Brookshear-*Computer Science: An Overview*, 9th edition

Test Bank—Chapter Seven (Software Engineering)

Multiple Choice Questions

1. Which of the following software engineering methodologies is the most rigid?

A. Incremental model B. Waterfall model
C. Extreme programming D. Evolutionary prototyping

ANSWER: B

2. Which of the following is a notational system for representing object-oriented designs?

A. UML B. Structure charts C. Modular designs D. Dataflow diagrams

ANSWER: A

3. Which of the following is an attempt to construct software from off-the-shelf components as is done in other engineering fields?

A. Extreme programming B. Evolutionary prototyping
C. Component architecture D. Open-source development

ANSWER: C

4. Which of the following is most likely an example of a one-to-one relationship?

A. Subscribers and magazines B. Birth dates and people
C. Planets and their moonsD. Dinner guests and table settings

ANSWER: D

5. Which of the following is most likely an example of a many-to-many relationship?

A. Subscribers and magazines B. Birth dates and people
C. Planets and their moonsD. Dinner guests and table settings

ANSWER: A

6. Which of the following is not a feature of UML?

A. Case use diagrams B. Class diagrams
C. Dataflow diagrams D. Collaboration diagrams

ANSWER: C

7. The use of design patterns in software engineering was adopted from what other field?

A. Business administration B. Architecture
C. Biology D. Chemistry

ANSWER: B

8. Which of the following is a form of glass-box testing?

A. basis path testing B. Boundary value analysis C. Beta testing

ANSWER: A

9. Which of the following is a means of controlling the complexity of a software system?

A. CRC cards B. Modularity C. Specifications D. Beta testing

ANSWER: B

10. Which of the following is a way of testing the design of a software system?

A. Entity-relationship diagram B. Class diagram
C. Structure chart D. Structured walkthrough

ANSWER: D

11. Which of the following is not related to the others?

A. Structure Chart B. Imperative paradigm
C. Class diagram D. Procedure

ANSWER: C

12. Which of the following is the method proposed by UML for representing sequences of communication between objects?

A. Class diagram B. Use case diagram
C. Collaboration diagram D. Generalization

ANSWER: C

13. Which of the following is not represented in a class diagram?

A. Generalizations B. The methods within a class
C. The attributes within a class D. The number of instances each class will have

ANSWER: D

14. Which of the following is least related to the Pareto principle?

A. When it rains, it pours.
B. Birds of a feather flock together.
C. Better late than never.

ANSWER: C

15. The Pareto principle is traditionally applied during which phase of software development?

A. Analysis B. Design C. Implementation D. Testing

ANSWER: D

16. Which of the following is the oldest approach to software development?

A. Component architecture
B. Waterfall model
C. Open-source development
D. Extreme programming

ANSWER: B

17. Which of the following is not a tool for designing modular systems?

A. Structure charts
B. Data dictionaries
C. Class diagrams
D. Collaboration diagrams

ANSWER: B

18. Which of the following is a stronger form of cohesion?

A. Functional cohesion
B. Logical cohesion

ANSWER: A

19. Which of the following appears to be the most functionally cohesive?

A. A module that handles all of a customers banking needs
B. A module that handles only transactions related to checking accounts
C. A module that only records deposits to checking accounts
D. A module that collects data for monthly statements

ANSWER: C

20. If a class diagram indicates a one-to-one relationship between class X and class Y, then

A. there will be only one object in the system of "type" X.
B. each object of "type" X will be associated with only one object of "type" Y.
C. there will be exactly one object of "type" X and exactly one object of "type" Y.
D. an object of "type" Y cannot occur without first constructing an object of "type" X.

ANSWER: B

21. Copyright laws were established

A. to allow authors to distribute their work while maintaining certain ownership rights.
B. to allow authors to maintain ownership of their ideas.
C. to restrict access to publications to certain groups within society.
D. to allow ideas to be traced back to their origins.

ANSWER: A

Fill-in-the-blank/Short-answer Questions

1. Identify the stage of software development in which each of the following activities is performed.

A. _____ Programming is conducted.

B. _____ Class diagrams are drawn.

C. _____ User needs are analyzed.

ANSWER: A. Implementation B. Design C. Analysis

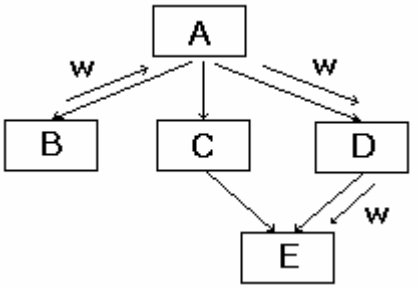
2. During the analysis stage of software development, user needs are identified in the form of non-technical _____ that are then converted into technical _____ .

