



Sample Questions

1. ISO/IEC 12207:1995 describes which of the following:

- I. A standard software life cycle process
- II. A framework for software life cycle processes
- III. A process for defining and improving software life cycle processes
- IV. A process for the acquisition of systems and software products

[a] I and III

[b] II and III

[c] II and IV

[d] III and IV

2. The typical elements of the requirements engineering process are:

- I. problem analysis
- II. software design
- III. analysis of staffing needs
- IV. external behavior specification

[a] I and IV only

[b] II and III only

[c] I, III, and IV only

[d] I, II, and III only

3. Domain engineering consists of which of the following sets of activities?

[a] Data modeling, requirements elicitation, and requirements verification

[b] Analysis, modeling, and infrastructure development

[c] Classifying components, designing a library, and populating the library

[d] Understanding the user community, building an E-R model, and establishing traceability

4. What activities need to be completed prior to performing the allocation of system requirements?

I. System architecture

II. System requirements and system architecture

III. System requirements and software requirements

[a] I

[b] II

[c] I and III

[d] III

5. In a large real-time systems project, the following items (or components) of the system were included in the component list produced as part of the architectural design:

I. Input signal pre-processing

II. Main control processing

III. Network interfacing

Where would the non-functional requirement of reliability be allocated?

[a] I

[b] I and II

[c] II and III

[d] I, II and III

6. Suppose that the problem of defending a nation against enemy missiles is undergoing problem analysis. A defensive missile can be examined from the perspectives of the missile system operator, the commander making the decision, the intelligence analyst, and the enemy missile. This analysis is an example of:

- [a] Abstraction
- [b] Partitioning
- [c] Elicitation
- [d] Projection

7. According to IEEE Standard 830-1998, which of the following are characteristics of a good Software Requirements Specification?

- I. Complete
- II. Hierarchical
- III. Verifiable
- IV. Provable
- V. Traceable

- [a] I, II, and IV
- [b] I, III, and V
- [c] II, III, and IV
- [d] II, III, and V

8. Which of the following is NOT an attribute of a well-written software requirements specification?

- [a] Everything that the software is supposed to do is included in the specification.
- [b] Every requirement stated therein has only one interpretation.
- [c] Some of the requirements specify a target software architecture.
- [d] All of the requirements are understandable by non-computer-specialist customers.

9. A software system that has been validated but not verified may not:

- [a] meet user expectations
- [b] meet specified requirements
- [c] be testable
- [d] be reusable

10. An object identified during object-oriented analysis:

- [a] Is always preserved identically as an object in the subsequent design.
- [b] Has no relationship with any object identified during subsequent design.
- [c] May have a mapping to one or more objects identified during subsequent design.
- [d] Has no attributes identified for it.

11. A design that can be modified easily to run on a variety of hardware and software environments is highly:

- [a] portable
- [b] interoperable
- [c] profitable
- [d] usable

12. Suppose you have an object-oriented programming language that adopts the following strategy. An “equal” method may be defined and used to overload the equality operator (e.g., “==”). If the equality operator is not overloaded in this fashion for a particular class, then use of the equality operator on objects of that class means that a bitwise comparison of the objects is done. Which of the following statements is true?

- [a] The default bitwise comparison always gives wrong results.
- [b] The default bitwise comparison sometimes says two objects are equal when they logically should not be considered equal.
- [c] The default bitwise comparison sometimes says two objects are unequal when they logically should be considered equal.
- [d] Both [b] and [c] are true.

13. The elements of the software architecture of a computing system include:

- I. Software components
- II. Class diagrams
- III. Connectors expressing relationships between software components
- IV. Entity-relationship diagrams

[a] I and II

[b] I and III

[c] I, III, and IV

[d] I, II, III, and IV

14. One component of a software project requires the determination of the k^{th} largest element of a dynamically changing set with m elements, where $1 \leq k \leq 10$ and $1 \leq m \leq 10^6$. Two alternative algorithmic approaches to this problem are (1) to sort the elements of the set each time an element is requested, and (2) to use a divide-and-conquer approach to find the requested element directly. Which method is better, and how much better will it be?

