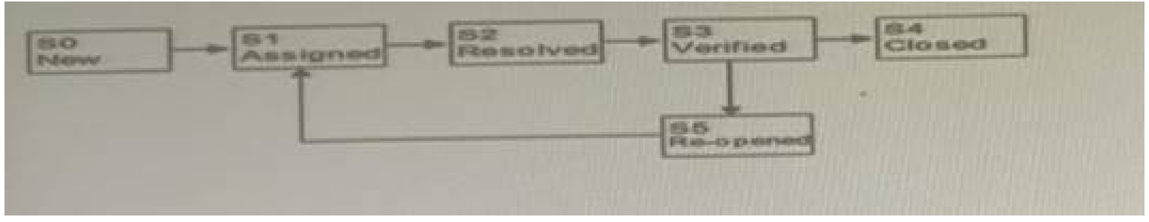
NEW QUESTION 66
Which of the following is NOT an objective of testing?

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

Answer: D

Explanation:
Analyzing and removing the cause of failures is not an objective of testing,
but rather a task of development or maintenance. A failure is an event or behavior that deviates from the expected or specified result of a system under test. A failure is caused by an error (also known as a mistake or a fault) in the software code, design, or specification. Analyzing and removing the cause of failures is a process of locating and fixing errors in the software code, design, or specification, which is also known as debugging or defect resolution. Analyzing and removing the cause of failures does not aim to find or report defects, but rather to correct or prevent them. The other options are objectives of testing. Finding defects is one of the main objectives of testing, as it helps to improve the quality and reliability of the software product. Providing information for decision-making is another objective of testing, as it helps to support decision making and risk management. Gaining confidence about the level of quality of the software is another objective of testing, as it helps to assure that the software product meets its requirements and customer or user needs and expectations. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 3.

NEW QUESTION 71
Which sequence of state transition stated in the answer choices is correct in accordance with the following figure depicting me life-cycle of a defect?



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

Answer: C

Explanation:
The figure depicts the life-cycle of a defect using state transition testing. State transition testing is a technique that models how a system transitions from one state to another depending on events or conditions. The figure shows six states (S0 to S5) and seven transitions (T0 to T6). The correct sequence of state transitions that follows the figure is S0->S1->S2->S3->S5->S1->S2->S3. This sequence represents the following scenario:
? S0: The defect is not yet detected (initial state).
? T0: The defect is detected by testing (event).
? S1: The defect is reported and registered (state).
? T1: The defect is assigned to a developer for fixing (event).

? S2: The defect is being fixed by the developer (state).
? T2: The developer fixes the defect and delivers a new version (event).
? S3: The defect is verified by testing (state).
? T5: The testing fails to confirm that the defect is fixed (event).
? S5: The defect is rejected by testing (state).
? T6: The defect is reassigned to a developer for fixing (event).
? S1: The defect is reported and registered (state).
? T1: The defect is assigned to a developer for fixing (event).
? S2: The defect is being fixed by the developer (state).
? T2: The developer fixes the defect and delivers a new version (event).
? S3: The defect is verified by testing (state). The other sequences are incorrect, as they do not follow the transitions shown in the figure. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION 75
Which of the following is the main benefit of a configuration management of testware?

A. All testware is backed up with restore option, including incident reports and change request
B. The testware can be traced to information in requirements tools and to the bug tracking system.
C. All testware items are identified, version controlled, tracked tor changes with relation to each other
D. There is an easy way to assess the level to test coverage provided by the existing tests

Answer: C

Explanation:
Configuration management of testware is a critical aspect of maintaining the integrity and traceability of test assets throughout the testing lifecycle. The main benefit of configuration management is to ensure that all testware items, such as test cases, test scripts, test data, and test results, are systematically identified, version controlled, and tracked for changes in relation to each other.
Option C accurately describes this benefit. By applying configuration management principles to testware, teams can manage changes to test assets efficiently, ensuring that the testware remains consistent, up-to-date, and aligned with the version of the software under test. This control mechanism facilitates the reproducibility of tests, enhances the reliability of testing activities, and supports traceability from requirements through to defects.
Options A, B, and D describe other aspects of test management and testing processes but do not capture the core benefit of configuration management of testware, which is centered on the systematic control and tracking of testware items.

NEW QUESTION 79
Which of the following BEST describes checklist-based testing?

A. Checklist-based testing includes formal tests from detailed lists of test conditions, allowing much repeatability
B. Checklist-based testing may involve a list of tests based on what is important to the user as well as an understanding of why and how software fails
C. Checklist-based testing, while popular, provides little consistency and few guidelines concerning test case development
D. Checklist-based testing is restricted to non-functional testing, including usability, performance, and security test

Answer: B

Explanation:
Checklist-based testing involves using checklists that contain items, such as potential test conditions, that should be tested. These checklists are often based on insights into what is important to the user, potential areas where software might fail, and specific aspects that need to be tested. It provides a structured yet flexible approach to testing, ensuring key areas are covered while allowing testers to use their experience and understanding of the system. Checklist-based testing is not limited to non-functional testing but can be applied to various types of testing, including functional testing.References:
? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4.5.

NEW QUESTION 82
Which of the following statements BEST describes how test cases are derived from a use case?

A. Test cases are derived based on non-functional requirements such as usability
B. Test cases are created using white-box test techniques to execute scenarios of use cases
C. Test cases are derived based on pair testing between a user and a tester to find defects
D. Test cases are designed to cover various user behaviors, including basic, exceptional oralternative and error behaviors associated with human users or systems

Answer: D

Explanation:
Use cases describe a system's behavior as it responds to a request from a user. They typically consist of various scenarios, such as basic flow, alternative flow, and exceptional flow, which represent possible behaviors when a user interacts with the system. When deriving test cases from use cases, it is important to cover these different types of user behaviors.Test cases should be designed to verify how the system behaves during each of these scenarios. This ensures that the system operates correctly for normal and error conditions encountered by human users or systems interacting with the application. Thus, test cases derived from use cases aim to cover basic, exceptional, and alternative flows, ensuring comprehensive coverage.References:
? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.2.4.

NEW QUESTION 87
4 equivalence classes are given for integer values:
0 < x <100
100<= x <= 200
200 < x < 500
x >= 500
Which of the following options represent correct set of data for valid equivalence class partitions?

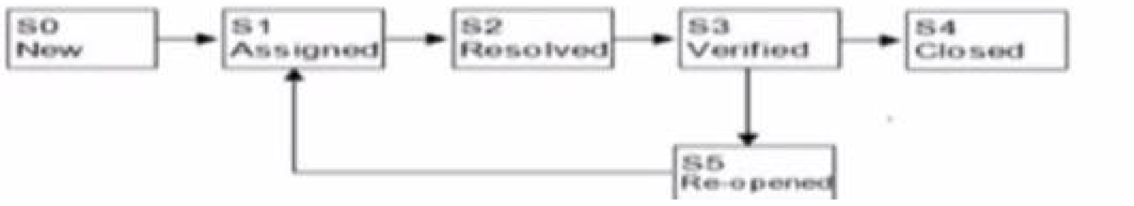
A. 50; 100; 200. 1000
B. 0. 1.99, 100.200,201.499, 500;

C. 0.50; 100; 150.200.350.500;
D. 50; 100; 250; 1000

Answer: C

Explanation:
The correct set of data for valid equivalence class partitions should include one value from each equivalence class, and no value from outside the range. Option C satisfies this condition, as it has one value from each of the four equivalence classes (50, 100, 250, 500). Option A has two values from the same equivalence class (100 and 200), option B has values outside the range (0 and 0.99), and option D has two values from the same equivalence class (1000 and 500). Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 35.

