1. What is Software Engineering?
A) Designing websites
B) Applying engineering to software development
C) Installing software
D) Writing code only
✅ **Answer:** B) Applying engineering to software development

---

2. Which of the following is *not* a characteristic of good software?
A) Maintainability
B) Complexity
C) Efficiency
D) Usability
✅ **Answer:** B) Complexity

---

3. Software engineering paradigms include:
A) Code paradigm only
B) Procedural paradigm only
C) Programming, Design, and Process paradigms
D) Hardware paradigm
✅ **Answer:** C) Programming, Design, and Process paradigms

---

4. What does the programming paradigm focus on?
A) Hardware management
B) Writing programs
C) Database queries
D) Documenting software
✅ **Answer:** B) Writing programs

---

5. The design paradigm is mainly concerned with:
A) UI Design
B) Implementation
C) Architecture and system design
D) Security
✅ **Answer:** C) Architecture and system design

🌐 Generic View of Software Engineering

6. What is the first phase of software engineering?

A) Implementation
B) Design
C) Communication
D) Testing

✅ **Answer:** C) Communication

---

7. In the generic view, the final phase is:

A) Deployment
B) Design
C) Testing
D) Maintenance

✅ **Answer:** D) Maintenance

---

8. Which activity includes requirements gathering?

A) Construction
B) Communication
C) Deployment
D) Testing

✅ **Answer:** B) Communication

---

9. Construction phase involves:

A) Requirement analysis
B) Maintenance
C) Coding and testing
D) Planning

✅ **Answer:** C) Coding and testing

---

10. Process framework activities include all except:

A) Communication
B) Planning
C) Compilation
D) Modeling

✅ **Answer:** C) Compilation

---

11. Deployment includes:

A) Building database
B) Delivering software to end-users

C) Planning design
D) Software backup only
✅ **Answer:** B) Delivering software to end-users

---

12. Support activity after deployment is known as:
A) Construction
B) Testing
C) Maintenance
D) Planning
✅ **Answer:** C) Maintenance

---

13. Which of the following is a part of the umbrella activity?
A) Testing
B) Risk management
C) Coding
D) Design
✅ **Answer:** B) Risk management

---

14. Software process framework includes how many generic activities?
A) 2
B) 4
C) 5
D) 6
✅ **Answer:** D) 6

---

15. Which of the following is NOT an umbrella activity in software engineering?
A) Risk management
B) Configuration management
C) Construction
D) Reusability management
✅ **Answer:** C) Construction

🔁 SDLC Models

16. SDLC stands for:
A) Software Device Life Cycle
B) System Development Logic Cycle
C) Software Development Life Cycle

D) Software Design Long Cycle

✅ **Answer:** C) Software Development Life Cycle

---

17. Which SDLC model is the oldest and simplest?

A) Spiral
B) Agile
C) Waterfall
D) V-Model

✅ **Answer:** C) Waterfall

---

18. In Waterfall model, each phase is:

A) Skipped sometimes
B) Iterative
C) Dependent on other
D) Completed before moving to the next

✅ **Answer:** D) Completed before moving to the next

---

19. Which model is best suited for small, clear, and well-defined projects?

A) Spiral
B) Waterfall
C) Agile
D) V-Model

✅ **Answer:** B) Waterfall

---

20. Which model emphasizes risk assessment?

A) Spiral
B) Waterfall
C) Agile
D) Incremental

✅ **Answer:** A) Spiral

---

21. The Spiral model is a combination of:

A) Waterfall and Agile
B) Design and Build
C) Iterative and Risk Analysis
D) Testing and Maintenance

✅ **Answer:** C) Iterative and Risk Analysis

---

22. Agile model focuses on:
A) Risk management
B) Documentation
C) Flexibility and rapid delivery
D) One-time delivery
✅ **Answer:** C) Flexibility and rapid delivery

---

23. Which model uses iterations called sprints?
A) Spiral
B) Agile
C) Waterfall
D) V-Model
✅ **Answer:** B) Agile

---

24. What is the main drawback of the Waterfall model?
A) Too flexible
B) Expensive
C) Not suitable for small projects
D) Inflexibility and late error discovery
✅ **Answer:** D) Inflexibility and late error discovery

---

25. In V-Model, testing activities are:
A) After development
B) Parallel to development
C) Ignored
D) Repeated after deployment
✅ **Answer:** B) Parallel to development

---

26. Agile development usually involves:
A) Only developers
B) Only testers
C) Cross-functional teams
D) Only project managers
✅ **Answer:** C) Cross-functional teams

---

27. Incremental model delivers software:
A) All at once
B) In phases
C) After maintenance
D) After hardware test
✅ **Answer:** B) In phases

---

28. Which model is useful when requirements are unclear?
A) Waterfall
B) Spiral
C) V-Model
D) Agile
✅ **Answer:** D) Agile

---

29. Which SDLC model is considered evolutionary?
A) Waterfall
B) Spiral
C) V-Model
D) Build-and-fix
✅ **Answer:** B) Spiral

---

30. Which is *not* a phase in the Waterfall model?
A) Design
B) Coding
C) Debugging
D) Testing
✅ **Answer:** C) Debugging

---

31. In Agile, customer involvement is:
A) Low
B) Only at the start
C) Continuous
D) Not required
✅ **Answer:** C) Continuous

---

32. Which model promotes prototype development?
A) V-Model
B) Incremental

C) Prototype Model
D) Waterfall
✅ **Answer:** C) Prototype Model

---

33. Prototyping model is useful when:
A) Cost is fixed
B) Time is fixed
C) Requirements are not clear
D) UI is not needed
✅ **Answer:** C) Requirements are not clear

---

34. Agile values __ over processes and tools.
A) Documentation
B) Individuals and interactions
C) Contracts
D) Formal methods
✅ **Answer:** B) Individuals and interactions

---

35. What is time-boxing in Agile?
A) Time-based salary
B) Fixed time for each sprint
C) Time limits for meetings
D) Estimation of testing time
✅ **Answer:** B) Fixed time for each sprint

---

36. In SDLC, maintenance phase involves:
A) Creating a new project
B) Gathering requirements
C) Fixing bugs and updates
D) Testing code only
✅ **Answer:** C) Fixing bugs and updates

---

37. What is the first activity in the SDLC?
A) Design
B) Coding
C) Requirement gathering
D) Testing
✅ **Answer:** C) Requirement gathering

---

38. SDLC model selection depends on:
A) Team size
B) User interface
C) Project size and clarity
D) Compiler used
✅ **Answer:** C) Project size and clarity