[a] sorting, by a factor of m

[b] direct approach, by a factor of m

[c] direct approach, by a factor of $\log m$

[d] sorting, by a factor of $\log m$

15. The software design description details all of the following EXCEPT the:

[a] software item input/output description

[b] rationale for software item design

[c] concept of execution, including data flow and control flow

[d] software life cycle model

16. Which of these diagrams is NOT used in performing object-oriented design?

- [a] class
- [b] activity
- [c] use-case
- [d] sequence

17. A good design will accomplish all of the following EXCEPT:

- [a] implement all explicit and implicit requirements
- [b] provide information for developers and testers
- [c] address data, functional, and behavioral domains from an implementation perspective
- [d] set up a software review methodology

18. In software user documentation, specific warnings must be placed:

- [a] immediately before the action that requires the warning
- [b] in the graphics for critical processes
- [c] on the back of the title page
- [d] in a separate section

19. A company is designing software for a medical instrument. They should pay particular attention to the following hazards EXCEPT:

- [a] the software may contain an error
- [b] the human interface may be ambiguous
- [c] the algorithm may be inefficient
- [d] the program may stop

20. Layered architectures are normally used for Web-based enterprise applications for which of the following reasons:

- I. Coding
- II. Portability
- III. Ease of system administration

[a] II only

[b] I and II only

[c] I and III only

[d] II and III only

21. All of the following are major design aspects of a program component EXCEPT:

[a] control structures

[b] algorithms

[c] requirements

[d] data structures

22. The main advantage of structured programming is:

[a] it is more efficient

[b] it tends to be more reliable

[c] it is easier to write

[d] it can be flowcharted

23. The best approach for the interchange of a large amount of structured data on the Web is to use:

[a] eXtensible Markup Language (XML)

[b] Hyperlink Markup Language (HTML)

[c] Component Object Model (COM)

[d] Java and Java-based technologies

24. The internal schema (physical model) of a database differs from the external schema (logical model) in that:
- [a] it describes the database
 - [b] it is not expected to change frequently
 - [c] it describes the physical structure of the database
 - [d] it is based on some data model
25. The most useful aspect of a help desk is the ability to:
- [a] access previously solved problems
 - [b] access source code
 - [c] access database design
 - [d] access requirement specifications
26. The most important aspect of structural testing (also known as “white-box” testing) is its ability to:
- [a] Reveal the presence of defects in various parts of the code.
 - [b] Establish the correctness of the module.
 - [c] Prove that every statement in the module is reachable.
 - [d] Prove that the module has a low cyclomatic complexity.
27. Beta testing is:
- [a] An inexpensive way to conduct a software quality assurance program
 - [b] A reasonable way to check for software compatibility problems with various machine configurations
 - [c] The best way to get focused user feedback to support user interface improvements
 - [d] A guaranteed way to obtain positive product publicity

28. Which of the following statements is true about unit testing of object-oriented systems?

- [a] Unit testing best takes place at the method level.
- [b] Unit testing best takes place at the class level.
- [c] Unit testing is infeasible.
- [d] Unit testing focuses on testing individual attributes rather than methods.

29. Which of the following are typically inputs to the system-level testing process

- I System Performance Requirements
- II System Functional Requirements
- III Program Code
- IV System Design Specifications

- [a] I and II
- [b] I, II and IV
- [c] I
- [d] I, II, III and IV

30. Consider the following code fragment:

```
input x;  
if x > 0 then  
    output x+1;  
else  
    output x-1;  
end if;  
while x > 5 then  
    if x = 10 then  
        output "blah"  
    else  
        output "ugh"  
    end if;  
    x = x - 1  
end while;
```

The minimal number of test cases to achieve branch coverage of this program is:

- [a] 1
- [b] 2
- [c] 3
- [d] 4

31. All of the following dynamic testing techniques are based on program structure EXCEPT:

- [a] mutation analysis
- [b] random testing
- [c] dataflow testing
- [d] automatic path-based test data generation

32. Suppose you have a binary search routine that handles a special case list size of 13 elements. If the structure of the algorithm is used to identify equivalence classes, what is the minimum number of test cases required to test all equivalence class boundaries of this routine?