NEW QUESTION 89
Which sequence of stated in the answer choices is correct in accordance with the following figure depicting the life-cycle of a defect?



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

Answer: D

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

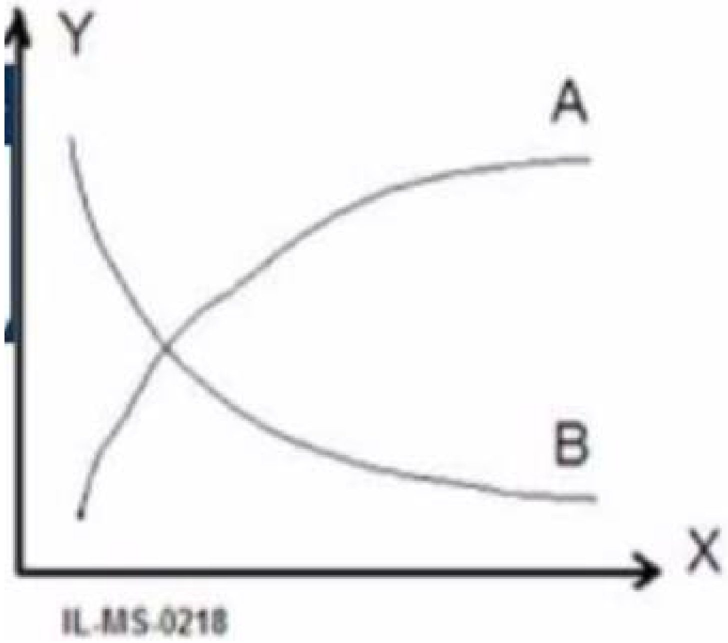
NEW QUESTION 92
The following open incident report provided: Date: 01.01.01
Description: When pressing the stop button the application status remain in "Attention" instead of "Ready'.
Severity: High
Life Cycle: Integration
Which of the following details are missing in the giving incident report?

A. Identification or configuration of the applicationI
B. The name of the developerII
C. Recommendation of the developerIV The actions and/or conditions that came before the pressing of the button
D. IV
E. IV
F. II
G. II, III

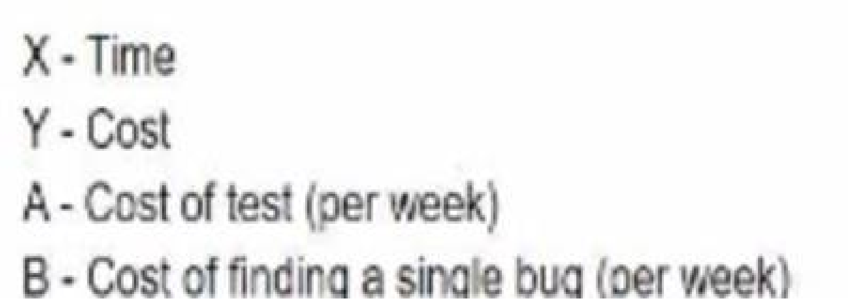
Answer: B

Explanation:
In an incident report, essential details provide context and facilitate the investigation and resolution of the incident. The missing elements in the given incident report are:
I. Identification or configuration of the application: This detail is crucial as it specifies which version or configuration of the application is affected, helping in reproducing the issue. IV. The actions and/or conditions that came before pressing the button: Understanding the sequence of actions leading to the issue is vital for replicating and diagnosing the problem. The name of the developer (II) and the recommendation of the developer (III) are not typically included in an incident report as they do not contribute to identifying or resolving the incident. The focus is on the incident's details, reproduction steps, and the system's state rather than on personnel or proposed solutions at this stage. Therefore, option B, which includes both I and IV, is the correct answer.

NEW QUESTION 97
The following chart represents metrics related to testing of a project that was competed. Indicate what is represented by tie lines A, B and the axes X.Y

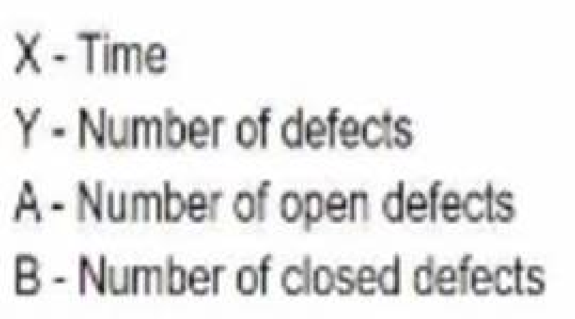


A)



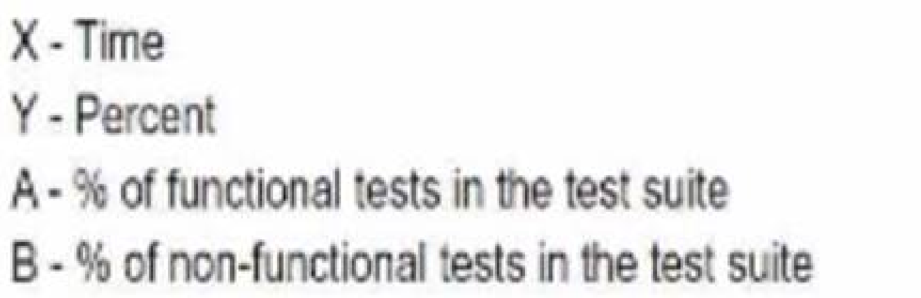
X - Time
Y - Cost
A - Cost of test (per week)
B - Cost of finding a single bug (per week)

B)



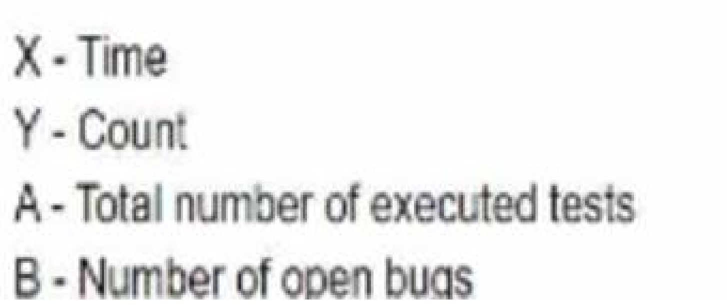
X - Time
Y - Number of defects
A - Number of open defects
B - Number of closed defects

C)



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

D)



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

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

Answer: D

Explanation:
Option D correctly explains what is represented by the lines A, B and the axes X, Y in a testing metrics chart. According to option D:
? X-axis represents Time
? Y-axis represents Count
? Line A represents Number of open bugs
? Line B represents Total number of executed tests
This information is essential in understanding and analyzing the testing metrics of a completed project.
References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 2.5.1, Page 35.