---

39. What is the main goal of SDLC?
A) Buy new software
B) Develop software efficiently
C) Test only
D) Install OS
✅ **Answer:** B) Develop software efficiently

---

40. In which model is feedback incorporated at each stage?
A) Waterfall
B) Spiral
C) V-Model
D) Build-and-Fix
✅ **Answer:** B) Spiral

41. Which of the following best describes the *primary distinction* between the Spiral Model and the Incremental Model?
A) Spiral is sequential, Incremental is iterative
B) Spiral focuses on risk analysis, Incremental focuses on staged delivery
C) Spiral is used for maintenance only
D) Incremental is more expensive than Spiral
✅ **Answer:** B) Spiral focuses on risk analysis, Incremental focuses on staged delivery

---

42. Which quality attribute refers to the ease with which software can be modified to correct faults or improve performance?
A) Portability
B) Maintainability
C) Usability
D) Efficiency
✅ **Answer:** B) Maintainability

---

43. In software engineering, what does the term "refactoring" refer to?
A) Testing modules
B) Rewriting requirements
C) Improving internal code structure without changing its behavior
D) Redesigning the user interface
✅ **Answer:** C) Improving internal code structure without changing its behavior

---

44. Which SDLC model is *most suitable* for projects where high-level risks need continuous evaluation?
A) Waterfall Model
B) V-Model
C) Spiral Model
D) Agile Model
✅ **Answer:** C) Spiral Model

---

45. A major disadvantage of the Build-and-Fix model is:
A) High documentation overhead
B) No user involvement
C) No structured development process or maintenance planning
D) Too many iterations
✅ **Answer:** C) No structured development process or maintenance planning

---

46. In software engineering, which of the following is an example of a *non-functional requirement*?
A) The user must be able to log in
B) The system must support 200 concurrent users
C) The application shall allow adding items to cart
D) The form must have a name and email field
✅ **Answer:** B) The system must support 200 concurrent users

---

47. What is the main goal of *configuration management* in software projects?
A) Scheduling team meetings
B) Tracking and controlling changes in the software
C) Measuring code size
D) Refactoring database schema
✅ **Answer:** B) Tracking and controlling changes in the software

---

48. Which of the following is *not* typically considered an SDLC model?
A) Agile
B) Spiral
C) Scrum
D) V-Model
✅ **Answer:** C) Scrum
(*Note: Scrum is a project management framework used within Agile.*)

---

49. Which activity is most likely to detect integration-level defects?
A) Unit Testing
B) Regression Testing
C) Integration Testing
D) Acceptance Testing
✅ **Answer:** C) Integration Testing

---

50. Why is requirements traceability important in software engineering?
A) To trace code coverage
B) To trace bug sources
C) To ensure that all requirements are implemented and tested
D) To improve compiler performance
✅ **Answer:** C) To ensure that all requirements are implemented and tested

1. What does a "statement of system scope" describe?
A) The types of hardware required
B) The boundaries and objectives of the system
C) The code structure of the system
D) The programming language used
✅ **Answer:** B) The boundaries and objectives of the system

---

2. What is the purpose of isolating top-level processes in requirements analysis?
A) To reduce the number of users in the system
B) To define the system's main activities
C) To minimize the number of errors in coding
D) To identify system security measures
✅ **Answer:** B) To define the system's main activities

---

3. What is meant by "refinement and review" in the context of requirements analysis?
A) Writing code for the system
B) Reviewing the system documentation for completeness and correctness
C) Testing the system for bugs
D) Configuring the physical components of the system
✅ **Answer:** B) Reviewing the system documentation for completeness and correctness

---

4. A key objective of analyzing a problem in software engineering is to:
A) Find errors in the source code
B) Identify the requirements and constraints of the system
C) Create a design for the user interface
D) Select a programming language
✅ **Answer:** B) Identify the requirements and constraints of the system

---

5. Which of the following best describes a software specification document?
A) A document containing the system’s code structure
B) A document detailing the hardware setup for the system
C) A document that outlines system requirements and design criteria
D) A document that lists all the test cases for the system
✅ **Answer:** C) A document that outlines system requirements and design criteria

---

6. What is the purpose of reviewing the software specification document for consistency?
A) To ensure there is no duplication in requirements
B) To check the document for spelling mistakes
C) To ensure all team members agree on the requirements
D) To ensure that the design is up to date
✅ **Answer:** A) To ensure there is no duplication in requirements

---

7. In requirements analysis, which of the following ensures the software will meet its intended goals?
A) Correctness
B) Implementation
C) Security
D) Maintenance
✅ **Answer:** A) Correctness

---

8. A software specification document should be reviewed for completeness to ensure that:
A) It includes all necessary system requirements
B) It follows the correct coding conventions
C) All team members agree with the design
D) The code is efficient
✅ **Answer:** A) It includes all necessary system requirements

---

9. During requirements analysis, isolating top-level processes involves:
A) Developing the user interface
B) Defining the system’s functional modules
C) Writing the system code
D) Setting up the database
✅ **Answer:** B) Defining the system’s functional modules

---

10. The main goal of requirements analysis is to:
A) Generate code
B) Create a detailed specification for system development
C) Review the system design
D) Test the system
✅ **Answer:** B) Create a detailed specification for system development

---

11. Which of the following is NOT a part of the requirements analysis phase?
A) Defining the system’s scope
B) Isolating the system’s top-level processes
C) Reviewing the system’s code
D) Creating the software specification document
✅ **Answer:** C) Reviewing the system’s code

---

12. What should be included in a software specification document?
A) Test cases
B) A detailed description of system functionality
C) Code snippets
D) Hardware specifications
✅ **Answer:** B) A detailed description of system functionality

---

13. A software specification document must be reviewed for correctness to:
A) Check if the software works on all operating systems
B) Ensure that it matches the user’s needs and requirements
C) Review the coding style
D) Ensure proper code indentation
✅ **Answer:** B) Ensure that it matches the user’s needs and requirements

---

14. How can a software specification document be tested for completeness?
A) By writing the system code
B) By reviewing if all system requirements are covered
C) By creating unit tests
D) By deploying the system
✅ **Answer:** B) By reviewing if all system requirements are covered

---

15. In requirements analysis, "allocation to physical elements" refers to:
A) Identifying the software architecture
B) Mapping software functions to hardware components
C) Coding the system
D) Testing the system on physical hardware
✅ **Answer:** B) Mapping software functions to hardware components

---

16. The process of reviewing a software specification document for consistency ensures that:
A) The requirements align with user needs

B) All the system’s functional requirements are addressed in the same way
C) The system is able to meet deadlines
D) The team has written the documentation in a structured format
✅ **Answer:** B) All the system’s functional requirements are addressed in the same way

