# Chapter 1 Multiple Choice Quiz

1. Which question no longer concerns the modern software engineer?
   *a. Why does computer hardware cost so much?*
   b. Why does software take a long time to finish?
   c. Why does it cost so much to develop a piece of software?
   d. Why can't software errors be removed from products prior to delivery?
2. Today the increased power of the personal computer has brought about an abandonment of the practice of team development of software.
   a. True
   *b. False*
3. Software is a product and can be manufactured using the same technologies used for other engineering artifacts.
   a. True
   *b. False*
4. Software deteriorates rather than wears out because
   a. Software suffers from exposure to hostile environments
   b. Defects are more likely to arise after software has been used often
   *c. Multiple change requests introduce errors in component interactions*
   d. Software spare parts become harder to order
5. Most software continues to be custom built because
   a. Component reuse is common in the software world.
   b. Reusable components are too expensive to use.
   c. Software is easier to build without using someone else's components.
   *d. Off-the-shelf software components are unavailable in many application domains.*
6. The nature of software applications can be characterized by their information
   a. complexity
   b. content
   c. determinacy
   *d. both b and c*
7. Modern software applications are so complex that it is hard to develop mutually exclusive category names.
   *a. True*
   b. False
8. The so called "new economy" that gripped commerce and finance during the 1990s died and no longer influences decisions made by businesses and software engineers.
   a. True
   *b. False*
9. The functionality of most computer systems does not need to be enhanced the lifetime of the system.
   a. True
   *b. False*

10. Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind.
    *a. True*
    b. False
11. Most software development projects are initiated to try to meet some business need.
    *a. True*
    b. False
12. In general software only succeeds if its behavior is consistent with the objectives of its designers.
    a. True
    *b. False*

# Chapter 2 Multiple Choice Quiz

1. Which of the items listed below is not one of the software engineering layers?
    a. Process
    *b. Manufacturing*
    c. Methods
    d. Tools

2. Software engineering umbrella activities are only applied during the initial phases of software development projects.
    a. True
    *b. False*

3. Which of these are the 5 generic software engineering framework activities?
    *a. communication, planning, modeling, construction, deployment*
    b. communication, risk management, measurement, production, reviewing
    c. analysis, designing, programming, debugging, maintenance
    d. analysis, planning, designing, programming, testing

4. Process models are described as agile because they
    a. eliminate the need for cumbersome documentation
    *b. emphasize maneuverability and adaptability*
    c. do not waste development time on planning activities
    d. make extensive use of prototype creation

5. Which of these terms are level names in the Capability Maturity Model?
    a. Performed
    b. Repeated
    c. Reused
    d. Optimized
    *e. both a and d*

6. Software processes can be constructed out of pre-existing software patterns to best meet the needs of a software project.
   *a. True*
   b. False

7. Which of these are standards for assessing software processes?
   a. SEI
   b. SPICE
   c. ISO 19002
   d. ISO 9001
   *e. a and b*

8. The best software process model is one that has been created by the people who will actually be doing the work.
   *a. True*
   b. False

9. Which of these is not a characteristic of Personal Software Process?
   a. Emphasizes personal measurement of work product
   *b. Practitioner requires careful supervision by the project manager*
   c. Individual practitioner is responsible for estimating and scheduling
   d. Practitioner is empowered to control quality of software work products

10. Which of these are objectives of Team Software Process?
    a. Accelerate software process improvement
    b. Allow better time management by highly trained professionals
    c. Build self-directed software teams
    d. Show managers how to reduce costs and sustain quality
    *e. b and c*

11. Process technology tools allow software organizations to compress schedules by skipping unimportant activities.
    a. True
    *b. False*

12. It is generally accepted that one cannot have weak software processes and create high quality end products.
    *a. True*
    b. False

# Chapter 3 Multiple Choice Quiz

1. The linear sequential model of software development is
   *a. A reasonable approach when requirements are well defined.*
   b. A good approach when a working program is required quickly.

c. The best approach to use for projects with large development teams.
d. An old fashioned model that cannot be used in a modern context.

2. The linear sequential model of software development is also known as the
   a. Classical life cycle model
   b. Fountain model
   c. Spiral model
   d. Waterfall model
   *e. a and d*

3. The incremental model of software development is
   a. A reasonable approach when requirements are well defined.
   *b. A good approach when a working core product is required quickly.*
   c. The best approach to use for projects with large development teams.
   d. A revolutionary model that is not used for commercial products.

4. 4: The rapid application development model is
   a. Another name for component-based development.
   b. A useful approach when a customer cannot define requirements clearly.
   *c. A high speed adaptation of the linear sequential model.*
   d. All of the above.

5. Evolutionary software process models
   a. Are iterative in nature
   b. Can easily accommodate product requirements changes
   c. Do not generally produce throwaway systems
   *d. All of the above*

6. The prototyping model of software development is
   a. A reasonable approach when requirements are well defined.
   *b. A useful approach when a customer cannot define requirements clearly.*
   c. The best approach to use for projects with large development teams.
   d. A risky model that rarely produces a meaningful product.