NEW QUESTION 101
The testers in company A were part of the development team. Due to an organizational change they moved to be part of the support team.
What are the advantages and the disadvantages of this change?

A. Advantage: More independence in deciding what and how to test, Disadvantage: Isolation from me development team knowledge

B. Advantage: being closer to customer perspective, Disadvantage less independence in perspectives
C. Advantage: pulled to support tasks and having less time for testing, Disadvantage less chances to move a tester to development
D. Advantage: increased chances to move a tester to development; Disadvantage: pulled to support tasks and having less time for testing

Answer: B

Explanation:
Being part of the support team means that the testers are closer to the customer perspective, which is an advantage for testing, as they can better understand the user needs and expectations, and identify more realistic scenarios and risks. However, being part of the support team also means that they have less independence in deciding what and how to test, as they may be influenced by the customer's preferences or requests, which could compromise the objectivity and effectiveness of testing. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 6.

NEW QUESTION 106
Which of the following statements is not correct?

A. Looking for defects in a system may require Ignoring system details
B. Identifying defects may be perceived as criticism against product
C. Looking for defects in system requires professional pessimism and curiosity
D. Testing is often seen as a destructive activity instead of constructive activity

Answer: A

Explanation:
? Looking for defects in a system does not require ignoring system details, but rather paying attention to them and understanding how they affect the system's quality, functionality, and usability. Ignoring system details could lead to missing important defects or testing irrelevant aspects of the system.
? Identifying defects may be perceived as criticism against product, especially by the developers or stakeholders who are invested in the product's success. However, identifying defects is not meant to be a personal attack, but rather a constructive feedback that helps to improve the product and ensure its alignment with the requirements and expectations of the users and clients.
? Looking for defects in system requires professional pessimism and curiosity, as testers need to anticipate and explore the possible ways that the system could fail, malfunction, or behave unexpectedly. Professional pessimism means being skeptical and critical of the system's quality and reliability, while curiosity means being eager and interested in finding out the root causes and consequences of the defects.
? Testing is often seen as a destructive activity instead of constructive activity, as it involves finding and reporting the flaws and weaknesses of the system, rather than creating or enhancing it. However, testing is actually a constructive activity, as it contributes to the system's improvement, verification, validation, and optimization, and ultimately to the delivery of a high-quality product that meets the needs and expectations of the users and clients.

NEW QUESTION 108
Which statement about use case testing is true?

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

Answer: D

Explanation:
Use case testing is a technique that helps identify test cases that exercise the whole system on a transaction by transaction basis from start to finish. Use cases are descriptions of how users interact with the system to achieve a specific goal. Use case testing is not focused on data flow, but rather on process flow. Use case testing can be performed by professional testers, customers or end users, depending on the context. Use case testing does not require the test cases to be designed by customers or end users, but rather by anyone who has access to the use case specifications. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 4, page 36.

NEW QUESTION 113
Which of the following statements is the BEST example of non-functional testing?

A. Tests which capture the time it takes to save a file
B. Tests which calculate overtime pay for those employees entitled to such
C. Tests related to "what" the system should do
D. Tests based on the internal structure of a component or system

Answer: A

Explanation:
Non-functional testing refers to testing aspects that do not relate to specific behaviors or functions of the software but to attributes such as performance, usability, reliability, etc. Tests that capture the time it takes to save a file directly relate to the performance of the system, thus falling under non-functional testing.References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.2.5 "Functional and Non- functional Testing".

NEW QUESTION 115
Which of the following statements about static analysis are FALSE?

A. Static analysis can be used Instead of dynamic testing.I
B. Stalk: analysis can uncover defects like security vulnerabilities.II
C. Static analysis can be used to check conformance to specifications and standard
D. IV Static analysis typically detects failures prior to component testing.
E. II
F. I
G. Ill
H. II
I. IV

J. I, IV

Answer: D

Explanation:
Static analysis involves analyzing the software's code, design, and structure without executing the program. It can uncover various types of defects, including security vulnerabilities (II) and non-conformance to specifications and standards (III). However, static analysis cannot replace dynamic testing (I), which involves executing the software to observe its behavior under various conditions. Dynamic testing can identify failures that static analysis cannot, such as those related to runtime issues and interaction between different parts of the software. Statement IV is false because static analysis does not detect failures; it detects defects. Failures are observed when the software is executed, which is beyond the scope of static analysis.

NEW QUESTION 120
A software application incorrectly provided customers discounts of 50% off their total purchases if the purchases exceeded S100. It was discovered through an audit that the discount should have been only 5% off these purchases. A root cause analysis uncovered that the requirements Incorrectly stated 50% instead of 5% in this scenario.
Which of the following MOST accurately reflects this scenario?

A. The audit finding Is the root cause, the incorrect calculation of 50% Is the defect, and the Incorrect requirement Is the effect
B. The incorrect customer discount is the effect and the reason for the requirement error is the root cause
C. The incorrect discount is the root cause, requiring a root cause analysis which led to investigating the software code, design, and requirements
D. A defect in the code is determined to be the root cause of the incorrect calculation

Answer: B

Explanation:
According to the ISTQB CTFL syllabus, a defect is a deviation from the expected result which in this scenario is the incorrect discount applied to the customers. The root cause, as per the ISTQB definition, is the originating cause of a defect, which in this case is the incorrect requirement stating 50% instead of 5%. Therefore, the incorrect requirement is the root cause and the customer receiving the wrong discount is the effect of this root cause.References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.3 "Defects, Root Causes, and Effects".

NEW QUESTION 121
The following sentences refer to the 'Standard for Software Test Documentation' specification (IEEE 829).
Which sentence is correct?

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

Answer: B

Explanation:
The IEEE 829 standard is a widely used specification for test documentation, but it is not mandatory or universal. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization. The standard does not require any approval from management, marketing or development for any deviation, nor does it depend on the type of project (military or consumer market). The standard also does not guarantee high quality test documentation regimes, as it only provides a general outline and format, not the actual content or quality criteria. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 16.

NEW QUESTION 123
Which of the following is NOT a deciding factor m determining the extent of testing required?

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

Answer: C

Explanation:
The extent of testing required for a software product depends on various factors, such as the level of risk, the budget, and the time available. The level of risk reflects the potential impact of failures on the stakeholders and the environment. The budget determines how much resources can be allocated for testing. The time available defines the schedule and deadlines for testing activities. The particular tester involved in testing is not a deciding factor for the extent of testing required, as testing should be based on objective criteria and not on personal preferences or abilities. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 14-15.

NEW QUESTION 127
Which of the following exemplifies how a software bug can cause harm to a company?

A. "Print" prints the last page twice for a file with 1000 pages
B. The timeout on the login page of a web site is 9 minutes, while the requirement was for10 minutes
C. When uninstalling the application, the uninstall dialog has a spelling mistake
D. When calculating the final price in a shopping list, the price of the last item is not added

Answer: D

Explanation:
A software bug can cause harm to a company by directly affecting its operations, reputation, user satisfaction, and financials. Option D, "When calculating the final

price in a shopping list, the price of the last item is not added," describes a defect that directly impacts the core functionality of a financial transaction, potentially leading to financial loss and customer dissatisfaction. This can have severe implications for the company's credibility and revenue. Options A, B, and C describe bugs that, while potentially annoying, do not have the same direct impact on the company's core operations and financial integrity as option D.