---

17. Which of the following is a common tool used for reviewing a software specification document?
A) Debugger
B) Requirements management software
C) Integrated Development Environment (IDE)
D) Code profiler
✅ **Answer:** B) Requirements management software

---

18. The purpose of creating a software specification document is to:
A) Share the software design with the development team
B) Communicate the requirements to stakeholders
C) Write the actual code
D) Perform unit testing
✅ **Answer:** B) Communicate the requirements to stakeholders

---

19. During requirements analysis, "refinement" refers to:
A) Optimizing the system’s code
B) Adding detail to high-level requirements
C) Refining the user interface design
D) Refactoring existing code
✅ **Answer:** B) Adding detail to high-level requirements

---

20. Which of the following is a benefit of isolating top-level processes in the system?
A) It helps in identifying key stakeholders
B) It reduces the system’s complexity by breaking it into smaller components
C) It simplifies the user interface design
D) It allows for faster coding
✅ **Answer:** B) It reduces the system’s complexity by breaking it into smaller components

---

Advanced-Level MCQs on Requirements Analysis

---

21. During the software specification document review, what should be done to verify the document's correctness?
A) Check if the design meets the functional requirements
B) Review code quality
C) Test the system
D) Verify user interface consistency
✅ **Answer:** A) Check if the design meets the functional requirements

---

22. Which of the following is an advantage of reviewing a software specification document for completeness?
A) Ensuring that all system requirements are clearly defined and achievable
B) Checking for spelling and grammatical errors
C) Making sure the system code is error-free
D) Confirming the software runs on all operating systems
✅ **Answer:** A) Ensuring that all system requirements are clearly defined and achievable

---

23. Which technique is typically used to ensure that the software specification document is correct and consistent?
A) User feedback
B) Formal verification
C) Code review
D) Unit testing
✅ **Answer:** B) Formal verification

---

24. What is the primary purpose of isolating top-level processes during requirements analysis?
A) To design the user interface
B) To create a list of required hardware components
C) To break down the system into manageable processes and functions
D) To finalize the project budget
✅ **Answer:** C) To break down the system into manageable processes and functions

---

25. Which of the following would indicate a software specification document is not complete?
A) Missing detailed system requirements
B) The design is complex and hard to understand
C) No stakeholders have reviewed the document

D) It includes a lot of unnecessary features

✅ **Answer:** A) Missing detailed system requirements

---

26. In the context of software specification, what does the term "allocation to physical elements" refer to?

A) Assigning system functions to physical hardware components

B) Assigning the project budget to different tasks

C) Deciding which programming language to use

D) Allocating software testing to different team members

✅ **Answer:** A) Assigning system functions to physical hardware components

---

27. Which of the following best describes the concept of "refinement" in software requirements analysis?

A) Making vague requirements more detailed and specific

B) Reducing the size of the software system

C) Testing the software code

D) Writing the final software documentation

✅ **Answer:** A) Making vague requirements more detailed and specific

---

28. What is the main objective of reviewing a software specification document for consistency?

A) To ensure all functions and requirements are described in the same way

B) To check if there are any duplicate functions in the system

C) To ensure that the code is written according to the design

D) To verify if the system can be deployed without issues

✅ **Answer:** A) To ensure all functions and requirements are described in the same way

---

29. Which of the following should be avoided when creating a software specification document?

A) Including clear and detailed functional requirements

B) Including vague or ambiguous language

C) Ensuring stakeholder involvement

D) Providing non-functional requirements

✅ **Answer:** B) Including vague or ambiguous language

---

30. The process of refining a software specification document helps in:

A) Improving the system’s performance

B) Adding more features to the system
C) Making the requirements more detailed and clear
D) Removing unnecessary hardware
✅ **Answer:** C) Making the requirements more detailed and clear

31. Which of the following is an example of a functional requirement in software specification?

A) The system must support multi-language functionality
B) The system must have a user-friendly interface
C) The system should allow users to log in with username and password
D) The system must be compatible with mobile devices
✅ **Answer:** C) The system should allow users to log in with username and password

---

32. What is the role of the "statement of system scope" in requirements analysis?

A) It lists the features to be included in the system
B) It describes the system's design architecture
C) It defines the boundaries and objectives of the system
D) It provides the project cost and timeline
✅ **Answer:** C) It defines the boundaries and objectives of the system

---

33. Which technique is commonly used to ensure that the requirements are complete during the analysis phase?

A) Code reviews
B) Requirements walkthroughs
C) Performance testing
D) User acceptance testing
✅ **Answer:** B) Requirements walkthroughs

---

34. What is the purpose of "refining" requirements in the analysis phase?

A) To add more code to the system
B) To make vague or general requirements more detailed and specific
C) To test the system for bugs
D) To choose a programming language
✅ **Answer:** B) To make vague or general requirements more detailed and specific

---

35. What is typically included in a software specification document?

A) A list of the coding standards to be followed
B) A list of potential bugs
C) Detailed descriptions of system functionalities and requirements
D) The source code
✅ **Answer:** C) Detailed descriptions of system functionalities and requirements

---

36. During requirements analysis, the process of ″allocation to physical elements″ refers to:

A) Assigning the system's features to hardware components
B) Writing code for system functionalities
C) Allocating the budget for the project
D) Assigning team members to development tasks
✅ **Answer:** A) Assigning the system's features to hardware components

---

37. Which of the following is an example of a non-functional requirement?

A) The system must process transactions in real-time
B) The system must allow users to log in
C) The system should provide a search feature
D) The system must allow users to place orders
✅ **Answer:** A) The system must process transactions in real-time

---

38. A software specification document should be reviewed for correctness to ensure:

A) The system works on all platforms
B) The document matches the user’s needs and requirements
C) The project deadline is met
D) The source code is error-free
✅ **Answer:** B) The document matches the user’s needs and requirements

---

39. In the requirements analysis phase, the primary objective of isolating top-level processes is to:

A) Break down the system into smaller, manageable modules
B) Choose the programming language for the project
C) Begin the coding process
D) Identify the hardware components required for the system
✅ **Answer:** A) Break down the system into smaller, manageable modules

---

40. In a software specification document, which section defines the limits or boundaries of the system?

A) Functional requirements
B) System scope
C) Non-functional requirements
D) Risk assessment
✅ **Answer:** B) System scope

---

41. A key component of reviewing a software specification document for completeness is ensuring that:

A) All system features and functionalities have been captured
B) The system’s code is written correctly
C) The interface design is finalized
D) All stakeholders have signed off on the document
✅ **Answer:** A) All system features and functionalities have been captured