7. The spiral model of software development
   a. Ends with the delivery of the software product
   b. Is more chaotic than the incremental model
   *c. Includes project risks evaluation during each iteration*
   d. All of the above
8. The concurrent development model is
   a. Another name for the rapid application development model.
   *b. Often used for the development of client/server applications.*
   c. Only used for development of parallel or distributed systems.
   d. Used whenever a large number of change requests are anticipated.

9. The component-based development model is
   a. Only appropriate for computer hardware design.
   b. Not able to support the development of reusable components.
   *c. Works best when object technologies are available for support.*
   d. Not cost effective by known quantifiable software metrics.

10. The formal methods model of software development makes use of mathematical methods to
   a. Define the specification for computer-based systems
   b. Develop defect free computer-based systems
   c. Verify the correctness of computer-based systems
   *d. All of the above*

11. Which of these is not one of the phase names defined by the Unified Process model for software development?
   a. Inception phase
   b. Elaboration phase
   c. Construction phase
   *d. Validation phase*

12. In the Unified Process model requirements are determined iteratively and may span more than one phase of the process.
   *a. True*
   b. False

# Chapter7 Multiple Choice Quiz

1. Requirements engineering is a generic process that does not vary from one software project to another.
   *a. True*
   b. False

2. During project inception the intent of the of the tasks are to determine
   a. basic problem understanding
   b. nature of the solution needed
   c. people who want a solution
   d. none of the bbove
   *e. a, b, and c*

3. Three things that make requirements elicitation difficult are problems of
   a. budgeting
   b. scope
   c. understanding

d. volatility
*e. b, c and d*

4. The result of the requirements engineering elaboration task is an analysis model that defines which of the following problem domain(s)?
   a. information
   b. functional
   c. behavioral
   *d. all of the above*

5. It is relatively common for different customers to propose conflicting requirements, each arguing that his or her version is the right one.
   *a. True*
   b. False

6. The system specification describes the
   *a. Function, performance and constraints of a computer-based system*
   b. implementation of each allocated system
   c. element software architecture
   d. time required for system simulation

7. The best way to conduct a requirements validation review is to
   a. examine the system model for errors
   b. have the customer look over the requirements
   c. send them to the design team and see if they have any concerns
   *d. use a checklist of questions to examine each requirement*

8. The use of traceability tables helps to
   a. debug programs following the detection of run-time errors
   b. determine the performance of algorithm implementations
   *c. identify, control, and track requirements changes*
   d. none of the above

9. A stakeholder is anyone who will purchase the completed software system under development.
   a. True
   *b. False*

10. The job of the requirements engineer is to categorize all stakeholder information in a way that allows decision makers to choose an internally consistent set of requirements.
    *a. True*
    b. False

11. The nature of collaboration is such that all system requirements are defined by

consensus of a committee of customers and developers.

a. True
*b. False*

12. Which of the following is not one of the context-free questions that would be used during project inception?

a. What will be the economic benefit from a good solution?
*b. Who is against this project?*
c. Who will pay for the work?
d. Who will use the solution?

13. In collaborative requirements gathering, the facilitator

a. cannot be a member of the software team
b. cannot be a customer
*c. controls and facilitates the process*
d. must be an outsider

14. Which of the following is not one of the requirement classifications used in Quality Function Deployment (QFD)?

a. exciting
b. expected
*c. mandatory*
d. normal

15. Developers and customers create use-cases to help the software team understand how different classes of end-users will use functions.

*a. True*
b. False

16. The work products produced during requirement elicitation will vary depending on the

a. size of the budget
*b. size of the product being built*
c. software process being used
d. stakeholders needs

17. Use-case actors are always people, never system devices.

a. True
*b. False*

18. Which of following is not a UML diagram used creating a system analysis model?

*a. activity diagram*
b. class diagram
c. dataflow diagram

d. state diagram

19. Analysis patterns facilitate the transformation of the analysis model into a design model by suggesting reliable solutions to common problems.
    a. True
    *b. False*

20. In win-win negotiation, the customer's needs are met even though the developer's need may not be.
    a. True
    *b. False*

21. In requirements validation the requirements model is reviewed to ensure its technical feasibility.
    a. True
    *b. False*

# Chapter 8 Multiple Choice Quiz

1. Which of the following is not an objective for building an analysis model?
    a. define set of software requirements that can be validated
    b. describe customer requirements
    *c. develop an abbreviated solution for the problem*
    d. establish basis for software design

2. Object-oriented domain analysis is concerned with the identification and specification of reusable classes within an application domain.
    *a. True*
    b. False

3. The data dictionary contains descriptions of each software
    a. control item
    b. data object
    c. diagram
    d. notation
    *e. both a and b*

4. Which of these is not an element of an object-oriented analysis model?
    a. Behavioral elements
    b. Class-based elements
    *c. Data elements*
    d. Scenario-based elements

5. In analysis models the only data objects that need representation are those that will be implemented using software classes.

a. True
*b. False*

6. The values that are assigned to an object's attributes make that object unique.
*a. True*
b. False

7. The relationships shown in a data model must be classified to show their
a. cardinality
b. directionality
c. modality
d. probability
*e. both a and c*