NEW QUESTION 132
Decision table testing is being performed on tran-sactions in a bank's ATM (Automated Teller Machine) system. Two test cases have already been generated for rules 1 and 4. which are shown below:
SEE ATTACHMENT 1
Given the following additional test cases: SEE ATTACHMENT 2
Which two of the additional test cases would achieve full coverage of the full decision table (when combined with the test cases that have already been generated for rules 1 and 4)?

A. DT1, DT4
B. DT3, DT4
C. DT2, DT3
D. DT1.DT2

Answer: C

Explanation:
Decision table testing is used to analyze combinations of inputs to determine the appropriate outputs, often based on specific rules or conditions.
For the problem statement:
? Rule 1: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = True, Correct PIN = True)
? Rule 4: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = True, Correct
PIN = False)
The additional test cases are:
? DT1: (Withdrawal = Allowed, Balance = Insufficient, Fast Cash = True, Correct PIN
= True)
? DT2: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = False, Correct PIN
= True)
? DT3: (Withdrawal = Allowed, Balance = Insufficient, Fast Cash = True, Correct PIN
= False)
? DT4: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = False, Correct PIN
= False)
From the given test cases, DT2 covers the scenario where Fast Cash is False, which is not covered in the initial cases. DT3 covers the case where Balance is Insufficient and PIN is incorrect.
Combining Rules 1 and 4 with DT2 and DT3 covers all the scenarios. References:
? Certified Tester Foundation Level v4.0
? 10 Sample Exams ISTQB Foundation Level (CTFL) v4.0

NEW QUESTION 133
Consider the following statements about risk-based testing.
I) Risk-based testing has the objective to reduce the level of protect risks.
II) Tests should be prioritized to find tie critical detects as early as possible.
III) Non-testing activities may also help to reduce risk
IV) Risks have to be reassessed on a regular basis.
V) The project stakeholders can give useful input to determine the risks

A. I III IV and V are tru
B. II is false.
C. II, III IV and V are correc
D. I is false.
E. I, II and IV are tru
F. III and V are false.
G. II, Ill and V are tru
H. 1 ants Iv are false.

Answer: B

Explanation:
The following statements about risk-based testing are correct:
? II) Tests should be prioritized to find tie critical detects as early as possible. Risk- based testing involves prioritizing tests based on risk level, which reflects both the likelihood and impact of defects or failures. Tests with higher risk level should be executed earlier than tests with lower risk level, in order to find and fix critical defects as soon as possible.
? III) Non-testing activities may also help to reduce risk. Risk-based testing does not
only involve testing activities, but also other activities that can help mitigate risks, such as reviews, inspections, audits, simulations or prototyping.

NEW QUESTION 134
Which of the following is MOST likely to be an example of a PROJECT risk?

A. A computation is not always performed correctly in some situations
B. A system architecture may not support some non-functional requirements
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

Answer: C

Explanation:
A project risk relates to potential issues that could affect the overall success of the project. Among the options provided, the risk that "Team members' skills may not be sufficient for the assigned work" is clearly a project risk because it pertains to the potential failure of the project due to inadequate skillsets among the team.

This risk affects the entire project's ability to meet its objectives.References:
? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.2.

NEW QUESTION 137
Which of the following is an INCORRECT statement about the benefit of traceability between the test basis and test work products?

A. Traceability may be required by IT governance rules
B. Traceability may help evaluate the extent of test coverage
C. Traceability may allow testing to be auditable
D. Traceability may make it harder to understand the impact of changes

Answer: D

Explanation:
The statement "Traceability may make it harder to understand the impact of changes" is incorrect. Traceability in testing actually facilitates understanding the impact of changes by linking test cases to requirements. This linkage helps ensure that any changes in the requirements are adequately reflected and verified in the test cases, thus supporting effective management of changes and maintaining the integrity of the system or product being developed (ISTQB not-for-profit association) (ISTQB Main Web).References:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf
? ISTQB Official Website - CTFL Certification: https://www.istqb.org/certifications/certified-tester-foundation-level/

NEW QUESTION 139
Which or the following is a valid collection of equivalence classes for the following problem: An integer field shall contain values from and including 1 to and including 15.

A. Less than 0.1 through 14. 15 and more
B. Less than 1.1 through 14. more than 15
C. negative number
D. 1 through 15. above 15
E. Less than 1.1 through 15. more than 15

Answer: D

Explanation:
Equivalence partitioning is a black-box test design technique where inputs to the software or system are divided into groups that are expected to exhibit similar behavior. For an integer field that should accept values from 1 to 15, the valid equivalence class is 1 through 15. The invalid equivalence classes are numbers less than 1 and numbers more than 15. Therefore, option D, "Less than 1, 1 through 15, more than 15," correctly identifies the valid equivalence class along with the two invalid classes, covering all possible input scenarios for the field. Options A, B, and C either do not accurately capture the valid range or incorrectly specify the range boundaries.

NEW QUESTION 141
In which of the following cases you would NOT execute maintenance testing?

A. Retirement of the software or system
B. Modifications to a released software or system
C. Migration of the system data to a replacement system
D. Update to the Maintainability requirements during the development phase

Answer: D

Explanation:
Maintenance testing is testing performed on a software product after delivery to correct defects or improve performance or other attributes. Maintenance testing can be triggered by various situations, such as modifications to a released software or system, migration of the system data to a replacement system, or retirement of the software or system. Maintenance testing is not executed when there is an update to the maintainability requirements during the development phase, as this is not a maintenance situation but rather a change request that should be handled by the development process. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 18-19.

NEW QUESTION 142
Which statement best describes the key difference between a mindset for test activities and a mindset for development activities?

A. A tester possesses professional pessimism while a developer is concerned with validating the product
B. A tester is concerned with finding defects while a developer is interested in designing solutions
C. A tester is interested in building solutions while a developer is concerned with verifying the product
D. A tester is concerned with verifying the product while a developer possesses professional pessimism

Answer: B

Explanation:
The key difference between the mindsets for test activities and development activities lies in the objectives: a tester is primarily concerned with finding defects to ensure product quality, while a developer focuses on designing and building solutions. This distinct focus helps ensure thorough quality checks and balances within the software development life cycle (ISTQB not-for-profit association).References:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web- prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf

NEW QUESTION 145
For the following pseudo-code determine number of tests required for 100% statement coverage
IF Gender = Boy
If Age > 3 AND Age < 5 Shoe Size = 1
ELSE IF Age >=5 AND Age < 7

```
Shoe Size = 2 ENDIF
ELSE
IF Age > 3 AND Age < 5
Shoe Size = 0
ELSE IF Age >=5 AND Age < 7
Shoe Size = 1 ENDIF ENDIF
```

A. 6
B. 4
C. 2
D. 6

Answer: B

Explanation:
To achieve 100% statement coverage, we need to design test cases that ensure every statement in the given pseudo-code is executed at least once. Analyzing the pseudo-code, we notice that there are conditions based on two variables: Gender and Age. To cover all statements, we need to consider the paths that lead to each assignment of the Shoe Size variable.
? Gender = Boy, Age <= 3 (Shoe Size assignment is not reached, but the condition is evaluated)
? Gender = Boy, Age > 3 AND Age < 5 (Shoe Size = 1)
? Gender = Boy, Age >= 5 AND Age < 7 (Shoe Size = 2)
? Gender != Boy, Age <= 3 (Again, Shoe Size assignment is not reached, but the condition is evaluated)
? Gender != Boy, Age > 3 AND Age < 5 (Shoe Size = 0)
? Gender != Boy, Age >= 5 AND Age < 7 (Shoe Size = 1)
However, upon closer inspection, we see that tests 1 and 4 do not contribute to statement coverage as they do not lead to a Shoe Size assignment. Therefore, we only need 4 test cases to achieve 100% statement coverage, making option B the correct answer.