---

42. What is the role of ″input validation″ in the software specification document?

A) To ensure the user interface is consistent
B) To ensure that the input data is accurate and follows the required format
C) To check the system's compatibility with different devices
D) To optimize the system’s performance
✅ **Answer:** B) To ensure that the input data is accurate and follows the required format

---

43. When a software specification document is reviewed for consistency, which of the following is being checked?

A) Whether the system meets the budget requirements
B) Whether the requirements are described in a consistent and coherent manner
C) Whether the source code is clean and efficient
D) Whether all team members understand the requirements
✅ **Answer:** B) Whether the requirements are described in a consistent and coherent manner

---

44. Which of the following is NOT a key step during the requirements analysis phase?

A) Documenting functional and non-functional requirements
B) Reviewing the code for performance optimization
C) Identifying system boundaries and scope
D) Analyzing and validating user requirements
✅ **Answer:** B) Reviewing the code for performance optimization

---

45. The primary goal of reviewing a software specification document for ″correctness″ is to ensure that:

A) The document contains no grammatical errors
B) The system will meet the user’s needs and expectations
C) The system is properly tested for bugs
D) The requirements are clear to the development team
✅ **Answer:** B) The system will meet the user’s needs and expectations

---

46. The process of allocating system functions to physical hardware elements during requirements analysis is known as:

A) Software optimization
B) System architecture design
C) Configuration management
D) Physical allocation
✅ **Answer:** D) Physical allocation

---

47. Which of the following best describes the purpose of reviewing a software specification document for ″completeness″?

A) To check for technical errors in the system code
B) To verify that all features and requirements are included and nothing is missing
C) To ensure that the design is feasible and cost-effective
D) To ensure that the system meets the specified timeline
✅ **Answer:** B) To verify that all features and requirements are included and nothing is missing

---

48. What does the process of ″refining″ a requirement involve?

A) Correcting mistakes in the code
B) Clarifying and adding detail to vague or general requirements
C) Testing the system for defects
D) Creating new system requirements
✅ **Answer:** B) Clarifying and adding detail to vague or general requirements

---

49. A software specification document is considered ″consistent″ when:

A) The requirements are not conflicting with one another
B) It contains no errors or bugs
C) The system meets the user interface design standards
D) All stakeholders agree on the specifications
✅ **Answer:** A) The requirements are not conflicting with one another

---

50. Which of the following is an important factor when reviewing a software specification document for correctness?

A) The document is properly formatted
B) The system’s functionality matches the user’s expectations
C) The system code is free from errors
D) The project team is in agreement with the scope
✅ **Answer:** B) The system’s functionality matches the user’s expectations

1. Which of the following is the primary objective of refining a software specification?

A) To write the source code
B) To ensure that requirements are clear and achievable
C) To test the system's performance
D) To deploy the system to production
✅ **Answer:** B) To ensure that requirements are clear and achievable

---

2. What is the primary focus of the software blueprint methodology in design?

A) To identify user interface elements
B) To create a detailed design document based on requirements
C) To test the system's functionality
D) To define the programming language to be used
✅ **Answer:** B) To create a detailed design document based on requirements

---

3. In the context of software design, which of the following is an example of an architectural design decision?

A) Choosing a user interface layout
B) Deciding how to store data in the database
C) Selecting a programming language for development
D) Defining the system's data flow and overall structure
✅ **Answer:** D) Defining the system's data flow and overall structure

---

4. Which design paradigm focuses on modeling real-world entities and their interactions using classes and objects?

A) Procedural design
B) Object-oriented design
C) Data design
D) Architectural design
✅ **Answer:** B) Object-oriented design

---

5. Which of the following is a key principle of object-oriented design?

A) Focus on processes and functions
B) Focus on the data and structures
C) Focus on real-world entities through classes and objects
D) Focus on file management
✅ **Answer:** C) Focus on real-world entities through classes and objects

---

6. In the software design process, what is the purpose of the design document?

A) To implement the system’s functionality
B) To specify how the system will be built
C) To describe the software's performance benchmarks
D) To deploy the application
✅ **Answer:** B) To specify how the system will be built

---

7. When creating a design document, it is crucial to review its conformance to:

A) The performance requirements
B) The coding standards
C) The software requirements and quality standards
D) The deployment strategy
✅ **Answer:** C) The software requirements and quality standards

---

8. What is an important consideration when applying fundamental design concepts for data design?

A) Minimizing the size of the source code
B) Ensuring the data structures are easy to implement and maintain
C) Focusing on the user interface design
D) Selecting the best programming language
✅ **Answer:** B) Ensuring the data structures are easy to implement and maintain

---

9. Which of the following is NOT typically addressed in procedural design?

A) Defining the sequence of operations
B) Identifying data entities

C) Structuring the flow of control in a program

D) Specifying algorithms for system functions

✅ **Answer:** B) Identifying data entities

---

10. What does the architectural design focus on in the software development process?

A) User interface design

B) High-level system structure and components

C) Database schema design

D) Coding specific system modules

✅ **Answer:** B) High-level system structure and components

---

11. Which phase of software design involves breaking down the system into smaller, manageable components?

A) Data design

B) Architectural design

C) Procedural design

D) System deployment

✅ **Answer:** B) Architectural design

---

12. What is the first step in applying software blueprint methodology in design?

A) Writing the code

B) Identifying functional and non-functional requirements

C) Choosing the system architecture

D) Designing the user interface

✅ **Answer:** B) Identifying functional and non-functional requirements

---

13. Object-oriented design primarily focuses on:

A) Dividing the system into smaller functions

B) Creating data models for storage

C) Modeling real-world objects and their interactions

D) Ensuring system performance under load

✅ **Answer:** C) Modeling real-world objects and their interactions

---

14. Which of the following is a characteristic of a good design document?

A) It is concise and free of technical details
B) It includes both high-level and low-level system designs
C) It avoids any mention of testing requirements
D) It only focuses on the user interface design
✅ **Answer:** B) It includes both high-level and low-level system designs

---

15. How is quality assurance typically incorporated into software design?

A) By focusing on aesthetic design
B) By reviewing the design document for clarity and correctness
C) By adding extra features and functionalities
D) By testing the system after coding is completed
✅ **Answer:** B) By reviewing the design document for clarity and correctness

---

16. When designing software, which concept helps in creating modular and reusable components?

A) Structured programming
B) Data normalization
C) Encapsulation
D) Inheritance
✅ **Answer:** C) Encapsulation

---

17. What is the main benefit of using the object-oriented design paradigm?

A) It simplifies the coding process
B) It promotes the reuse of code through inheritance
C) It eliminates the need for testing
D) It minimizes the development time
✅ **Answer:** B) It promotes the reuse of code through inheritance

---

18. During the creation of the design document, what should be reviewed to ensure conformance to the software requirements?