8. The entity relationship diagram
*a. depicts relationships between data objects*
b. depicts functions that transform the data flow
c. indicates how data are transformed by the system
d. indicates system reactions to external events

9. A generalized description of a collection of similar objects is a
*a. class*
b. instance
c. subclass
d. super class

10. Operations are object procedures that are invoked when an object receives a message.
*a. True*
b. False

11. In many cases there is no need to create a graphical representation of a usage scenario.
*a. True*
b. False

12. UML activity diagrams are useful in representing which analysis model elements?
a. Behavioral elements
b. Class-based elements
c. Flow-based elements
*d. Scenario-based elements*

13. The data flow diagram

a. depicts relationships between data objects
b. depicts functions that transform the data flow
c. indicates how data are transformed by the system
d. indicates system reactions to external events
*e. both b and c*

14. Control flow diagrams are
a. needed to model event driven systems.
b. required for all systems.
c. used in place of data flow diagrams.
d. useful for modeling real-time systems.
*e. both a and d*

15. The data flow diagram must be augmented by descriptive text in order to describe the functional requirements for a software product.
*a. True*
b. False

16. Which of the following should be considered as candidate objects in a problem space?
a. events
b. people
c. structures
*d. all of the above*

17. Attributes cannot be defined for a class until design has been completed.
a. True
*b. False*

18. Which of the following is not one of the broad categories used to classify operations?
a. computation
b. data manipulation
c. event monitors
*d. transformers*

19. Which of the following items does not appear on a CRC card?
a. class collaborators
b. class name
*c. class reliability*
d. class responsibilities

20. Class responsibilities are defined by
a. its attributes only

b. its collaborators
c. its operations only
*d. both its attributes and operations*

21. An analysis package involves the categorization of analysis model elements into useful groupings.
*a. True*
b. False

22. Events occur whenever a(n)
*a. actor and the OO system exchange information*
b. class operation is invoked
c. messages are passed between objects
d. all of the above

23. The state diagram
a. depicts relationships between data objects
b. depicts functions that transform the data flow
c. indicates how data are transformed by the system
*d. indicates system reactions to external events*

24. For purposes of behavior modeling a state is any
a. consumer or producer of data.
b. data object hierarchy.
*c. observable mode of behavior.*
d. well defined process.

# Chapter 9 Multiple Choice Quiz

1. Which of the following are areas of concern in the design model?
a. architecture
b. data
c. interfaces
d. project scope
*e. a, b and c*

2. The importance of software design can be summarized in a single word
a. accuracy
b. complexity
c. efficiency
*d. quality*

3. Which of these are characteristics of a good design?
a. exhibits strong coupling between its modules
b. implements all requirements in the analysis model

c. includes test cases for all components
d. provides a complete picture of the software]
*e. both b and d*

4. Which of the following is not a characteristic common to all design methods?
*a. configuration management*
b. functional component
c. notation quality assessment
d. guidelines refinement heuristics

5. Software design is an iterative generic process that may be applied without modification to any software project.
a. True
*b. False*

6. What types of abstraction are used in software design?
a. control
b. data
c. environmental
d. procedural
*e. a, b and d*

7. Which of the following models can be used to represent the architectural design of a piece of software.
a. Dynamic models
b. Functional models
c. Structural models
*d. All of the above*

8. Design patterns are not applicable to the design of object-oriented software?
a. True
*b. False*

9. Since modularity is an important design goal it is not possible to have too many modules in a proposed design.
a. True
*b. False*

10. Information hiding makes program maintenance easier by hiding data and procedure from unaffected parts of the program.
*a) True*
b) False

11. Cohesion is a qualitative indication of the degree to which a module

a. can be written more compactly.
*b. focuses on just one thing.*
c. is able to complete its function in a timely manner.
d. is connected to other modules and the outside world.

12. Coupling is a qualitative indication of the degree to which a module
    a. can be written more compactly.
    b. focuses on just one thing.
    c. is able to complete its function in a timely manner.
    *d. is connected to other modules and the outside world.*

13. When using structured design methodologies the process of stepwise refinement is unnecessary.
    a. True
    *b. False*

14. Software designs are refactored to allow the creation of software that is easier to integrate, easier to test, and easier to maintain.
    *a. True*
    b. False

15. Inheritance provides a mechanism by which changes to lower level classes can be propagated to all super classes quickly.
    a. True
    *b. False*

16. Polymorphism reduces the effort required to extend an object system by
    a. coupling objects together more tightly.
    *b. enabling a number of different operations to share the same name.*
    c. making objects more dependent on one another.
    d. removing the barriers imposed by encapsulation.

