ITCertTest

ITCERTTEST
IT Exam study guide / simulations

http://www.itcerttest.com

IT exam study guide / simulations

Exam : **CPSA-FL**

Title : ISAQB Certified Professional for Software Architecture - Foundation Level

Vendor : ISQI

Version : DEMO

NO.1 Concerning external interfaces, Postel's law suggests: "Be conservative in what you do, be liberal in what you accept from others." Assume that Postel's law has been consistently applied in your system. (Assign all answers.)

true	false	
○	○	A) Response time of the system is reduced
○	○	B) Implementation effort increases
○	○	C) Usability of the system is reduced
○	○	D) Robustness of the system is increased
○	○	E) The integrity of the data transferred via interfaces is increased
○	○	F) Availability of the system is reduced due to potentially bad quality of input data

Answer:

true	false	
○	○	A) Response time of the system is reduced
○	○	B) Implementation effort increases
○	○	C) Usability of the system is reduced
○	○	D) Robustness of the system is increased
○	○	E) The integrity of the data transferred via interfaces is increased
○	○	F) Availability of the system is reduced due to potentially bad quality of input data

Explanation:

A) FalseB) FalseC) FalseD) TrueE) TrueF) False

Postel's law, also known as the robustness principle, is a guideline in software engineering that advises to "be conservative in what you do, be liberal in what you accept from others"1. This principle has implications for system design and interaction with external interfaces:

A) Response time of the system is reduced: This is false. Postel's law does not directly relate to the response time of a system1.

B) Implementation effort increases: This is false. Being liberal in what is accepted can actually simplify implementation because the system is designed to handle a wider range of inputs without failure1.

C) Usability of the system is reduced: This is false. Postel's law aims to increase robustness and interoperability, which can enhance usability by making the system more resilient and accommodating1.

D) Robustness of the system is increased: This is true. By being conservative in outputs and liberal in inputs, the system becomes more robust, handling a variety of inputs without error1.

E) The integrity of the data transferred via interfaces is increased: This is true. Accepting a wide range of inputs and adhering strictly to output specifications helps maintain data integrity across different systems1.

F) Availability of the system is reduced due to potentially bad quality of input data: This is false. Postel's law suggests that the system should be designed to handle poor quality input data gracefully,

thus maintaining availability1.
Applying Postel's law helps create systems that are more tolerant of input variations and strict in their outputs, contributing to overall system robustness and reliability1.

NO.2 Conway's law sometimes is referred to as "If you have four teams working on the compiler, you'll get a 4-pass compiler." Which interpretations of this law are true, which are false? (Assign all answers.)

true	false	
○	○	A) You need four teams to build a compiler.
○	○	B) Structures of software architecture and associated organisation are congruent.
○	○	C) Software architecture is particularly important when creating compilers.
○	○	D) Certain types of software are not suitable for the use of software architecture.

Answer:

true	false	
○	○	A) You need four teams to build a compiler.
○	○	B) Structures of software architecture and associated organisation are congruent.
○	○	C) Software architecture is particularly important when creating compilers.
○	○	D) Certain types of software are not suitable for the use of software architecture.

Explanation:
The interpretations of Conway's law that are true and false are as follows:
* True:
* Structures of software architecture and associated organization are congruent (B).
* Software architecture is particularly important when creating compilers .
* False:
* You need four teams to build a compiler (A).
* Certain types of software are not suitable for the use of software architecture (D).
Conway's law suggests that the structure of systems designed by an organization will mimic the communication structures of that organization1. This means that the way teams are organized and communicate will influence the design of the software they are developing. Here's an explanation of each interpretation:
* True Statements:
* (B): The law implies that the technical structure of a system will reflect the social boundaries of the organization that produced it, which means the architecture of the software and the
* organization will be congruent1.
* : When creating complex systems like compilers, the architecture is influenced by organizational structures, making it crucial to consider how teams are organized and how they communicate1.
* False Statements:
* (A): The number of teams does not dictate the necessity of a multi-pass compiler; it's a metaphorical example to illustrate that organizational structure can influence technical design1.
* (D): All types of software can benefit from thoughtful software architecture; the statement is a

misinterpretation of Conway's law, which does not deem any software unsuitable for architectural practices1.
Conway's law is a valuable consideration in software engineering, reminding us that organizational decisions can have a significant impact on the design and functionality of the software.
References:
* Wikipedia article on Conway's law1.
* ThinkingLabs article on Shades of Conway's Law2.
* Dovetail article on What Is Conway's Law

