Unit-V

1. **What is the primary purpose of Computer-Aided Software Engineering (CASE) tools in the software development process?**
   a. To replace human developers
   b. To automate the entire software development life cycle
   c. To assist and support developers in various phases of the development process
   d. To conduct user acceptance testing

2. **Which phase of the software development life cycle is CASE tool usage most common?**
   a. Requirements analysis
   b. Design
   c. Coding
   d. Testing

3. **What does the term "forward engineering" refer to in the context of CASE tools?**
   a. The process of reverse-engineering existing code
   b. The generation of code from high-level models or specifications
   c. The analysis of system requirements
   d. The manual coding of software

4. **Which type of CASE tool is primarily used for creating graphical representations of the system, such as diagrams and charts?**
   a. Code generators
   b. Modeling tools
   c. Testing tools
   d. Debugging tools

5. **What is the primary advantage of using CASE tools in the analysis and design phases?**
   a. Increased project cost
   b. Reduced developer productivity
   c. Improved documentation and visualisation
   d. Limited collaboration among team members

6. **Which aspect of the software development process is often automated by CASE tools to ensure consistency and adherence to coding standards?**
   a. Requirement gathering
   b. User interface design
   c. Code generation
   d. Project management

7. **What does the term "reverse engineering" mean in the context of CASE tools?**
   a. Generating code from high-level models
   b. Analysing existing code to create high-level models or documentation
   c. Debugging code
   d. Creating system requirements

8. **Which CASE tool feature is essential for tracking changes made to the software over time and managing version control?**
   a. Code generation b. Version control
   c. Debugging d. Testing

9. **What is the purpose of using CASE tools for documentation in the analysis and design phases?**
   a. To replace the need for documentation
   b. To create visually appealing documents
   c. To automate the entire documentation process
   d. To produce consistent and comprehensive documentation

10. Which CASE tool type supports the testing phase by automating test case generation and execution?
a. Modeling tools b. Testing tools
c. Debugging tools d. Code generators

11. What is the primary goal of Object-Oriented Analysis and Design (OOAD)?
a. To focus on procedural programming
b. To emphasize hardware components
c. To model the system using objects and their interactions
d. To ignore the need for system documentation

12. In OOAD, what is an "object"?
a. A physical device
b. An instance of a class with specific attributes and behaviors
c. A programming language
d. A database table

13. What is the purpose of a Class Diagram in OOAD?
a. To represent the flow of data within a system
b. To illustrate the system's architecture
c. To define the structore and relationships of classes in the system
d. To focus on user interfaces

14. Which OOAD concept represents the behavior of an object in response to a message or a method call?
a. Inheritance b. Encapsulation
c. Polymorphism d. Abstraction

15. What does the term "encapsulation" mean in the context of OOAD?
a. Combining data and methods that operate on the data into a single unit
b. Separating data and methods to increase modularity
c. Representing the structure of a system using objects
d. Ignoring the need for system documentation

16. Which OOAD concept allows a new class to inherit attributes and behaviors from an existing class?
a. Encapsulation b. Polymorphism
c. Inheritance d. Abstraction

17. What is the purpose of an Object Diagram in OOAD?
a. To represent the flow of data within a system
b. To illustrate the system's architecture
c. To provide an instance-level view of a system's classes and their relationships
d. To focus on user interfaces

18. Which OOAD concept involves simplifying complex systems by modeling only the essential features of the system?
a. Abstraction b. Polymorphism
c. Inheritance d. Encapsulation

19. What is the purpose of a Use Case Diagram in OOAD?
a. To define the structure and relationships of classes
b. To represent the flow of data within a system
c. To illustrate how users interact with the system and its features
d. To provide an instance-level view of a system's classes

20. In OOAD, what is a "message" in the context of object interactions?
a. A notification from the operating system
b. A request for a specific service or information from an object
c. A piece of system documentation
d. A hardware component