17. Which of the following is not one of the five design class types
    a. Business domain classes
    *b. Entity classes*
    c. Process classes
    d. User interface classes

18. Which design model elements are used to depict a model of information represented from the user's view?
    a. Architectural design elements
    b. Component-level design elements
    *c. Data design elements*
    d. Interface design elements

19. Which design is analogous to the floor plan of a house?
    *a. Architectural design*
    b. Component-level design
    c. Data design
    d. Interface design

20. Which design model is analogous to the detailed drawings of the access points and external utilities for a house?
    a. Architectural design
    b. Component-level design
    c. Data design
    *d. Interface design*

21. Which design model is analogous to a set of detailed drawings for each room in a house?
    a. Architectural design
    *b. Component-level design*
    c. Data design
    d. Interface design

22. The deployment design elements specify the build order for the software components.
    a. True
    *b. False*

23. One of the key problems in software reuse is the inability to find existing reusable design patterns when hundreds of candidates exist.
    *a. True*
    b. False

24. Design patterns are best thought of as coding patterns.
    a. True
    *b. False*

25. Frameworks and design patterns are the same thing as far as designers are concerned.
    a. True
    *b. False*

# Chapter10 Multiple Choice Quiz

1. The best representation of system architecture is an operational software prototype.
    a. True

*b. False*

2. The architectural representations can be an enabler for communication among project stakeholders.
   *a. True*
   b. False

3. Which of these characteristics are true of a data warehouse, but not a typical data base?
   a. business level orientation
   b. currency of information
   c. integration
   d. nonvolatility
   *e. both c and d*

4. Data design actually begins during the creation of the analysis model, not the architectural model.
   *a. True*
   b. False

5. An architectural style encompasses which of the following elements?
   a. constraints
   b. set of components
   c. semantic models
   d. syntactic models
   *e. a, b and c*

6. To determine the architectural style or combination of styles that best fits the proposed system, requirements engineering is used to uncover
   a. algorithmic complexity
   *b. characteristics and constraints*
   c. control and data
   d. design patterns

7. Before an architectural pattern can be chosen for use in a specific system it must have a code implementation to facilitate its reuse.
   a. True
   *b. False*

8. The criteria used to assess the quality of an architectural design should be based on system
   a. accessibility
   b. control
   c. data

d. implementation
*e. both b and c*

9. During the process of modeling the system in context, systems that interact with the target system are represented as
   a. Peer-level systems
   b. Subordinate systems
   c. Superordinate systems
   d. Working systems
   *e. a, b and c*

10. Once selected, archetypes always need to be refined further as architectural design proceeds.
   *a. True*
   b. False

11. Which of the following is not an example of infrastructure components that may need to be integrated into the software architecture?
   a. Communications components
   b. Database components
   *c. Interface components*
   d. Memory management components

12. In the architecture trade-off analysis method the architectural style should be described using the
   a. data flow view
   b. module view
   c. process view
   d. user view
   *e. a, b and c*

13. Quantitative methods(量化方法) for assessing the quality of proposed architectural designs are readily available.
   a. True
   *b. False*

14. A useful technique for evaluating the overall complexity of a proposed architecture is to look at the component
   a. cohesion flow
   b. dependencies
   c. sharing dependencies
   d. size
   *e. both b and c*

15. When the overall flow in a segment of a data flow diagram is largely sequential and follows straight-line paths, _____ is present.
    a. low coupling
    b. good modularity
    c. transaction flow
    *d. transform flow*

16. When a single item that triggers other data flow along one of many paths of a data flow diagram, _____ characterizes the information flow.
    a. high coupling
    b. poor modularity
    *c. transaction flow*
    d. transform flow

17. When you encounter both transform flow and transaction flow in the same DFD the flow is partitioned and the appropriate mapping technique is used on each part of the DFD.
    *a. True*
    b. False

18. In transaction mapping the first level factoring results in the
    a. creation of a CFD
    *b. derivation of the control hierarchy*
    c. distribution of worker modules
    d. refinement of the module view

19. A successful application of transform or transaction mapping to create an architectural design is supplemented by
    a. entity relationship diagrams
    b. module interface descriptions
    c. processing narratives for each module
    d. test cases for each module
    *e. both b and c*

# Chapter11 Multiple Choice Quiz

1. In the most general sense a component is a modular building block for computer software.
    *a. True*
    b. False

2. In the context of object-oriented software engineering a component contains
    a. attributes and operations
    b. instances of each class
    c. roles for each actor (device or user)

*d. a set of collaborating classes*

3. In traditional software engineering, modules must serve in which of the following roles?
   a. Control component
   b. Infrastructure component
   c. Problem domain component
   *d. All of the above*

4. Software engineers always need to create components from scratch in order to meet customer expectations fully.
   a. True
   *b. False*

5. Which of the following is not one of the four principles used to guide component-level design?
   a. Dependency Inversion Principle
   b. Interface Segregation Principle
   c. Open-Closed Principle
   *d. Parsimonious Complexity Principle*

6. During component-level design it is customary to ignore organization issues like subsystem membership or packaging.
   a. True
   *b. False*

7. The use of stereotypes can help identify the nature of components at the detailed design level.
   *a. True*
   b. False

8. Classes and components that exhibit functional, layer, or communicational cohesion are relatively easy to implement, test, and maintain.
   *a. True*
   b. False

9. Software coupling is a sign of poor architectural design and can always be avoided in every system.
   a. True
   *b. False*