NO.3 For which quality characteristics is the software architect responsible?
Please name the two characteristics that best match the role of the software architect. (Choose two.)
A. The technical quality of the software implementation
B. The software is free of errors
C. The suitability of the software design for its purpose
D. The performance of the software
Answer: C,D
Explanation:
The software architect is typically responsible for the following quality characteristics:A. The performance of the software - Ensuring that the software meets performance criteria and behaves efficiently under specified conditions is a primary responsibility of the software architect.C. The suitability of the software design for its purpose - Architects must ensure that the design effectively meets the intended functions and requirements, addressing the suitability of the software for its purpose.
These answers are derived from established software architecture practices and principles, focusing on the roles and responsibilities of software architects in ensuring quality and effective documentation.

NO.4 Which of the following statements about the coupling between building blocks are correct? (Assign all answers.)

true	false	
○	○	A) A high degree of coupling of a building block reduces its reusability.
○	○	B) Low coupling of a building block improves the ability to meet functional requirements.
○	○	C) Low cohesion often leads to high coupling.
○	○	D) Loose coupling often leads to less effort for making changes.
○	○	E) For call dependencies, the degree of coupling is independent of the direction of the call.
○	○	F) In object-oriented programming languages, inheritance reduces coupling.

Answer:

true	false	
☑	○	A) A high degree of coupling of a building block reduces its reusability.
○	☑	B) Low coupling of a building block improves the ability to meet functional requirements.
☑	○	C) Low cohesion often leads to high coupling.
☑	○	D) Loose coupling often leads to less effort for making changes.
○	☑	E) For call dependencies, the degree of coupling is independent of the direction of the call.
○	☑	F) In object-oriented programming languages, inheritance reduces coupling.

Explanation:
A) CorrectB) IncorrectC) CorrectD) CorrectE) IncorrectF) Incorrect
The concept of coupling between building blocks is crucial in software architecture, and the iSAQB SOFTWARE ARCHITECTURE - FOUNDATION LEVEL provides guidance on this topic. Here's an analysis of the statements based on the curriculum:
A) A high degree of coupling of a building block reduces its reusability: This is correct. High coupling means that a building block is heavily dependent on other components, which can limit its ability to be reused in different contexts1.
B) Low coupling of a building block improves the ability to meet functional requirements: This statement is incorrect. While low coupling can contribute to a more maintainable and flexible architecture, it does not directly improve the ability to meet functional requirements1.
C) Low cohesion often leads to high coupling: This is correct. Cohesion refers to how closely related the responsibilities of a module are. Low cohesion can result in modules that perform a wide range of actions, which often leads to higher coupling with many other modules1.
D) Loose coupling often leads to less effort for making changes: This is correct. Loose coupling allows for easier modification of a system with minimal impact on other components, thus reducing the effort required for changes1.
E) For call dependencies, the degree of coupling is independent of the direction of the call: This statement is incorrect. The degree of coupling can be affected by the direction of the call, as it determines which module controls the interaction1.
F) In object-oriented programming languages, inheritance reduces coupling: This statement is incorrect.
Inheritance can actually increase coupling because it creates a direct dependency between the superclass and the subclass1.
These insights are aligned with the principles outlined in the iSAQB SOFTWARE ARCHITECTURE - FOUNDATION LEVEL curriculum, which emphasizes the importance of understanding and managing coupling for effective software architecture design1.