- [a] 2
- [b] 3
- [c] 4
- [d] 5

33. In designing a set of test cases that execute all linearly independent paths in a program, which of the following statements is true?

- [a] The set of needed test cases can typically be determined by a simple visual inspection of the program.
- [b] Static analysis tools exist that can automatically generate the required test cases.
- [c] It is often necessary to instrument the code and dynamically determine during testing which paths have not yet been executed.
- [d] Special care must be taken to design test cases that execute dead code.

34. Suppose you have a software routine that controls a temperature sensor that drives a warning light on an airplane to notify the pilot of potential icing problems. The specification says that this light is to glow red whenever the temperature is strictly less than -20 degrees Celsius, yellow between -20 and 10 degrees Celsius (inclusive), and green for all temperatures strictly greater than 10 degrees Celsius. What is the minimum number of test cases needed to achieve statement coverage of this specification?

- [a] 2
- [b] 3
- [c] 6
- [d] 8

35. A test oracle:

- [a] Generates predictions of expected test results.
- [b] Manages the running of program tests.
- [c] Generates test data for the program to be tested.
- [d] Counts the number of times that a particular statement has been executed.

36. Which of the following types of test plans is most likely to arise from the requirements specification process?

- [a] System integration test plan
- [b] Acceptance test plan
- [c] Sub-system integration test plan
- [d] Module unit test plan

37. Mean time to change relates to the time between:

- [a] source code version changes
- [b] a change request and distribution of information about the change to all users
- [c] distribution to all users and a change request
- [d] distribution to all users and source code version change

38. Corrective maintenance generally includes all of the following EXCEPT:

- [a] requirements analysis
- [b] change control management
- [c] traceability analysis
- [d] code inspection

39. Which of the following is not appropriate to be in a Software Maintenance Plan:

- [a] The scope of the Software maintenance process
- [b] Problems that the software maintenance process has corrected
- [c] Resources allocated to the software maintenance process
- [d] How to track software maintenance performance

40. Software configuration management is:

- [a] A technique for naming records and files
- [b] A discipline for object-oriented analysis
- [c] A discipline for managing the evolution of computer software products
- [d] A methodology for designing computer system interfaces

41. A configuration status report consists of:

- I. a listing of the approved configuration identification
- II. status of proposed changes to the configuration
- III. implementation status of approved changes
- IV. results of test runs
- V. quality assessment report

[a] I, II, and V

[b] I, II, and III

[c] II, IV, V

[d] I, II, III, IV

42. A software project is estimated to take a nominal twelve months with a four-person team. Project stakeholders desire the project to be finished as soon as possible and suggest using an eight-person team to achieve that desire. Given this change, the stakeholders can expect the project to be completed in:

[a] 9 months

[b] 4 months

[c] 7 months

[d] 6 months

43. Which of the following statements regarding peer reviews is correct?

[a] They are aimed at validating and approving requirements to establish a baseline.

[b] They are held after implementation is complete.

[c] They look at the product during a specific time period of a life cycle.

[d] They are performed when the appropriate application components have been completed.

44. What must you achieve in order to satisfy the intent of a key process activity?

[a] Demonstrate all of its Abilities to Perform

[b] Accomplish all of its Goals

[c] Accomplish all of its Activities Performed

[d] Perform the stated Measurement and Analysis

45. Once a Software V&V Plan has been developed for a process:

[a] It is never changed

[b] It can be changed up until a contract is started

[c] It can be changed until actual development begins

[d] Can be changed throughout the life cycle

46. An SQA plan has been prepared for a project and includes the following topics.

- I. Tools for performing QA activities
- II. Procedures for contract review
- III. List of project team members
- IV. The activities and tasks required for performing problem resolution
- V. Schedule for conducting QA activities

Which option covers the correct set of topics for a software quality assurance plan?