10. In component design, elaboration requires which of the following elements to be described in detail?
    a. Source code

b. Attributes
c. Interfaces
d. Operations
*e. b, c and d*

11. In component-level design "persistent data sources" refer to
    a. Component libraries
    b. Databases
    c. Files
    d. All of the above
    *e. both b and c*

12. The object constraint language (OCL) complements UML by allowing a software engineer to use a formal grammar to construct unambiguous statements about design model elements.
    *a. True*
    b. False

13. OCL is not strong enough to be used to describe pre- or post conditions for design actions.
    a. True
    *b. False*

14. Which of these constructs is used in structured programming?
    a. branching
    b. condition
    c. repetition
    d. sequence
    *e. b, c, and d*

15. Which of these is a graphical notation for depicting procedural detail?
    a. process diagram
    b. decision table
    c. ER diagram
    *d. flowchart*

16. A decision table should be used
    a. to document all conditional statements
    b. to guide the development of the project management plan
    c. only when building an expert system
    *d. when a complex set of conditions and actions appears in a component*

17. A program design language (PDL) is often a
    *a. combination of programming constructs and narrative text*

b. legitimate programming language in its own right
c. machine readable software development language
d. useful way to represent software architecture

18. Which of these criteria are useful in assessing the effectiveness of a particular design notation?
    a. maintainability
    b. modularity
    c. simplicity
    d. size
    *e. a, b and c*

# Chapter12 Multiple Choice Quiz

1. Which of the following interface design principles does not allow the user to remain in control of the interaction with a computer?
    a. allow interaction to interruptible
    b. allow interaction to be undoable
    c. hide technical internals from casual users
    *d. only provide one defined method for accomplishing a task*

2. Which of the following interface design principles reduces the user's memory load?
    a. define intuitive shortcuts
    b. disclose information in a progressive fashion
    c. establish meaningful defaults
    d. provide an on-line tutorial
    *e. answers a, b and c*

3. The reason for reducing the user's memory load is make his or her interaction with the computer quicker to complete.
    a. True
    *b. False*

4. Interface consistency implies that
    a. each application should have its own distinctive look and feel
    b. input mechanisms remain the same throughout the application
    c. navigational methods are context sensitive
    d. visual information is organized according to a design standard
    *e. both b and d*

5. If past interactive models have created certain user expectations it is not generally good to make changes to the model.
    *a. True*
    b. False

6. Which model depicts the profile of the end users of a computer system?
   a. design model
   b. implementation model
   *c. user model*
   d. user's model

7. Which model depicts the image of a system that an end user creates in his or her head?
   a. design model
   b. user model
   c. system model
   *d. system perception*

8. Which model depicts the look and feel of the user interface along with all supporting information?
   *a. Implementation model*
   b. user model
   c. user's model
   d. system perception

9. Which of these framework activities is not normally associated with the user interface design processes?
   *a. cost estimation*
   b. interface construction
   c. interface validation
   d. user and task analysis

10. Which approach(es) to user task analysis can be useful inuser interface design?
    a. have users indicate their preferences on questionnaires
    b. rely on the judgement of experienced programmers
    c. study existing computer-based solutions
    d. observe users performing tasks manually
    *e. both c and d*

11. Object-oriented analysis techniques can be used to identify and refine user task objects and actions without any need to refer to the user voice.
    a. True
    *b. False*

12. The computer's display capabilities are the primary determinant of the order in which user interface design activities are completed.
    a. True
    *b. False*

13. It is sometimes possible that the interface designer is constrained by environmental factors that mitigate against ease of use for many users.
    *a. True*
    b. False

14. One means of defining user interface objects and actions is to conduct a grammatical parse of the user scenario.
    *a. True*
    b. False

15. Interface design patterns typically include a complete component-level design (design classes, attributes, operations, and interfaces).
    *a. True*
    b. False

16. Several common design issues surface for almost every user interface including
    a. adaptive user profiles
    b. error handling resolution of graphics
    c. displays system
    d. response time
    *e. both b and d*

17. Add-on help facilities are almost always better received by users than integrated help facilities.
    a. True
    *b. False*

18. User interface development systems typically provide several mechanisms for building interface prototypes including
    a. code generation
    b. drawing tools
    c. input validation
    d. windows handlers
    *e. both c and d*

19. Usability questionnaires are most meaningful to the interface designers when completed by
    a. customers
    b. experienced programmers
    *c. product users*
    d. project managers

20. Several usability measures can be collected while observing users interacting

with a computer system including

a. down time for the application
b. number of user errors
c. software reliability
d. time spent looking at help materials
*e. both b and d*

# Chapter13 Multiple Choice Quiz

1. In software quality assurance work there is no difference between software verification and software validation.
   a. True
   *b. False*

2. The best reason for using Independent software test teams is that
   a. software developers do not need to do any testing
   *b. a test team will test the software more thoroughly*
   c. testers do not get involved with the project until testing begins
   d. arguments between developers and testers are reduced

3. What is the normal order of activities in which traditional software testing is organized?a. integration testingb. system testingc. unit testingd.validation testing
   a. a, d, c, b
   b. b, d, a, c
   *c. c, a, d, b*
   d. d, b, c, a