A) The system’s memory usage
B) The accuracy of functional specifications
C) The system’s error handling strategy
D) The adherence to coding standards
✅ **Answer:** B) The accuracy of functional specifications

---

19. Which of the following design concepts focuses on the relationships between system components?

A) Data design
B) Architectural design
C) Procedural design
D) User interface design
✅ **Answer:** B) Architectural design

---

20. What is the purpose of the refinement step in software design?

A) To optimize the system’s performance
B) To make the design more detailed and precise
C) To test the system against user requirements
D) To finalize the user interface design
✅ **Answer:** B) To make the design more detailed and precise

---

21. In object-oriented design, what is used to represent real-world entities as classes?

A) Methods
B) Objects
C) Attributes
D) Functions
✅ **Answer:** B) Objects

---

22. Which of the following is NOT a benefit of using object-oriented design?

A) It allows for better code maintenance
B) It supports polymorphism and encapsulation
C) It reduces the need for code reuse
D) It enables easier management of large systems
✅ **Answer:** C) It reduces the need for code reuse

---

23. What does the ″procedural design″ aspect of software design mainly focus on?

A) Modeling data and objects
B) Determining the structure and flow of processes and functions
C) Defining system requirements
D) Designing user interfaces
✅ **Answer:** B) Determining the structure and flow of processes and functions

---

24. Which of the following is a key characteristic of a software blueprint?

A) It contains detailed, low-level implementation code
B) It serves as a high-level plan for designing the system
C) It is used solely for testing system performance
D) It focuses on designing the user interface
✅ **Answer:** B) It serves as a high-level plan for designing the system

---

25. In a software design document, what does the ″data design″ section typically address?

A) The structure of the database and data flow
B) The user interface design
C) The choice of programming language
D) The system’s performance benchmarks
✅ **Answer:** A) The structure of the database and data flow

---

26. Which of the following is a fundamental design concept used in object-oriented design?

A) Inheritance
B) Static variables
C) Function overloading
D) Binary search
✅ **Answer:** A) Inheritance

---

27. Which of the following is an example of a design decision made during the architectural design phase?

A) How the data will be validated
B) Which programming language will be used
C) The structure of the system’s modules and components
D) The system’s user interface layout
✅ **Answer:** C) The structure of the system’s modules and components

---

28. What is the role of a ″system blueprint″ in the software design process?

A) To provide a detailed plan for the implementation phase
B) To define the system's hardware requirements
C) To specify the exact source code
D) To outline the system’s overall structure and functionality
✅ **Answer:** D) To outline the system’s overall structure and functionality

---

29. What should be reviewed when assessing the conformance of a design document to quality standards?

A) The system's coding style
B) The completeness and correctness of the requirements
C) The choice of programming language
D) The size of the source code
✅ **Answer:** B) The completeness and correctness of the requirements

---

30. In software design, which of the following is a key goal of ″refinement″?

A) To eliminate non-functional requirements
B) To increase the level of detail and accuracy in the design

C) To test the system’s functionality
D) To write the first version of the system’s code
✅ **Answer:** B) To increase the level of detail and accuracy in the design

---

31. What is a major advantage of object-oriented design?

A) It focuses primarily on the database design
B) It promotes the creation of reusable software components
C) It minimizes the need for testing
D) It requires no initial planning
✅ **Answer:** B) It promotes the creation of reusable software components

---

32. Which of the following best describes the concept of ″modularity″ in software design?

A) Dividing the system into smaller, independent, and reusable components
B) Focusing on system performance testing
C) Designing the system without defining the components
D) Writing complex and large blocks of code
✅ **Answer:** A) Dividing the system into smaller, independent, and reusable components

---

33. In software design, what is meant by the term ″conformance″ in design review?

A) Ensuring that the design meets functional and non-functional requirements
B) Checking if the design adheres to user interface standards
C) Verifying that the code is bug-free
D) Validating the deployment process
✅ **Answer:** A) Ensuring that the design meets functional and non-functional requirements

---

34. What is the purpose of ″procedural design″ in the software development process?

A) To define the structure of the data used in the system
B) To describe the flow of control and actions in the system
C) To outline the system’s architecture

D) To create the user interface layout
✅ **Answer:** B) To describe the flow of control and actions in the system

---

35. What is the advantage of using a software blueprint methodology in system design?

A) It allows for quick development without planning
B) It offers a structured approach to designing complex systems
C) It minimizes the time required for coding
D) It eliminates the need for testing
✅ **Answer:** B) It offers a structured approach to designing complex systems

---

36. Which of the following is part of the ″review of conformance to quality″ in a design document?

A) Ensuring that the design has no errors
B) Confirming that the system design follows industry standards and best practices
C) Testing the code after deployment
D) Evaluating the user interface design for aesthetics
✅ **Answer:** B) Confirming that the system design follows industry standards and best practices

---

37. What is a key advantage of the object-oriented design approach in terms of system maintenance?

A) It requires less code
B) It simplifies debugging and code changes due to encapsulation
C) It eliminates the need for testing
D) It reduces system complexity
✅ **Answer:** B) It simplifies debugging and code changes due to encapsulation

---

38. When creating a design document, which of the following is essential for ensuring the software meets user expectations?

A) Describing how the system will be deployed
B) Identifying the technical specifications of hardware
C) Including detailed functional and non-functional requirements

D) Defining the coding style to be followed

✅ **Answer:** C) Including detailed functional and non-functional requirements

---

39. Which of the following is a primary concern of architectural design?

A) How to implement the system's features
B) How the data will be validated
C) How to structure the system's components for scalability and maintainability
D) How to create the user interface

✅ **Answer:** C) How to structure the system's components for scalability and maintainability

---

40. What is the significance of refinement in software design?

A) To rewrite the system's source code
B) To clarify and elaborate on the design details
C) To check if the system meets the project budget
D) To identify which features should be eliminated

✅ **Answer:** B) To clarify and elaborate on the design details

1. What is the primary relationship between software design and implementation?

A) Design is the first step, while implementation follows it with little relation
B) Design outlines the structure, while implementation translates it into code
C) Design focuses on coding languages, while implementation defines the architecture
D) Design involves writing the code, and implementation is used to test the design
✅ **Answer:** B) Design outlines the structure, while implementation translates it into code

---

2. Which of the following is an implementation issue that must be addressed during the software development process?

A) Database schema design
B) Identifying user requirements
C) Resource management, such as memory and CPU usage
D) Defining project timelines
✅ **Answer:** C) Resource management, such as memory and CPU usage