[a] All

[b] I,II, III and V

[c] I, IV, and V

[d] I,II,IV and V

47. An Object-Oriented UML Activity diagram differs from a flowchart in that:

[a] It shows actions in boxes and connections between boxes.

[b] It has a starting point and one or more ending points.

[c] It shows concurrent activity.

[d] It shows data as well as functionality.

48. In Object-Orientation, Polymorphism means:

[a] There can be many objects in the design.

[b] Methods can be changed in many ways.

[c] Many objects can be instantiated of a class

[d] Objects can implement the same method in many ways.

49. In Object-Orientation, a CRC card contains:

- [a] Constraints, requirements and containers
- [b] Classes, responsibilities and collaborators
- [c] Conditions, relationships and code
- [d] None of the above

50. In Object-Orientation, the difference between an instance and a class is:

- [a] An instance declares data items that are called attributes.
- [b] An instance declares subroutines that are called methods.
- [c] An instance has memory space allocated to it
- [d] An instance has a name that can be referred to.

Sample Question Answers

1. B

Rationale: Options II and III are supported in Clause 1.1 ("Purpose"). Option I does not work because 12207 does not specify a specific standard SLC process. Option IV does not work: although 12207 applies to the acquisition of systems and software (see Clause 1.2), it does not specify a process for system or software acquisition.

2. A

Rationale: Options I and IV are cited in the reference, Marciniak, *Encyclopedia of Software Engineering*, 1994, p. 1044

3. B

Rationale: Domain Engineering is the discipline of understanding, analyzing, modeling a problem domain and creating a set of core assets for the product line that can be reused for each of the products in the product line. Following URL has excellent material on this subject. http://www.sei.cmu.edu/domain-engineering/domain_engineering.html

4. B

Rationale: System Architecture alone is not enough requirements must also be known.

5. D

Rationale: Reliability is an "emergent" property and must be allocated to all processing components.

6. D

Rationale: Projection of B onto A means that A is one view of B, while abstracting A to B means that A is an example of B and partitioning means that A is a part of B.

7. B

Rationale: "Complete," "Verifiable," and "Traceable" are all characteristics of a good SRS, while "Hierarchical" is applicable to architecture, not to requirements.

8. C

Rationale: A, B and D ARE attributes (Alan Davis, *Software Requirements: Objects, Functions and States*, pp. 181-194.)

9. B

Rationale: Validation guarantees that a system meets user needs. Verification is an activity that allows us to make sure that the system meets its stated requirements.

10. C

Rationale: The object may be modified, be preserved, typically has attributes and will often have a mapping to one or more design objects.

11. A

Rationale: There is little effort required to transfer between different platforms, therefore it is portable. No information given on the coupling of systems or on the effort required to learn and operate the system and it is only profitable if there is a market for the product; no information is given on this.

12. C

Rationale: Bitwise comparison is sometimes perfectly valid, particularly for simple objects. If two objects are bitwise identical, then there is no way to distinguish their state. Under any reasonable definition of equality, the objects should be considered equal.

A simple example says that that C is true: Consider a stack object implemented as an array with a top pointer. Consider stack `s1 = new.push(10).push(20).pop()` and `s2 = new.push(10)`. The arrays will not be bitwise identical, yet by a reasonable definition of "equal" for stacks, the two objects should be treated as equal.

13. B

Rationale: Software architectures consist of components and connectors.

14. C

Rationale: The best sorting algorithm will run in time $O(m \log m)$, and the (direct) divide-and-conquer algorithm will run in time $O(m)$. The direct approach is therefore better by a factor of $\log m$.

15. D

Rationale: The software life cycle model is a typical part of the project management plan and the software life cycle model description.

Options a, b, and c are part of the standard definition of the SDD.

16. C

Rationale: Use cases are used for requirement elicitation.

17. D

Rationale: The review methodology is not part of the design process.