4. Class testing of object-oriented software is equivalent to unit testing for traditional software.
   *a. True*
   b. False

5. By collecting software metrics and making use of existing software reliability models it is possible to develop meaningful guidelines for determining when software testing is finished.
   *a. True*
   b. False

6. Which of the following strategic issues needs to be addressed in a successful software testing process?
   a. conduct formal technical reviews prior to testing
   b. specify requirements in a quantifiable manner
   c. use independent test teams
   d. wait till code is written prior to writing the test plan
   *e. both a and b*

7. Which of the following need to be assessed during unit testing?
    a. algorithmic performance
    b. code stability
    c. error handling
    d. execution paths
    *e. both c and d*

8. Drivers and stubs are not needed for unit testing because the modules are tested independently of one another.
    a. True
    *b. False*

9. Top-down integration testing has as it's major advantage(s) that
    a. low level modules never need testing
    b. major decision points are tested early
    c. no drivers need to be written
    d. no stubs need to be written
    *e. both b and c*

10. Bottom-up integration testing has as it's major advantage(s) that
    a. major decision points are tested early
    b. no drivers need to be written
    *c. no stubs need to be written*
    d. regression testing is not required

11. Regression testing should be a normal part of integration testing because as a new module is added to the system new
    a. control logic is invoked
    b. data flow paths are established
    c. drivers require testing
    d. all of the above
    *e. both a and b*

12. Smoke testing might best be described as
    a. bulletproofing shrink-wrapped software
    *b. rolling integration testing*
    c. testing that hides implementation errors
    d. unit testing for small programs

13. When testing object-oriented software it is important to test each class operation separately as part of the unit testing process.
    a. True
    *b. False*

14. The OO testing integration strategy involves testing
    *a. groups of classes that collaborate or communicate in some way*
    b. single operations as they are added to the evolving class implementation
    c. operator programs derived from use-case scenarios
    d. none of the above

15. The focus of validation testing is to uncover places that a user will be able to observe failure of the software to conform to its requirements.
    *a. True*
    b. False

16. Software validation is achieved through a series of tests performed by the user once the software is deployed in his or her work environment.
    a. True
    *b. False*

17. Configuration reviews are not needed if regression testing has been rigorously applied during software integration.
    a. True
    *b. False*

18. Acceptance tests are normally conducted by the
    a. developer
    *b. end users*
    c. test team
    d. systems engineers

19. Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that software is able to continue execution without interruption.
    a. True
    *b. False*

20. Security testing attempts to verify that protection mechanisms built into a system protect it from improper penetration.
    *a. True*
    b. False

21. Stress testing examines the pressures placed on the user during system use in extreme environments.
    a. True
    *b. False*

22. Performance testing is only important for real-time or embedded systems.
    a. True
    *b. False*

23. Debugging is not testing, but always occurs as a consequence of testing.
    *a. True*
    b. False

# Chapter14 Multiple Choice Quiz

1. With thorough testing it is possible to remove all defects from a program prior to delivery to the customer.
    a. True
    *b. False*

2. Which of the following are characteristics of testable software?
    a. observability
    b. simplicity
    c. stability
    *d. all of the above*

3. The testing technique that requires devising test cases to demonstrate that each program function is operational is called
    *a. black-box testing*
    b. glass-box testing
    c. grey-box testing
    d. white-box testing

4. The testing technique that requires devising test cases to exercise the internal logic of a software module is called
    a. behavioral testing
    b. black-box testing
    c. grey-box testing
    *d. white-box testing*

5. What types of errors are missed by black-box testing and can be uncovered by white-box testing?
    a. behavioral errors
    b. logic errors
    c. performance errors
    d. typographical errors
    *e. both b and d*

6. Program flow graphs are identical to program flowcharts.
    a. True

*b. False*

7. The cyclomatic complexity metric provides the designer with information regarding the number of
   a. cycles in the program
   *b. errors in the program*
   c. independent logic paths in the program
   d. statements in the program

8. The cyclomatic complexity of a program can be computed directly from a PDL representation of an algorithm without drawing a program flow graph.
   *a. True*
   b. False

9. Condition testing is a control structure testing technique where the criteria used to design test cases is that they
   a. rely on basis path testing
   *b. exercise the logical conditions in a program module*
   c. select test paths based on the locations and uses of variables
   d. focus on testing the validity of loop constructs

10. Data flow testing is a control structure testing technique where the criteria used to design test cases is that they
    a. rely on basis path testing
    b. exercise the logical conditions in a program module
    *c. select test paths based on the locations and uses of variables*
    d. focus on testing the validity of loop constructs

11. Loop testing is a control structure testing technique where the criteria used to design test cases is that they
    a. rely basis path testing
    b. exercise the logical conditions in a program module
    c. select test paths based on the locations and uses of variables
    *d. focus on testing the validity of loop constructs*

12. Black-box testing attempts to find errors in which of the following categories
    a. incorrect or missing functions
    b. interface errors
    c. performance errors
    *d. all of the above*
    e. none of the above