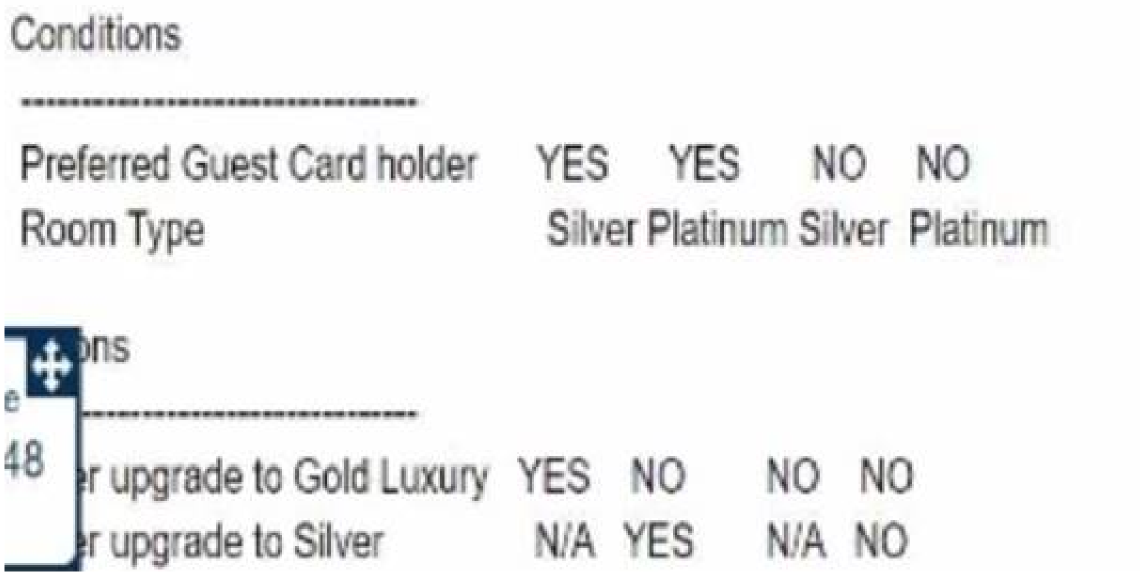
NEW QUESTION 147
Which of the following statements about decision tables are TRUE?

A. Generally, decision tables are generated for low risk test items.I
B. Test cases derived from decision tables can be used for component tests.II
C. Several test cases can be selected for each column of the decision table.I
D. The conditions in the decision table represent negative tests generally.
E. Ill
F. I, IV
G. I
H. IV
I. I
J. Ill

Answer: D

Explanation:
A decision table is a technique that shows combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects). A decision table consists of four quadrants: conditions (inputs), actions (outputs), condition entries (values) and action entries (results). The following statements about decision tables are true:
? II. Test cases derived from decision tables can be used for component tests.
Decision tables can be used to test components that have multiple inputs and outputs that depend on logical combinations of conditions. Decision tables can help cover all possible combinations or scenarios in a systematic way.
? III. Several test cases can be selected for each column of the decision table. A column of a decision table represents a unique combination of condition entries and action entries. Several test cases can be selected for each column by varying other input values or expected results that are not part of the decision table. The following statements about decision tables are false:
? I. Generally, decision tables are generated for low risk test items. Decision tables are not related to risk level, but rather to complexity level. Decision tables are generated for test items that have complex logic or multiple conditions and actions that need to be tested.
? IV. The conditions in the decision table represent negative tests generally. The conditions in the decision table represent both positive and negative tests, depending on whether they are valid or invalid inputs for the test item. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 4, page 42-43.

NEW QUESTION 150
You are testing a room upgrade system for a hotel. The system accepts three differed types of room (increasing order of luxury): Platinum. Silver and Gold Luxury. ONLY a Preferred Guest Card holder s eligible for an upgrade.
Below you can find the decision table defining the upgrade eligibility:



Conditions				
Preferred Guest Card holder	YES	YES	NO	NO
Room Type	Silver	Platinum	Silver	Platinum
ons				
r upgrade to Gold Luxury	YES	NO	NO	NO
r upgrade to Silver	N/A	YES	N/A	NO

What is the expected result for each of the following test cases?

Customer A: Preference Guest Card holder, holding a Silver room Customer B: Non Preferred Guest Card holder, holding a Platinum room

A. Customer A; doesn't offer any upgrade; Customer B: offers upgrade to Gold luxury room
B. Customer A: doesn't offer any upgrade; Customer B: doesn't offer any upgrade.
C. Customer A: offers upgrade to Gold Luxury room; Customer B: doesn't offer any upgrade
D. Customer A: offers upgrade to Silver room; Customer B: offers upgrade to Silver room.

Answer: C

Explanation:
According to the decision table in the image, a Preferred Guest Card holder with a Silver room is eligible for an upgrade to Gold Luxury (YES), while a non-Preferred Guest Card holder, regardless of room type, is not eligible for any upgrade (NO). Therefore, Customer A (a Preferred Guest Card holder with a Silver room) would be offered an upgrade to Gold Luxury, and Customer B (a non-Preferred Guest Card holder with a Platinum room) would not be offered any upgrade.
References = The answer is derived directly from the decision table provided in the image; specific ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents are not referenced.

NEW QUESTION 155
Can "cost" be regarded as Exit criteria?

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

Answer: A

Explanation:
Cost can be regarded as an exit criterion for testing, because it is a factor that affects the profitability and feasibility of the software product. Testing is an investment that aims to improve the quality and reliability of the software product, but it also consumes resources, such as time, money, and human effort. Therefore, testing should be planned and executed in a way that balances the cost and benefit of testing activities. Having cost as an exit criterion helps to avoid spending too much money on testing, which may result in an unprofitable product or a loss of competitive advantage. Cost can also help to prioritize and focus the testing efforts on the most critical and valuable features and functions of the software product. However, cost should not be the only exit criterion for testing, as it may not reflect the true quality and risk level of the software product. Other exit criteria, such as defect rate, test coverage, user satisfaction, etc., should also be considered and defined in the test plan.
The other options are incorrect, because they either deny the importance of cost as an exit criterion, or they make false or unrealistic assumptions about the cost of testing. Option B is incorrect, because the financial value of product quality can be estimated, for example, by using cost-benefit analysis, return on investment, or cost of quality models. Option C is incorrect, because going by cost as an exit criterion does not necessarily constrain the testing project or help achieve the desired quality level. Cost is a relative and variable factor that depends on the scope, complexity, and context of the software product and the testing project. Option D is incorrect, because the cost of testing can be measured effectively, for example, by using metrics, such as test effort, test resources, test tools, test environment, etc.