ANSWER: requirements, specifications

3. Prototyping occurs in two forms. In one, called _____ prototyping the original prototype is slowly enhanced to become the final product. In the other, called _____ prototyping, the original prototype is used as an "experimental" system that is ultimately discarded.

ANSWER: Evolutionary, throwaway

4. Answer the following questions in terms of the structure chart below.



A. What modules directly use the services of module E?

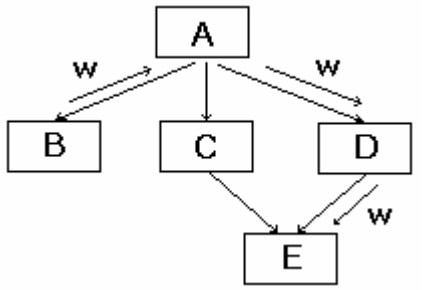
_____

B. The services of which modules are directly used by module A?

_____

ANSWER: A. C and D B. B, C, and D

5. Based on the structure chart below, in which module does the data item w originate?



_____

ANSWER: B

6. In an object-oriented design using UML, _____ diagrams are used to represent classes and their basic relationships, whereas _____ diagrams are used to represent communication between objects.

ANSWER: Class, collaboration

7. In each of the following cases, indicate whether the information would be represented within a case use diagram, a class diagram, or a collaboration diagram.

A. _____ The methods within a class

B. _____ The ways in which the system will interact with its environment

C. _____ The manner in which its internal objects will interact

D. _____ Relationships among classes

ANSWER: A. Class diagram B. Use case diagram C. Collaboration diagram D. Class diagram

8. _____ is a notational system for representing object-oriented designs. It includes standards for representing _____ diagrams that show how users interact with the proposed system as well as _____diagrams that show how objects within the proposed system will interact.

ANSWER: UML, case use, collaboration

9. Give an example of a one-to-many relationship.

_____

ANSWER: Answers may vary. Examples include: classrooms to chairs (a classroom has many chairs but each chair is in only one classroom), mothers to children (a mother may have many children but each child has only one mother), and many others.

10. In each case below indicate whether the activity relates to a structure chart or a class diagram.

A. _____ Identifying actions to be performed

B. _____ Identifying the types of objects in a system

C. _____ Identifying relationships between “types” of objects

D. _____ Identifying how activities performed by different procedures relate to one another

ANSWER: A. Structure chart B. Class diagram C. Class diagram D. Structure chart

11. In each case below indicate whether the activity relates to a collaboration diagram or a dataflow diagram.

A. _____ Identifying messages passed between objects

B. _____ Identifying how data items are combined to produce new items

C. _____ Identifying relationships between objects

D. _____ Identifying how information and leaves a system

ANSWER: A. Collaboration diagram B. Dataflow diagram C. Collaboration diagram D. Dataflow diagram

12. In each case below indicate whether the phrase relates to coupling or cohesion.

A. _____ The interaction between modules

B. _____ Passing data from one module to another

C. _____ Ensuring that a module performs a unique task in its entirety

ANSWER: A. Coupling B. Coupling C. Cohesion

13. Identify two forms of inter-module coupling.

_____

_____

ANSWER: Data coupling and control coupling

14. In each case below indicate whether the activity is a form of glass-box testing or black-box testing.

A. _____ Basis path testing

B. _____ Boundary value analysis

C. _____ Beta testing

ANSWER: A. Glass-box testing B. Black-box testing C. Black-box testing

15. In each case below indicate whether the activity relates to glass-box testing or black-box testing.

A. _____ Testing to see if the system performs in a timely manner

B. _____ Designing test data to ensure that each instruction is executed at least once

C. _____ Testing to see if the software system meets the requirements identified during original analysis

ANSWER: A. Black-box testing B. Glass-box testing C. Black-box testing

16. State the Pareto principle in the context of software engineering.

_____

ANSWER: Errors in a software system tend to be concentrated in relatively small areas.

17. In each case below indicate whether the activity is primarily top-down or bottom-up.

A. _____ Building software from previously constructed components

B. _____ Dividing a module into smaller modules to obtain greater cohesion

C. _____ Designing a dataflow diagram by successively adding more specificity

ANSWER: A. Bottom-up B. Top-down C. Top-down

18. As a general rule, one should strive to _____ (maximize or minimize) coupling between modules and to _____ (maximize or minimize) cohesion within modules.

ANSWER: minimize, maximize

19. Give two examples of recent advances in software engineering.

_____

_____

ANSWER: There are many possible answers (and they vary depending on the interpretation of "recent." Answers include: component architecture, the application of design patterns, open-source development, extreme programming, the use of prototypes, CASE tools, the development of UML, and others.

20. Identify two legal techniques that have been applied to protect a software developer's ownership rights.

_____

_____

ANSWER: Possible answers include copyright law, patent law, and nondisclosure agreements.