---

3. What does a programming support environment typically include?

A) Network configuration
B) Tools like IDEs, debuggers, and version control systems
C) Hardware requirements
D) Legal and licensing documents
✅ **Answer:** B) Tools like IDEs, debuggers, and version control systems

---

4. Which phase of software development focuses on converting design elements into executable code?

A) Testing
B) Design
C) Implementation
D) Maintenance
✅ **Answer:** C) Implementation

---

5. Good coding style primarily improves:

A) The appearance of the code
B) The performance of the system
C) The readability and maintainability of the code
D) The size of the executable file
✅ **Answer:** C) The readability and maintainability of the code

---

6. Which of the following is an example of a good coding practice?

A) Writing all code in a single line to save space
B) Using descriptive variable names and consistent indentation
C) Avoiding the use of comments in code
D) Writing code without considering performance
✅ **Answer:** B) Using descriptive variable names and consistent indentation

---

7. During the implementation phase, reviewing the code for correctness helps ensure:

A) The code follows the coding style guidelines
B) The system meets the specified requirements
C) The code is optimized for performance
D) All of the above
✅ **Answer:** D) All of the above

---

8. What does the term ″code readability″ refer to in software implementation?

A) The efficiency of the code in terms of execution time
B) The ability to understand the code easily by other developers
C) The complexity of the algorithm used in the code
D) The number of lines of code written
✅ **Answer:** B) The ability to understand the code easily by other developers

---

9. Which of the following is a common type of testing performed during software implementation?

A) Compatibility testing
B) Unit testing
C) Market research testing
D) Deployment testing
✅ **Answer:** B) Unit testing

---

10. In which of the following situations is unit testing most appropriate?

A) Testing the entire system after deployment
B) Testing individual components or functions for correctness
C) Testing the system's user interface
D) Testing network connectivity
✅ **Answer:** B) Testing individual components or functions for correctness

---

11. Which of the following is a goal of good coding style?

A) Minimizing the number of lines in the code
B) Making the code easy to read, understand, and maintain
C) Writing complex code that is difficult to optimize
D) Focusing on the visual appearance of the code
✅ **Answer:** B) Making the code easy to read, understand, and maintain

---

12. Which of the following is NOT part of the testing phase in software implementation?

A) Unit testing
B) Integration testing
C) Stress testing
D) Requirements gathering
✅ **Answer:** D) Requirements gathering

---

13. What is the primary purpose of integration testing?

A) To test individual functions and methods
B) To check the system’s performance under load
C) To ensure that different modules or components work together correctly
D) To test the security features of the system

✅ **Answer:** C) To ensure that different modules or components work together correctly

---

14. Which of the following is an issue that could be encountered during the implementation phase?

A) Lack of sufficient system resources
B) Incorrect user requirements
C) Insufficient training for end-users
D) The absence of a project manager
✅ **Answer:** A) Lack of sufficient system resources

---

15. In the context of implementation, what is the primary focus of procedural coding?

A) Defining the system’s architecture
B) Writing a sequence of instructions to be executed by the computer
C) Creating a modular and reusable code structure
D) Defining object interactions and behaviors
✅ **Answer:** B) Writing a sequence of instructions to be executed by the computer

---

16. Which of the following testing techniques focuses on evaluating the system’ s overall functionality?

A) Unit testing
B) Integration testing
C) System testing
D) Regression testing
✅ **Answer:** C) System testing

---

17. Which of the following is NOT an example of testing during software implementation?

A) Debugging errors in code
B) Testing individual software modules
C) Testing hardware compatibility
D) Writing detailed requirements documents
✅ **Answer:** D) Writing detailed requirements documents

---

18. What is the purpose of ″test-driven development″ (TDD)?

A) To write tests after coding the entire system
B) To write tests before writing the actual code
C) To minimize the amount of testing performed during development
D) To focus solely on unit testing
✅ **Answer:** B) To write tests before writing the actual code

---

19. Which of the following testing types is designed to ensure that the system continues to work as expected after changes have been made?

A) Regression testing
B) Load testing
C) User acceptance testing
D) Stress testing
✅ **Answer:** A) Regression testing

---

20. What is the main advantage of a good programming support environment?

A) It helps in writing code faster
B) It allows for efficient debugging, version control, and collaboration
C) It eliminates the need for testing
D) It reduces the system's complexity
✅ **Answer:** B) It allows for efficient debugging, version control, and collaboration

---

21. Which of the following would be a common implementation problem when coding?

A) Ensuring that the software is designed according to user requirements
B) Finding bugs and errors in the code
C) Gathering feedback from stakeholders
D) Writing user manuals
✅ **Answer:** B) Finding bugs and errors in the code

---

22. What type of testing is done to check how well the system behaves under stress or heavy loads?

A) Unit testing
B) Load testing
C) Regression testing
D) Acceptance testing
✅ **Answer:** B) Load testing

---

23. Which of the following is an essential aspect of code readability?

A) Use of descriptive and meaningful variable names
B) Including very few comments
C) Writing as many lines of code as possible
D) Using only abbreviations for function names
✅ **Answer:** A) Use of descriptive and meaningful variable names

---

24. In which testing phase would you check if the software is compatible with different operating systems and browsers?

A) Unit testing
B) Integration testing
C) Compatibility testing
D) System testing
✅ **Answer:** C) Compatibility testing

---

25. What does ″refactoring″ refer to in the context of software implementation?

A) Writing new features for the system
B) Rewriting code to improve its readability and structure without changing its functionality
C) Implementing new system requirements
D) Adding new users to the system
✅ **Answer:** B) Rewriting code to improve its readability and structure without changing its functionality

---

26. Which of the following is a key characteristic of good code documentation?

A) It focuses on explaining the code's logic and purpose clearly
B) It is written after the code is completed
C) It only describes the function names
D) It includes irrelevant information
✅ **Answer:** A) It focuses on explaining the code's logic and purpose clearly

---

27. What is the main goal of using version control systems in the implementation phase?

A) To track and manage changes in the codebase over time
B) To design the user interface
C) To perform the final system testing
D) To optimize the code performance
✅ **Answer:** A) To track and manage changes in the codebase over time

---

28. Which of the following is an example of a coding error?

A) Using descriptive variable names
B) Forgetting to close a loop or function
C) Including comments to explain code logic
D) Using modular code structures
✅ **Answer:** B) Forgetting to close a loop or function

---

29. What is the goal of system testing in software implementation?

A) To test the individual functions and modules
B) To ensure the software meets all system requirements and works as expected
C) To test the network performance
D) To check the hardware compatibility
✅ **Answer:** B) To ensure the software meets all system requirements and works as expected