NEW QUESTION 156
The following part of a business process flow is specified; REPEAT (book a bill) UNTIL (User presses Cancel). How many test cases are necessary in order to achieve 100% branch coverage of the process flow?

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

Answer: C

Explanation:
To achieve 100% branch coverage of the process flow, we need to test both the true and false outcomes of the condition (User presses Cancel). Branch coverage is a type of structural testing that measures how many decision outcomes in a program have been executed by a test suite. Branch coverage can be used to assess the adequacy or completeness of a test suite.
To test the true outcome of the condition, we need a test case that simulates the user pressing Cancel after booking a bill. This test case will exit the loop and end the process flow.
To test the false outcome of the condition, we need a test case that simulates the user not pressing Cancel after booking a bill. This test case will repeat the loop and book another bill.
Therefore, we need at least two test cases to achieve 100% branch coverage of the process flow. One test case for each possible outcome of the condition.
Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION 159
Which of the following is NOT a common objective of testing?

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

Answer: C

Explanation:
Debugging the software to find the reason for defects is not a common objective of testing, but rather a task of development or maintenance. Debugging is a process of locating and fixing errors in the software code, while testing is a process of finding and reporting defects in the software behavior or quality. Testing does not aim to fix defects, but rather to provide information on their existence and impact. The other options are common objectives of testing. Finding defects in

the software is one of the main objectives of testing, as it helps to improve the quality and reliability of the software. Preventing defects is another objective of testing, as it helps to avoid rework and reduce costs and risks. Providing information on the status of the system is another objective of testing, as it helps to support decision making and risk management. Verified
References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 3.

NEW QUESTION 163
Your manager asked you when testing will be complete. In order to answer this question, you'll most likely use:

A. Test progress reports
B. Your colleagues advice
C. A conversion spreadsheet
D. A Test Oracle

Answer: A

Explanation:
When a manager asks when testing will be complete, the most appropriate and informative resource to provide an answer is test progress reports (Option A). Test progress reports contain detailed information on the status of testing activities, including what has been accomplished, what remains to be done, the results of the tests conducted, and any issues or risks that might impact the completion of testing. These reports allow for an informed assessment of the testing progress and estimation of when testing might be completed. Options B, C, and D do not provide the structured, detailed, and specific information required to accurately answer the manager's question about the completion of testing.

NEW QUESTION 166
Which of the following would be the LEAST likely to be used as the basis for a test exit criteria?

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

Answer: A

Explanation:
Test exit criteria are the conditions or requirements that must be met before testing can be concluded. Test exit criteria are usually defined in the test plan and agreed by the stakeholders. Test exit criteria can be based on various factors, such as test coverage, defect status, quality level, risk level, etc. Test schedules would be the least likely to be used as the basis for test exit criteria, because test schedules are not directly related to the quality or performance of the software product, but rather to the time or resources allocated for testing. Test schedules can be used as the basis for test entry criteria, which are the conditions or requirements that must be met before testing can start. The other options are more likely to be used as the basis for test exit criteria. Cost of testing performed so far can be used as a basis for test exit criteria, because it can indicate the return on investment or the cost-benefit ratio of testing. Confidence of testers in tested code can be used as a basis for test exit criteria, because it can reflect the level of satisfaction or assurance of the testers about the quality or reliability of the software product. Number of unfixed defects can be used as a basis for test exit criteria, because it can indicate the level of risk or impact of the remaining defects on the software product. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 13.

NEW QUESTION 168
Which of the following is true about Oracles?

A. Sometimes old version of a product can be used as an Oracle
B. Oracles help in reproducing the irreproducible bugs
C. Oracles are derived from the design
D. Oracles can be generated automatically using data generators

Answer: A

Explanation:
An oracle is a mechanism or source that can provide the expected result for a given test input or situation. Sometimes old version of a product can be used as an oracle, if it is assumed that the old version behaves correctly for the test cases that are executed on the new version. This is also known as back-to-back testing. Oracles do not help in reproducing the irreproducible bugs, as they only provide the expected results, not the actual results. Oracles are not derived from the design, but from the requirements or specifications. Oracles cannot be generated automatically using data generators, as data generators only provide test inputs, not test outputs. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 9.

NEW QUESTION 171
Why it is essential that defects found in a review be reported objectively?

A. In order to facilitate easy entry of detected defects in a OTS (Defect Tracking System)
B. In order to allow the author of reviewed work product(S) to take the feedback positively as an effort at improving the product (S) and not as a personal assault
C. In order to allow the review moderator to easily understand them, and assign them to the right developer for fixing
D. In order to allow augmentation of existing checklists used for reviewing the work product (S)

Answer: B

Explanation:
The purpose of a review is to find defects and improve the quality of the work product, not to criticize or blame the author. Reporting defects objectively means describing them factually and constructively, without using negative or emotional language that could offend the author or damage their motivation. This way, the author can take the feedback positively as an effort at improving the product and not as a personal assault. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 138.

NEW QUESTION 175
Which of the following is correct with regards to debugging?

A. Debugging identifies the cause of a failure
B. Debugging is often performed by test engineers
C. Debugging is considered part of the testing activities
D. Debugging is intended to find as many defects as possible in the code

Answer: A

Explanation:
Debugging is the process of finding, analyzing and removing the causes of failures in software. Debugging is not considered part of testing, but rather a development activity that can involve testing. Debugging is not intended to find as many defects as possible, but rather to fix the specific failure that was observed. Debugging is usually performed by developers, not by test engineers. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 1, page 6.

NEW QUESTION 179
A software calculates the annual car tax using three inputs:
- E; the emission level of the vehicle
- P: the power of the vehicle
-T the type of the vehicle
The input value for P can be integer positive values between 15 and 350.
Which of the following answers contains a correct list of a boundary values for the P input?

A. 14,351
B. 14,15,350,351
C. 15,350
D. 5.175.500

Answer: B

Explanation:
A correct list of boundary values for the P input should include the minimum and maximum values of the valid range (15 and 350), as well as the values just below and above the boundaries (14 and 351). Boundary value analysis is a test design technique that involves testing the values at or near the boundaries of an input domain or output range, as these values are more likely to cause errors than values in the middle. Option B satisfies this condition, as it has all four boundary values (14, 15, 350, 351). Option A has only two boundary values (14 and 351), option C has only two boundary values (15 and 350), and option D has no boundary values at all. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 34.

NEW QUESTION 181
During component testing of a program if 100% decision coverage is achieved, which of the following coverage criteria is also guaranteed to be 100%?