21. What is the primary focus of dynamic modeling in Object-Oriented Analysis and Design (OOAD)?
a. Representing the structure of a system
b. Capturing the static relationships between objects
c. Modeling the behavior and interactions of objects over time
d. Ignoring system documentation

22. Which UML diagram is commonly used for dynamic modeling to represent the sequence of interactions between objects?
a. Class Diagram
b. Use Case Diagram
c. Statechart Diagram
d. Sequence Diagram

23. In dynamic modeling, what does a state represent in a Statechart Diagram?
a. A specific instance of a class
b. The attributes of an object
c. The behavior of an object in response to events
d. The relationships between classes

24. What is the purpose of an Activity Diagram in dynamic modeling?
a. To represent the flow of data within a system
b. To illustrate the system's architecture
c. To model the dynamic aspects of a system, such as workflow or business processes
d. To focus on user interfaces

25. What does the term "collaboration" refer to in the context of dynamic modeling?
a. The cooperation between developers
b. The interaction between classes and objects
c. The arrangement of classes in a system
d. The encapsulation of data and methods in a class

26. What is the primary goal of Functional Object-Oriented Design (FOOD)?
a. To emphasize procedural programming
b. To model the system using objects and their interactions
c. To focus solely on end-user needs
d. To represent the system in terms of functions and data transformations

27. Which concept in FOOD involves organising functions into classes and associating data with those functions?
a. Abstraction b. Encapsulation
c. Inheritance d. Polymorphism

28. In FOOD, what is the significance of the term "functional decomposition"?
a. Breaking down complex functions into smaller, manageable functions
b. Combining functions and data into a single unit
c. Creating class hierarchies
d. Modeling the dynamic aspects of a system

29. What is the purpose of a Functional Decomposition Diagram in FOOD?
a. To represent the flow of data within a system
b. To illustrate the system's architecture
c. To model the dynamic behavior of objects
d. To decompose functions into sub-functions and show their relationships

30. Which concept in FOOD involves the reuse of functions and data structures in different parts of the system?
a. Abstraction b. Encapsulation
c. Inheritance d. Polymorphism

31. What is the primary goal of Object-Oriented Programming (OOP) in system implementation?
a. To emphasize procedural programming
b. To create complex algorithms
c. To model the system using objects and their interactions
d. To focus solely on end-user needs

32. Which OOP principle involves bundling data and methods that operate on the data into a single unit?
a. Abstraction b. Encapsulation
c. Inheritance d. Polymorphism

33. What is the purpose of constructors in OOP?
a. To create new classes
b. To destroy objects
c. To initialise the state of an object when it is created
d. To define class relationships

34. Which OOP concept allows a new class to inherit attributes and behaviors from an existing class?
a. Encapsulation b. Polymorphism
c. Inheritance d. Abstraction

35. What does the term "polymorphism" mean in the context of OOP?
a. The ability of an object to take on multiple forms
b. The encapsulation of data and methods in a class
c. The process of creating new classes from existing ones
d. The separation of interface from implementation

36. Which OOP principle allows a class to have multiple methods with the same name but different parameters?
a. Abstraction b. Encapsulation
c. Inheritance d. Method Overloading

37. In OOP, what is the purpose of the "super" keyword in Java or similar constructs in other languages?
a. To access the superclass of a derived class
b. To declare a variable
c. To create a new object
d. To define an interface

38. **Which OOP concept involves the ability of a class to provide a common interface for multiple classes?**
    a. Abstraction
    b. Encapsulation
    c. Inheritance
    d. Interface Implementation

39. **What is the purpose of the "this" keyword in OOP languages like Java or C++?**
    a. To create a new object
    b. To refer to the current instance of a class
    c. To access the superclass of a derived class
    d. To declare a variable

40. **What is the significance of the term 'dynamic binding' in OOP?**
    a. The process of linking the program at compile-time
    b. The ability to access methods and properties of an object at runtime
    c. The process of creating new classes
    d. The encapsulation of data and methods in a class