---

30. What is the purpose of "error handling" in software implementation?

A) To prevent the system from encountering bugs
B) To allow the system to gracefully handle and recover from errors
C) To prevent the system from being tested
D) To improve the performance of the system
✅ **Answer:** B) To allow the system to gracefully handle and recover from errors

---

31. Which of the following is a characteristic of well-structured code?

A) Code that is difficult for others to understand
B) Code that follows consistent indentation and naming conventions
C) Code that only runs on one platform
D) Code that does not require testing
✅ **Answer:** B) Code that follows consistent indentation and naming conventions

---

32. Which testing phase is primarily concerned with ensuring the software meets user requirements?

A) Unit testing
B) Integration testing
C) Acceptance testing
D) System testing
✅ **Answer:** C) Acceptance testing

---

33. What does ″debugging″ involve during the implementation phase?

A) Writing new features
B) Identifying and fixing errors in the code
C) Designing the system architecture
D) Writing the system documentation
✅ **Answer:** B) Identifying and fixing errors in the code

---

34. What is the benefit of using automated testing tools in software implementation?

A) It reduces the time required to write the code
B) It allows the software to be tested more frequently and efficiently

C) It eliminates the need for testing altogether
D) It ensures the system will run without errors
✅ **Answer:** B) It allows the software to be tested more frequently and efficiently

---

35. Which of the following is NOT an example of a type of testing during the implementation phase?

A) Unit testing
B) Integration testing
C) Performance testing
D) User interface design testing
✅ **Answer:** D) User interface design testing

---

36. When should testing be performed in the software implementation phase?

A) Only after the code is fully developed
B) Continuously throughout the implementation process
C) Only at the end of the project
D) Only after user feedback is collected
✅ **Answer:** B) Continuously throughout the implementation process

---

37. Which of the following is a common testing technique used to identify performance issues in software?

A) Load testing
B) Unit testing
C) Code reviews
D) User acceptance testing
✅ **Answer:** A) Load testing

---

38. Why is it important to use version control systems during implementation?

A) To track changes in the code and collaborate with other developers
B) To reduce the size of the codebase
C) To test the system’s performance

D) To increase the speed of code execution

✅ **Answer:** A) To track changes in the code and collaborate with other developers

---

39. In software implementation, what is the primary function of the coding phase?

A) To test the software
B) To define system requirements
C) To translate design specifications into executable code
D) To deploy the system to users

✅ **Answer:** C) To translate design specifications into executable code

---

40. Which of the following best describes the ″maintenance″ phase in the software development lifecycle?

A) Making fixes, improvements, and updates after the software is deployed
B) Writing code for the core functionality
C) Testing the code for errors
D) Gathering user feedback for the initial system requirements

✅ **Answer:** A) Making fixes, improvements, and updates after the software is deployed

1. What is the primary goal of software maintenance?

A) To evaluate the software for performance improvements
B) To ensure the software continues to meet the user’s evolving needs
C) To update the software’s user interface
D) To rewrite the codebase from scratch
✅ **Answer:** B) To ensure the software continues to meet the user’s evolving needs

---

2. Which of the following is NOT a reason for software maintenance?

A) Adding new features
B) Fixing defects
C) Rewriting the entire software
D) Adapting to changes in the environment
✅ **Answer:** C) Rewriting the entire software

---

3. Which of the following is an example of adaptive maintenance?

A) Fixing a bug in the software
B) Upgrading software to run on a new operating system
C) Adding a new feature to the software
D) Improving the software’s speed and performance
✅ **Answer:** B) Upgrading software to run on a new operating system

---

4. What does corrective maintenance aim to do?

A) Add new features
B) Fix defects or bugs in the software
C) Adapt software to new environments
D) Improve the software’s usability
✅ **Answer:** B) Fix defects or bugs in the software

---

5. What type of software maintenance involves modifying the software to make it more efficient?

A) Adaptive maintenance
B) Corrective maintenance
C) Perfective maintenance
D) Preventive maintenance
✅ **Answer:** C) Perfective maintenance

---

6. Which of the following is a key principle when designing software for maintainability?

A) Minimizing the number of features
B) Writing as much code as possible
C) Ensuring code is modular and easy to understand
D) Complicating the code structure to prevent future changes
✅ **Answer:** C) Ensuring code is modular and easy to understand

---

7. Which of the following is a technique commonly used for software maintenance?

A) Reverse engineering
B) High-level design
C) Requirements gathering
D) Data modeling
✅ **Answer:** A) Reverse engineering

---

8. What is reverse engineering used for in software maintenance?

A) To write new software from scratch
B) To understand and recreate the software's functionality based on its existing structure
C) To optimize the software's performance
D) To test the software under different conditions
✅ **Answer:** B) To understand and recreate the software's functionality based on its existing structure

---

9. Which type of maintenance is used to make software work in a new environment?

A) Corrective maintenance
B) Adaptive maintenance
C) Perfective maintenance
D) Preventive maintenance
✅ **Answer:** B) Adaptive maintenance

---

10. What is software re-engineering primarily concerned with?

A) Rewriting the software code entirely
B) Modifying the software to improve its structure without changing its functionality
C) Adding new features to the software
D) Removing outdated features from the software
✅ **Answer:** B) Modifying the software to improve its structure without changing its functionality

---

11. What is the primary aim of preventive maintenance?

A) To fix defects
B) To ensure that no future defects will occur
C) To improve the software's performance
D) To adapt the software to a new environment
✅ **Answer:** B) To ensure that no future defects will occur

---

12. Which of the following is a key challenge in software maintenance?

A) Writing code from scratch
B) Balancing bug fixes with new feature development
C) Designing new architectures
D) Reducing the software’s cost
✅ **Answer:** B) Balancing bug fixes with new feature development

---

13. What is maintenance as part of software evaluation focused on?

A) Conducting market research for new features
B) Determining the system’s health and making necessary changes
C) Identifying new opportunities for software development

D) Testing the software under stress

✅ **Answer:** B) Determining the system’s health and making necessary changes

---

14. In software maintenance, what is the process of making software more maintainable known as?

A) Refactoring

B) Version control

C) Reverse engineering

D) Debugging

✅ **Answer:** A) Refactoring

---

15. What is a key advantage of designing software for maintainability?

A) Reduces the initial development time

B) Makes it easier to add new features, fix bugs, and adapt to changes in the environment

C) Increases the cost of the project

D) Eliminates the need for testing

✅ **Answer:** B) Makes it easier to add new features, fix bugs, and adapt to changes in the environment

---

16. Which of the following is a common tool used in software maintenance?

A) Compilers

B) Code editors

C) Debuggers

D) UML diagrams

✅ **Answer:** C) Debuggers

---

17. What does software reverse engineering typically involve?

A) Writing code from scratch

B) Deconstructing the software to understand its inner workings

C) Creating software without a specification

D) Analyzing the hardware that runs the software
✅ **Answer:** B) Deconstructing the software to understand its inner workings

---

18. Which of the following is a key disadvantage of reverse engineering?

A) It is difficult to analyze the code
B) It often requires breaking down existing software into simpler forms
C) It is time-consuming and costly
D) It can violate intellectual property laws
✅ **Answer:** D) It can violate intellectual property laws

---

19. Which type of maintenance involves changes to improve software performance or add features without changing the system’ s functionality?