13. Graph-based testing methods can only be used for object-oriented systems
    a. True

*b. False*

14. Equivalence testing divides the input domain into classes of data from which test cases can be derived to reduce the total number of test cases that must be developed.
    *a. True*
    b. False

15. Boundary value analysis can only be used to do white-box testing.
    a. True
    *b. False*

16. Comparison testing is typically done to test two competing products as part of customer market analysis prior to product release.
    a. True
    *b. False*

17. Orthogonal array testing enables the test designer to maximize the coverage of the test cases devised for relatively small input domains.
    *a. True*
    b. False

18. Test case design "in the small" for OO software is driven by the algorithmic detail of the individual operations.
    *a. True*
    b. False

19. Encapsulation of attributes and operations inside objects makes it easy to obtain object state information during testing.
    a. True
    *b. False*

20. Use-cases can provide useful input into the design of black-box and state-based tests of OO software.
    *a. True*
    b. False

21. Fault-based testing is best reserved for
    a. conventional software testing
    *b. operations and classes that are critical or suspect*
    c. use-case validation
    d. white-box testing of operator algorithms

22. Testing OO class operations is made more difficult by

a. encapsulation
b. inheritance
c. polymorphism
*d. both b and c*

23. Scenario-based testing
*a. concentrates on actor and software interaction*
b. misses errors in specifications
c. misses errors in subsystem interactions
d. both a and b

24. Deep structure testing is not designed to
a. examine object behaviors
b. exercise communication mechanisms
c. exercise object dependencies
*d. exercise structure observable by the user*

25. Random order tests are conducted to exercise different class instance life histories.
*a. True*
b. False

26. Which of these techniques is not useful for partition testing at the class level
a. attribute-based partitioning
b. category-based partitioning
*c. equivalence class partitioning*
d. state-based partitioning

27. Multiple class testing is too complex to be tested using random test cases.
a. True
*b. False*

28. Tests derived from behavioral class models should be based on the
a. data flow diagram
b. object-relation diagram
*c. state diagram*
d. use-case diagram

29. Client/server architectures cannot be properly tested because network load is highly variable.
a. True
*b. False*

30. Real-time applications add a new and potentially difficult element to the testing

mix

a. performance
b. reliability
c. security
*d. time*

# Chapter16 Multiple Choice Quiz

1. Which of the following is not a characteristic of a WebApp?
   a. content driven
   b. continuously evolving
   *c. easily measurable*
   d. network intensive

2. Which of these application categories are commonly encountered in WebE work?
   a. informational
   b. transaction-oriented
   c. portal
   *d. all of the above*

3. WebApps must be developed and deployed quickly, making the application of software engineering processes impossible.
   a. True
   *b. False*

4. Which process model best describes WebE?
   a. Linear model
   *b. Incremental model*
   c. Formal model
   d. all of the above

5. The mechanics of software engineering analysis, design, and testing must be adapted to accommodate the special characteristics of WebApps.
   *a. True*
   b. False

6. Which of the following technologies is important to Web engineers?
   a. component-based development
   b. internet standards
   c. security
   *d. all of the above*

7. An evolutionary process model would never be chosen over an agile process model to build a WebApp.

a. True
*b. False*

8. Which of the following is not one of the characteristics that we need to take into account when a process framework for WebE is formulated.
   a. Changes occur frequently
   *b. Graphic design expertise is hard to acquire*
   c. Timelines are short
   d. WebApps are delivered incrementally

9. During the analysis/formulation step of the WebE process two types of goals need to be defined
   a. applicative goals and aesthetic goals
   *b. applicative goals and informational goals*
   c. information goals and performance goals
   d. aesthetic goals and performance goals

10. With extremely short time-lines it is impossible to develop plans for WebApp development projects.
    a. True
    *b. False*

11. Which activities are conducted during the WebE modeling process?
    a. content analysis
    b. refine user tasks
    c. design architecture
    *d. all of the above*

12. Which test(s) are not performed during WebE construction?
    a. configuration
    b. navigation
    *c. reliability*
    d. usability

13. WebE are usually delivered to users untested and then debugged as user complaints are registered.
    a. True
    *b. False*

14. Since WebApps are fairly standard it is not important for developers to understand the customer's business needs and objectives.
    a. True
    *b. False*

15. Scenario-based approaches to describing user interaction are good to use in WebE.
    *a. True*
    b. False

16. Since WebApps are usually developed using agile processes, modeling can safely be ignored or skipped altogether.
    a. True
    *b. False*

# Chapter18 Multiple Choice Quiz

1. Which of the following is not one of the WebApp requirements analysis tasks?
    a. Analysis modeling
    b. Formulation
    c. Requirements gathering
    *d. User interface prototyping*

2. User hierarchies are used to replace UML user representations for WebApps having large numbers of user categories?
    a. True
    *b. False*

3. WebApp use-cases might be described as bundles of functionality.
    a. True
    *b. False*

4. As use-cases are organized into functional packages, each functional package is assessed to ensure that it is
    a. Comprehensive
    b. Highly cohesive
    c. Loosely coupled
    *d. All of the above*