A. 100% Stale transition coverage
B. 100% Equivalence class coverage
C. 100% Boundary value coverage
D. 100% Statement coverage

Answer: D

Explanation:
Statement coverage is a structural coverage metric that measures the percentage of executable statements in the source code that are executed by a test suite1. Decision coverage is another structural coverage metric that measures the percentage of decision outcomes (such as branches or conditions) in the source code that are executed by a test suite1. Decision coverage is a stronger metric than statement coverage, because it requires that every possible outcome of each decision is tested, while statement coverage only requires that every statement is executed at least once2. Therefore, if a test suite achieves 100% decision coverage, it also implies that it achieves 100% statement coverage, because every statement in every branch or condition must have been executed. However, the converse is not true: 100% statement coverage does not guarantee 100% decision coverage, because some branches or conditions may have multiple outcomes that are not tested by the test suite2. For example, consider the following pseudocode:
if (x > 0) then print("Positive") else print("Non-positive") end if
A test suite that executes this code with x = 1 and x = -1 will achieve 100% statement coverage, because both print statements are executed. However, it will not achieve 100% decision coverage, because the condition x > 0 has only been tested with two outcomes: true and false. The third possible outcome, x = 0, has not been tested by the test suite. Therefore, the test suite may miss a potential bug or error in the condition or the branch.
The other options, such as stale transition coverage, equivalence class coverage, and boundary value coverage, are not guaranteed to be 100% by achieving 100% decision coverage. Stale transition coverage is a structural coverage metric that measures the percentage of transitions between states in a state machine that are executed by a test suite3. Equivalence class coverage is a functional coverage metric that measures the percentage of equivalence classes (or partitions) of input or output values that are tested by a test suite4. Boundary value coverage is another functional coverage metric that measures the percentage of boundary values (or extreme values) of input or output ranges that are tested by a test suite4. These metrics are independent of decision coverage, because they are based on different aspects of the system under test, such as its behavior, functionality, or specification. Therefore, achieving 100% decision coverage does not imply achieving 100% of any of these metrics, and vice versa. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Test Coverage in Software Testing -Guru99, Structural Coverage Metrics - MATLAB & Simulink - MathWorks India, Test Design Coverage in Software Testing - GeeksforGeeks.

NEW QUESTION 185
In what way do Configuration Management effects testing?

A. Without proper configuration management, test planning cannot proceed.
B. Proper configuration management ensures that testers can uniquely identity the tested item
C. Configuration management is important for developers, not for testers
D. There is very little influence of configuration management practices on the test project.

Answer: B

Explanation:
Configuration management is a process that establishes and maintains consistency among work products throughout their life cycle. Configuration management affects testing in various ways, such as:
? Proper configuration management ensures that testers can uniquely identify the tested item, which can help traceability, reproducibility and accountability.

? Proper configuration management ensures that testers have access to consistent versions of software components and testware, which can help reliability, compatibility and efficiency.
? Proper configuration management ensures that testers can track changes and defects in software components and testware, which can help verification, validation and reporting.
? Proper configuration management ensures that testers can control the configuration of the test environment, which can help stability, security and performance.
Configuration management is not a prerequisite for test planning, as test planning can proceed without configuration management, although it may be less effective or accurate. Configuration management is not important for developers only, but for testers as well, as it affects the quality and consistency of the testing process and products. Configuration management has a significant influence on the test project, as it affects various aspects of testing, such as traceability, reproducibility, reliability, compatibility, efficiency, verification, validation, reporting, stability, security and performance. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 6, page 60-61.

NEW QUESTION 189
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

Answer: D

Explanation:
Component integration tests are designed to verify the interactions and interfaces between integrated components. These tests should be carried out after component testing (where individual components are tested in isolation) but before system testing (where the entire system is tested as a whole). This ensures that any issues arising from the integration of components are identified and resolved early in the testing process, making option D the correct answer.

NEW QUESTION 192
Which of the following is a CORRECT statement about how a tester should communicate about defects, test results, and other test information?

A. Testers should include personal opinions and judgements in defect reports and review findings
B. Testers should emphasize the benefits of testing, such as increased quality and reduced risk
C. Testers should reject all questions about their test findings and information
D. Testers should take a command-and-control approach with the project team

Answer: B

Explanation:
Communication from testers about defects, test results, and other test information should emphasize the benefits of testing such as increased quality and reduced risk. This positive framing helps in reinforcing the value of testing and ensuring stakeholders understand the contribution of testing to the overall project success (ISTQB not-for-profit association).References:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf

NEW QUESTION 193
Which of the following is an example of the absence-of-errors fallacy?

A. Repeating the same test cases will continue to find new defects, even after dozens of executions
B. A small number of modules contains the most defects
C. Since testing found very few defects, the system certainly will be successful
D. Other than trivial cases, it is not feasible to test all combinations of inputs and preconditions

Answer: C

Explanation:
The absence-of-errors fallacy, as explained in the ISTQB syllabus, is the erroneous belief that having fewer defects found in testing equates to a system being more successful upon release. This misconception can lead stakeholders to undervalue thorough testing. Answer C illustrates this fallacy perfectly: assuming that because few defects were found, the system will be successful, neglects the many other factors that contribute to system success, including user satisfaction and fit-for-purpose.References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.1.5 "Absence-of-errors fallacy".

NEW QUESTION 195
Which of the following does MOT describe a reason why testing is necessary?

A. The customer decided that 100% branch coverage shall be achieved
B. The acquisition of test automation tools was based on the assumption that it will be used m all projects
C. For avionics and pharmaceutical systems software testing is mandated by standards
D. The risks associated with delivering the system are far higher than the cost of testing

Answer: B

Explanation:
Testing is necessary for various reasons, such as:
? To detect defects and failures that may affect the quality, performance, reliability or security of a software product or system
? To verify that a software product or system meets its specified requirements, expectations and standards
? To validate that a software product or system fulfills its intended purpose and satisfies its stakeholders' needs
? To provide information and feedback about the status and risks of a software product or system
? To comply with regulations or contractual obligations that mandate testing for certain types of software products or systems The following statements describe some reasons why testing is necessary:
? A) The customer decided that 100% branch coverage shall be achieved. This is a reason why testing is necessary, as it reflects a contractual obligation or a quality standard that requires testing to measure and achieve a certain level of code coverage.

? C) For avionics and pharmaceutical systems software testing is mandated by standards. This is a reason why testing is necessary, as it reflects a regulation or a compliance requirement that mandates testing for certain types of software products or systems that have high safety or security risks.
? D) The risks associated with delivering the system are far higher than the cost of testing. This is a reason why testing is necessary, as it reflects a risk-based approach that considers testing as an investment to reduce the probability and impact of potential failures or defects. The following statement does not describe a reason why testing is necessary:
? B) The acquisition of test automation tools was based on the assumption that it will be used m all projects. This is not a reason why testing is necessary, as it reflects a business decision or a resource allocation that does not justify the need or purpose of testing. Test automation tools are not always suitable or beneficial for all projects, and testing can be performed with or without test automation tools. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 5-6.

NEW QUESTION 199
Which ONE of the following statements about acceptance testing is NOT correct?

A. Testing of disaster recovery and backup/restore is usually NOT part of acceptance testing.
B. The customers or system users are often responsible for the acceptance testing.
C. The main goal of acceptance testing is to build confidence in the system, not find defects.
D. Acceptance testing is the last level of testing performed prior to system release.