A) Corrective maintenance
B) Adaptive maintenance
C) Perfective maintenance
D) Preventive maintenance
✅ **Answer:** C) Perfective maintenance

---

20. What is the main purpose of software re-engineering?

A) To re-create software in a new programming language
B) To improve the software’s architecture and design while maintaining its functionality
C) To develop entirely new features for the software
D) To debug the software and remove errors
✅ **Answer:** B) To improve the software’s architecture and design while maintaining its functionality

---

21. What is an example of corrective maintenance?

A) Changing the software to run on a new operating system
B) Fixing a bug reported by users
C) Adding a new feature to the software

D) Updating the software to support new hardware
✅ **Answer:** B) Fixing a bug reported by users

---

22. Which maintenance type deals with modifying the software to meet changes in the hardware or operating environment?

A) Corrective maintenance
B) Adaptive maintenance
C) Perfective maintenance
D) Preventive maintenance
✅ **Answer:** B) Adaptive maintenance

---

23. In which scenario would reverse engineering most likely be used?

A) To improve system security
B) To recover or understand software whose source code is unavailable
C) To create a new software product from scratch
D) To test the software
✅ **Answer:** B) To recover or understand software whose source code is unavailable

---

24. Which of the following is an example of preventive maintenance?

A) Modifying the software to ensure it performs better under increased load
B) Fixing a known bug in the system
C) Adapting software to a new operating system
D) Removing outdated features from the software
✅ **Answer:** A) Modifying the software to ensure it performs better under increased load

---

25. What is the main goal of re-engineering?

A) To convert old code into a new programming language
B) To remove redundant features from the system
C) To improve the structure and performance of software without altering its functionality
D) To develop entirely new features for the software

✅ **Answer:** C) To improve the structure and performance of software without altering its functionality

---

26. What does the reverse engineering process involve when applied to software?

A) Rewriting the software code to improve efficiency
B) Deconstructing and analyzing existing software to understand how it works
C) Testing software to find bugs
D) Adding new features to an existing software system
✅ **Answer:** B) Deconstructing and analyzing existing software to understand how it works

---

27. In software re-engineering, which of the following activities is typically performed?

A) Refactoring the code to improve its maintainability
B) Designing new software features
C) Building new software from scratch
D) Developing a new user interface
✅ **Answer:** A) Refactoring the code to improve its maintainability

---

28. Which of the following is a common challenge in software maintenance?

A) Lack of available design documentation
B) Overabundance of new features to add
C) Reduced need for ongoing testing
D) A clear and well-defined software architecture
✅ **Answer:** A) Lack of available design documentation

---

29. In software maintenance, why is maintainability an important design principle?

A) It ensures the software is easy to change and update over time
B) It minimizes the software’s execution time
C) It prevents the need for reverse engineering

D) It reduces the number of features
✅ **Answer:** A) It ensures the software is easy to change and update over time

---

30. Which of the following best describes perfective maintenance?

A) Maintenance focused on fixing defects
B) Maintenance focused on adapting software to a new environment
C) Maintenance focused on adding new features or improving the software’s performance
D) Maintenance focused on identifying and preventing future issues
✅ **Answer:** C) Maintenance focused on adding new features or improving the software’s performance

---

31. Which of the following is a common result of reverse engineering?

A) A redesign of the software’s architecture
B) A clear understanding of how a software system functions internally
C) A new version of the software with added features
D) The discovery of unnecessary software components
✅ **Answer:** B) A clear understanding of how a software system functions internally

---

32. Why is software re-engineering important for legacy systems?

A) It enables companies to reuse older systems with updated technologies
B) It reduces the time needed to develop new software from scratch
C) It fixes bugs and errors without any code changes
D) It updates the user interface of the system
✅ **Answer:** A) It enables companies to reuse older systems with updated technologies

---

33. What is a key benefit of preventive maintenance?

A) It ensures software meets current requirements
B) It anticipates and prevents future defects
C) It adds new features to the system

D) It fixes existing bugs in the software
✅ **Answer:** B) It anticipates and prevents future defects

---

34. Which of the following would NOT typically be considered adaptive maintenance?

A) Updating software to support a new operating system
B) Rewriting software in a new programming language
C) Modifying software to support new hardware
D) Adding new features to the system
✅ **Answer:** B) Rewriting software in a new programming language

---

35. What does reverse engineering NOT typically involve?

A) Understanding the software’s design and functionality
B) Extracting the source code from a compiled application
C) Modifying the software to improve its performance
D) Documenting the internal workings of the software
✅ **Answer:** C) Modifying the software to improve its performance

---

36. How can refactoring assist in software maintenance?

A) It helps identify and fix defects
B) It allows for adding new features
C) It improves the software’s internal structure without altering its functionality
D) It adds new functionality to the software
✅ **Answer:** C) It improves the software’s internal structure without altering its functionality

---

37. Which of the following is NOT a characteristic of good maintainable code?

A) Clear and consistent naming conventions
B) High coupling between components
C) Use of comments to explain complex logic
D) Modularity of code
✅ **Answer:** B) High coupling between components

---

38. Software re-engineering primarily involves which of the following?

A) Rewriting the software code from scratch
B) Making small changes to adapt the software to a new platform
C) Improving the software's internal design without changing its external functionality
D) Creating a new software product
✅ **Answer:** C) Improving the software's internal design without changing its external functionality

---

39. What is the role of version control in software maintenance?

A) To reduce the number of bugs in the software
B) To track changes and manage different versions of the software
C) To optimize the software’s performance
D) To add new features to the software
✅ **Answer:** B) To track changes and manage different versions of the software

---

40. Why is documentation important in software maintenance?

A) It allows the developers to write code faster
B) It helps maintain the software, especially when new developers are involved
C) It ensures the software will never require future maintenance
D) It speeds up the debugging process
✅ **Answer:** B) It helps maintain the software, especially when new developers are involved