Vocabulary (Matching) Questions

The following is a list of terms from the chapter along with descriptive phrases that can be used to produce questions (depending on the topics covered in your course) in which the students are ask to match phrases and terms. An example would be a question of the form, "In the blank next to each phrase, write the term from the following list that is best described by the phrase."

Term	**Descriptive Phrase**
metric	A means of quantifying
software life cycle	Develop, use, modify
waterfall model	An older, rather rigid approach to software development
prototyping	An approach to software development in which partial systems are

	constructed
component architecture	A means of constructing software from prefabricated units
structure chart	A means of representing procedural dependencies
cohesion	The “glue” that holds a module together
collaboration diagram	A diagram representing communication between objects
case use diagram	A diagram representing communication between a system and its users
UML	A standard notational system for representing object-oriented designs
global data	A means of implementing implicit coupling
modularity	A means of managing complexity within a large software system
structured walkthrough	A means of testing a design before it is implemented
beta testing	Allows potential users to experiment with preliminary versions of software
glass-box testing	Confirms that the internal structure of a software system is reliable
open-source development	A somewhat renegade methodology for software development
analysis	The beginning of the software development phase
specifications	System requirements translated into technical context
data dictionary	A central warehouse of information regarding data throughout a system
top-down	General to specific (as opposed to specific to general)
one-to-many	A type of relationship between entities

General Format Questions

1. Identify two distinctions between software engineering and other traditional fields of engineering.

ANSWER: Possible answers include: In contrast to traditional fields of engineering, there is a lack of metrics for measuring quantities in software engineering. Software engineering does not involve tolerances in the sense of traditional engineering. Traditional engineering builds products from off-the-shelf components; this is still a goal in software engineering.

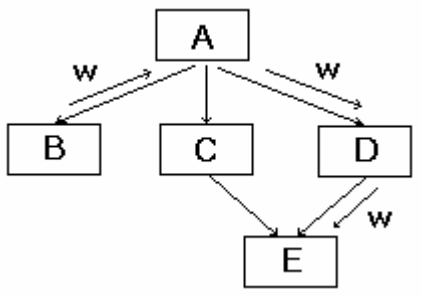
2. In what sense is the software life cycle different from the life cycle of other products?

ANSWER: Software does not wear out so rather than needing maintenance in the traditional sense, software requires modification due to changing environments or detection of errors.

3. Explain the distinction between open-source development and beta testing.

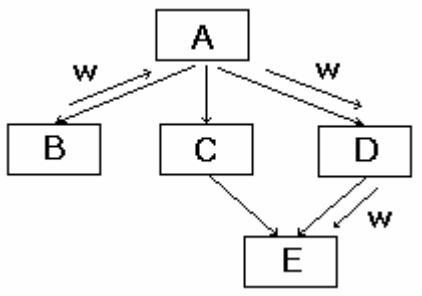
ANSWER: Open-source development involves “testers” to modify software whereas beta testing allows them only to report errors.

4. Describe the data coupling represented by the following structure chart.



ANSWER: The modules B, A, D, and E are coupled via the data item w. B creates w and passes it to A, A passes it to D, and D passes it to E.

5. Describe the control coupling represented by the following structure chart.



ANSWER: Module A can pass control to modules B, C, and D. Each of modules C and D can pass control to E.

6. Describe the process of a structured walkthrough.

ANSWER: A structured walkthrough is a "theatrical" exercise in which people play the roles of various software modules in order to identify flaws in the system's design.

7. In what sense is the object-oriented paradigm ideal for implementing design patterns?

ANSWER: The object-oriented paradigm uses classes as templates for constructing objects. This "template" approach is a natural means of implementing design patterns.

8. Give an argument supporting the statement that modularity is the most important principle in software engineering.

ANSWER: Modularity, which is found in all software engineering paradigms, is the primary means of dealing with complexity.

9. Explain the distinction between structure charts and class diagrams.

ANSWER: The two are used in different design paradigms. Structure charts are used to represent the relationship between procedural modules in an imperative design. Class diagrams are used to represent the relationship between classes in an object-oriented design.

10. Explain some of the ways in which software engineering has benefited from the development of the object-oriented paradigm.

ANSWER: The concept of classes and objects provides an excellent modularizing tool. Moreover, it has provided a means of implementing design patterns so that software can be constructed from prefabricated units.

11. Explain the role of each of the following forms of documentation: user documentation, technical documentation, and system documentation.

ANSWER: User documentation explains how to use a system as an abstract tool. Technical documentation explains how to install a system, how to update the system, and perhaps how to customize the system. System documentation explains the internal construction of the system to support internal modifications.

12. Explain why inheritance may not be the best way of implementing generalizations among classes.

ANSWER: Inheritance introduces a strong coupling between classes that may cease to be valid in later software modifications.