5. Dynamic elements of WebApp analysis models describe how users interaction with the system.
    a. True
    *b. False*

6. Which is not one of the analysis activities that is used to create a complete analysis model?
    a. Configuration analysis
    b. Content analysis
    c. Functional analysis
    *d. Data analysis*

7. The content model contains dynamic elements that encompass the WebApp content objects.
   a. True
   *b. False*

8. Content objects are extracted from use cases by examining the scenarios description for direct or indirect content references.
   *a. True*
   b. False

9. In building a content hierarchy is sufficient to examine a list of content objects and a brief description of each object.
   a. True
   *b. False*

10. By examining each use-case and building a class model for 1 or 2 representative users it is possible to derive the needed analysis classes.
    a. True
    *b. False*

11. What are the most useful UML diagrams and related information that can be used to represent a WebApp interaction model?
    a. activity diagrams, class diagrams, state diagrams, interface prototype
    b. activity diagrams, collaboration diagrams, sequence diagrams, state diagrams
    *c. use-cases, sequence diagrams, state diagrams, interface prototype*
    d. use-cases, sequence diagrams, state diagrams, sequence diagrams

12. A user interface prototype should not be created during WebApp analysis because doing so involves programming.
    a. True
    *b. False*

13. UML activity diagrams can be used to represent the user observable functionality delivered by the WebApp as well as the operations contained in each analysis class.
    *a. True*
    b. False

14. The construction details indicating how the user will invoke an operation are deferred until the WebApp design phase.
    *a. True*
    b. False

15. UML deployment diagrams can be used to create the configuration model for a complex WebApp.
    *a. True*
    b. False

16. Configuration analysis focuses on the architecture of the user's Web browsing environment.
    a. True
    *b. False*

17. Which of these are not steps of relationship-navigation analysis?
    a. Element analysis
    b. Evaluation analysis
    *c. Functional analysis*
    d. Stakeholder analysis

18. The answers to the relationship analysis questions help the Web engineer position a content element within the WebApp.
    *a. True*
    b. False

19. Once the WebApp architecture is modeled the Web engineer must consider requirements that dictate how users will navigate from one content element to another.
    a. True
    *b. False*

# Chapter19 Multiple Choice Quiz

1. Which of the following characteristics should not be used to assess the quality of a WebApp?
    *a. aesthetics*
    b. reliability
    c. maintainability
    d. usability

2. Which of the following are design goals for every WebApp?
    a. Simplicity
    b. Consistency
    c. Navigability
    d. Visual appeal
    *e. all of the above*

3. Which of the following are not part of the design pyramid for WebE design?
    a. Architectural design

b. *Business case design*
c. Content design
d. Navigation design

4. Every WebApp user interface should be easy to use, easy to navigate, error-free and functional.
   a. *True*
   b. False

5. With WebApps content is everything, a poorly defined user interface will be quickly overlooked by frequent users.
   a. True
   b. *False*

6. Which of these are WebApp interaction mechanisms?
   a. Graphic icons
   b. Graphic images
   c. Navigation menus
   d. *All of the above*

7. UML does not have any representation schemas that are useful in building WebApp design models.
   a. True
   b. *False*

8. Screen layout design has several widely accepted standards based on human factors research.
   a. True
   b. *False*

9. Graphic design considers every aspect of the look and feel of a WebApp.
   a. *True*
   b. False

10. Content design is conducted by
    a. Copywriters and graphic designer
    b. Web engineers
    c. *both a and b*
    d. none of the above

11. Content objects have both information attributes defined during analysis and implementation specific attributes specified during design.
    a. *True*
    b. False

12. Content objects are not normally chunked into Web pages until the implementation activities begin.
    a. True
    *b. False*

13. Which of the following is not one of the browsing primitives normally found in WebApp interfaces.
    a. Conditional browsing
    b. Nested browsing
    *c. Recursive browsing*
    d. Sequential browsing

14. Content architecture and WebApp architecture are pretty much the same thing for many WebApps?
    a. True
    *b. False*

15. Which of the following isnot one of the content architectural structures used by web engineers?
    a. linear
    b. grid
    c. hierarchical
    *d. parallel*

16. MVC is a three layer architecture that contains a
    a. machine, view, content objects
    b. model, view, and content objects
    *c. model, view, and controller*
    d. machine, view, controller

17. Web navigational design involves creating a semantic navigational unit for each goal associated with each defined user role.
    *a. True*
    b. False

18. To allow the user to feel in control of a WebApp, it is a good idea to mix both horizontal and vertical navigation mechanisms on the same page.
    a. True
    *b. False*

19. Component level design for WebApps is very similar to component level design for other software delivery environments.
    *a. True*

b. False

20. Which of the following is a navigation pattern used during web-based design?
    a. cycle
    b. counterpoint
    c. sieve
    *d. all of the above*

21. Which of these is not one of the design activities associated with object-oriented hypermedia design?
    a. abstract interface design
    b. conceptual design
    *c. content design*
    d. navigational design

22. Most WebApps can be easily characterized by judicious use of widely recognized suites of software metrics?
    a. True
    *b. False*