18. A

Rationale: Accurate placement of warnings is essential for user safety.

19. C

Rationale: Hazards (a), (b), and (d) may lead to life-threatening situations. It is extremely unlikely that an inefficient algorithm will have that effect.

20. D

Rationale: Coding is more than one level removed from the architecture.

21. C

Rationale: A designer considers control structures, algorithms, and data structures. Requirements are the input to the design phase.

22. B

Rationale: The work of Dijkstra and others in the late 1960's focuses on structured programming for program reliability.

23. A

Rationale: HTML is too simple; COM and Java are not relevant.

24. C

Rationale: A, B, and D apply to both internal and external schemas

25. A

Rationale: The primary purpose of a help desk is to give access to previously solved problems.

26. A

Rationale: Structural testing reveals defects; but does not prove that the module is correct or reachability. Cyclomatic complexity can be established by static analysis – testing is not necessary.

27. B

Rationale: It is difficult to configure a test environment that encompasses every possible machine configuration. Beta testing is arguably not cost effective; (see McConnell, S., *Software Project Survival Guide*, pp.136-138), positive product publicity may not be forthcoming if the product has many problems and it is too late in the testing phase to make significant user interface improvements, and is certainly not the “best” way to obtain this feedback

28. B

Rationale: Methods cannot typically be tested separately; they must interoperate with the rest of the class.

29. A

Rationale: The test team does not use designs nor code to develop test procedures or test plans. System design specifications may well be an input to

acceptance testing

30. B

Rationale: Empirical validation of this answer is immediate.

31. B

Rationale: Only random testing fails to use program structure.

32. D

Rationale: Element 1 and Element mid-1 are boundaries of the left side of the list. Element mid+1 and 13 are boundaries of the right side of the list. The mid element is a class by itself.

33. C

Rationale: From Sommerville, I. *Software Engineering*, 6th Edition, p. 451.

34. B

Rationale: There are three logical branches to this specification, one for each type of output.

35. A

Rationale: From Sommerville, I. *Software Engineering*, 6th Edition, p. 463. A test manager/director would manage the running of the tests.

36. B

Rationale: Immediate from the reference, Sommerville, I. *Software Engineering*, 6th Edition, p. 424. Acceptance tests are requirements based. The other types of tests mentioned are more dependent on the software architecture, which is not defined until later in the lifecycle.

37. B

Rationale: MTTC is the time to analyze the change request, design, implement, test, and distribute the change.

38. A

Rationale: Requirements analysis is usually not needed since the customer requirements remain the same. This helps test the difference between the development and maintenance processes. Traceability analysis is likely to be needed, unless the change was restricted only to source code.

39. B

Rationale: A software maintenance plan is prepared before development, so there are no results in it

40. C

Rationale: The definition in [c] is straight out of IEEE 1042, 1987, p. 10, clause 2.1.2. The others are irrelevant.

41. B

Rationale: IEEE Standard 610.12

42. A

Rationale: Practitioners should know that doubling the size of a project team will reduce the schedule somewhat, but not halve it. Maximal schedule reduction is about 75%.

43. D

Rationale: In-process reviews (peer, walkthrough, inspections) are conducted during each lifecycle phase.

44. B

Rationale: Abilities to perform are necessary but not sufficient, goals are both necessary and sufficient, activities performed are sufficient but not necessary, measurement and analysis are useful but neither necessary nor sufficient.

45. D

Rationale: "The Software V&V plan may require updating throughout the life cycle."

46. D

Rationale: Item III is part of the project plan, not the SQA plan, item II is part of the SQA plan. All four items are required in an SQA plan

47. C

Rationale: An Activity diagram shows concurrent activity while a flowchart shows only one logical path.

48. D

Rationale: Definition of polymorphism

49. B

Rationale: Definition of a CRC card

50. C

Rationale: A class is a blueprint for creating instances of objects. It describes all the characteristics of each object: attributes and methods, and it has a name. But it is a blueprint, and therefore does not have any memory space allocated to it.