Answer: A

Explanation:
Acceptance testing is a level of testing performed to verify that a software product meets the agreed acceptance criteria and is acceptable for delivery. Acceptance testing is often performed by the customers or system users, who are the main stakeholders of the software product. The main goal of acceptance testing is to build confidence in the system, not find defects, as defects should have been detected and fixed in earlier levels of testing. Acceptance testing is the last level of testing performed prior to system release, unless there are any changes or fixes that require re-testing. Testing of disaster recovery and backup/restore is usually part of acceptance testing, as these are important aspects of system reliability and security that affect the customer satisfaction and trust. Therefore, statement A is not correct, while statements B, C and D are correct. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 2, page 20-21.

NEW QUESTION 201
Which of the following s the most correct statement about state testing techniques?

A. Static techniques can be used before all code is ready for execution
B. Static techniques find more detects then dynamic techniques.
C. Static techniques can be used by inexperienced users.
D. Static techniques are always cheaper than dynamic techniques.

Answer: A

Explanation:
State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques require the system to be executed with test cases, whereas static testing techniques do not. Static testing techniques can be applied before the code is ready for execution, such as reviews, inspections, walkthroughs, and static analysis. Static testing techniques can help find defects early in the development process, improve the quality of the code, and reduce the cost and effort of dynamic testing. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 4, Section 4.2.1, Page 281; ISTQB Glossary of Testing Terms v4.0, Page 292

NEW QUESTION 203
A bank offers a savings account with various interest rates based on the current balance in the account. The balance ranges and respective interest rates are:
Up to $100.00 =2%
$100.01 to $500.00 = 4%
$500.01 to $1,000.00 = 5% Above $1,000.00 = 7%
Using two-point boundary value analysis, which of the following sets of test inputs provides the relatively highest level of boundary coverage?

A. $5.00.$100.00,$499.99,$1,000.00,$1,000.01
B. $100.00. $100.01,$100.02,$500.00,$999.99
C. $100.00, $500.00,$1,000.00,$1,000.01
D. $5.00.$100.00,$500.00,$1,000.01

Answer: B

Explanation:
Boundary Value Analysis (BVA) is a software testing technique in which tests are designed to include values at the boundaries. The concept is to focus on the boundaries since errors tend to occur at the edges of input ranges rather than in the middle.
Given the problem statement:
? Up to $100.00 = 2%
? $100.01 to $500.00 = 4%
? $500.01 to $1,000.00 = 5%
? Above $1,000.00 = 7%
Two-point boundary value analysis means testing the two boundaries of each range. For each range:
? The boundaries for "Up to $100.00" would be $100.00 and $100.01.
? The boundaries for "$100.01 to $500.00" would be $100.00 and $500.00.
? The boundaries for "$500.01 to $1,000.00" would be $500.00 and $1,000.00.
? The boundaries for "Above $1,000.00" would be $1,000.00 and $1,000.01. Now, let's examine the options:
? A. $5.00, $100.00, $499.99, $1,000.00, $1,000.01
? B. $100.00, $100.01, $100.02, $500.00, $999.99
? C. $100.00, $500.00, $1,000.00, $1,000.01
? D. $5.00, $100.00, $500.00, $1,000.01
Given the options, B provides the highest boundary coverage (ISTQB not-for-profit association) (Udemy).
References:
? Certified Tester Foundation Level v4.0

NEW QUESTION 207
Which of the following statements is CORRECT?

A. Test cases are made up of input values, expected results and actual results developed to cover test objectives
B. Test cases describe items or events to test that are derived from the test basis during the test analysis activity
C. Test cases are sequences of actions for test execution specified during the test implementation activity
D. Test cases are derived during the test design activity to cover test objectives or test conditions

Answer: C

Explanation:
A test case is a set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition. A test case is a sequence of actions for test execution that can be followed by a tester or a test automation tool. A test case is specified during the test implementation activity, which is the activity that prepares the testware needed for test execution. A test case does not include actual results, as these are obtained during test execution and compared with the expected results. A test case does not describe items or events to test, as these are derived from the test basis during the test analysis activity. A test case is not derived during the test design activity, as this is the activity that specifies the test conditions or objectives that need to be tested. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 3, page 23-24; Chapter 4, page 34.

NEW QUESTION 208
Which of the following are valid testing principles?
I) Exhaustive testing is in general impossible.
II) Exhaustive testing should be executed for code intended to be reused.
III) Testing may guarantee that a program is correct.
IV) Testing cannot guarantee that a program is correct.
V) Defects cluster together in certain areas of the product.

A. I, IV, V
B. II, Iv
C. I, V
D. I, III

Answer: A

Explanation:
Statements I, IV and V are valid testing principles according to the ISTQB syllabus. Statement I states that exhaustive testing is in general impossible, because it would require testing all possible inputs, outputs and combinations of states, which is usually impractical or impossible. Statement IV states that testing cannot guarantee that a program is correct, because testing can only show the presence of defects, not their absence. Statement V states that defects cluster together in certain areas of the product, which means that some modules or functions are more likely to contain defects than others. Statements II and III are invalid testing principles. Statement II states that exhaustive testing should be executed for code intended to be reused, which contradicts statement I. Statement III states that testing may guarantee that a program is correct, which contradicts statement IV. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, pages 4-5.

NEW QUESTION 212
Software was found lo take much more time than the stated requirement of less than one
second to save a file. Upon investigation it was found that there was an unnecessary check inside a loop which was slowing down the file-save operation. The software not being able to meet the desired response time is an example of

A. It is not a defect
B. Defect
C. Error
D. Failure

Answer: D

Explanation:
A failure is an event in which a component or system does not perform a required function within specified limits. A failure is observable by the software users or other stakeholders. A failure is caused by one or more defects in the software. In this case, the software not being able to meet the desired response time is an example of a failure, as it deviates from the stated requirement and affects the user experience. It is not a defect, which is a flaw in the software that causes the failure. It is not an error, which is a human action that produces an incorrect result. It is not a non-defect, as it clearly violates a specified requirement. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 4.

NEW QUESTION 215
Given the following examples of entry and exit criteria:
* 1. A defined level of code coverage has been achieved
* 2. The test automation tool has been installed and properly configured
* 3. The number of unresolved defects is within the predefined limit
* 4. The performance test environment has been set-up and is available
* 5. The user stories have proper acceptance criteria defined
* 6. The testing budget has been spent and the project sponsor bears the risk of not testing any further
Which of the following BEST categorizes them as entry and exit criteria:

A. Entry criteria - 2, 4, 5 Entry criteria -1, 3, 4
B. Entry criteria - 2, 4 Entry criteria - 2, 4, 5, 6
C. Exit criteria -1,3,6 Exit criteria - 2, 5. 6
D. Exit criteria -1,3,56 Exit criteria -1,3

Answer: A

Explanation:

Entry and exit criteria are used to determine when to start and stop testing, respectively.

? Entry Criteria:

? Exit Criteria:

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, these criteria help in effectively managing the testing process6†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 217

......

Relate Links

Contact us