Here's a list of **150 software engineering viva questions with answers** designed for university students, covering various key areas such as software development, design, testing, databases, and software management.

General Software Engineering Concepts

1. **What is software engineering?**
   Answer: Software engineering is the application of engineering principles to the design, development, maintenance, testing, and evaluation of software and systems.
2. **What are the main goals of software engineering?**
   Answer: The main goals are to deliver software that is reliable, maintainable, scalable, and meets user requirements while being efficient and cost-effective.
3. **Explain the Software Development Life Cycle (SDLC).**
   Answer: SDLC is a structured approach to software development, consisting of stages like requirement gathering, design, implementation, testing, deployment, and maintenance.
4. **What are the phases in SDLC?**
   Answer: The phases include:
   - Requirement gathering
   - System design
   - Coding/Implementation
   - Testing
   - Deployment
   - Maintenance
5. **What is the purpose of requirement gathering?**
   Answer: The purpose is to understand the needs of the users and stakeholders to ensure the software meets the business and technical requirements.
6. **What is the difference between functional and non-functional requirements?**
   Answer: Functional requirements describe what the software should do (e.g., features), while non-functional requirements describe how the software should perform (e.g., security, performance).
7. **What is a Software Requirement Specification (SRS)?**
   Answer: SRS is a document that describes the software system to be developed, including its functionality, interfaces, and constraints.
8. **Explain the Waterfall model in SDLC.**
   Answer: The Waterfall model is a linear and sequential approach where each phase must be completed before the next one begins.
9. **What is the Agile methodology?**
   Answer: Agile is an iterative and incremental approach to software development that focuses on collaboration, flexibility, and customer feedback.
10. **What are the advantages and disadvantages of the Agile model?**
    Answer: Advantages: Flexibility, fast delivery, customer collaboration. Disadvantages: Requires constant customer involvement, may lead to scope creep.

11. **What is the V-Model?**
    Answer: The V-Model is an SDLC model where each phase of development is directly associated with a testing phase, ensuring quality at each step.
12. **What is a Spiral model?**
    Answer: The Spiral model combines iterative development and the Waterfall model, focusing on risk analysis and refinement through repeated cycles.
13. **What is a Prototype model in SDLC?**
    Answer: A prototype model involves creating an initial version of the software (prototype), which is refined based on user feedback.
14. **What is an iterative model?**
    Answer: The iterative model involves repeating cycles of development and refinement, with each cycle improving the software based on feedback.
15. **What is incremental model in software development?**
    Answer: In the incremental model, software is developed in smaller, manageable portions or increments, and each increment builds on the previous one.
16. **What is the importance of documentation in software engineering?**
    Answer: Documentation provides a reference for developers, stakeholders, and future teams, ensuring consistency, clarity, and easier maintenance.
17. **Explain the concept of "Software maintenance".**
    Answer: Software maintenance involves modifying and updating software after it has been delivered, including fixing bugs and adding new features.
18. **What is software validation?**
    Answer: Software validation ensures that the software meets the business needs and requirements, verifying if it does what it is supposed to do.
19. **What is software verification?**
    Answer: Software verification ensures that the software has been developed correctly according to the specifications and design.
20. **What is the importance of software testing?**
    Answer: Software testing ensures that the software functions correctly, is free of defects, and meets user requirements.
21. **What are the limitations of the Waterfall model?**
    Answer: The Waterfall model is rigid, doesn't handle changes easily, and requires complete requirements upfront, making it less suitable for complex projects with evolving requirements.
22. **What is the significance of version control?**
    Answer: Version control helps track changes to code, collaborate among team members, manage different versions, and avoid conflicts.
23. **What is the purpose of software design?**
    Answer: Software design defines the architecture, components, interfaces, and data flow, ensuring that the software meets the requirements efficiently.
24. **What is the difference between software architecture and software design?**
    Answer: Software architecture refers to the high-level structure and overall design of the system, while design focuses on specific components and their implementation.
25. **What is the role of a software engineer?**
    Answer: A software engineer is responsible for designing, coding, testing, and

maintaining software systems, ensuring they meet user requirements and are of high quality.

26. **What is software refactoring?**
    Answer: Refactoring is the process of restructuring existing code to improve readability, maintainability, and performance without changing its functionality.
27. **Explain the term "technical debt".**
    Answer: Technical debt refers to the additional work required to fix issues in the software that arise from poor design or quick solutions that were implemented to meet deadlines.
28. **What is the importance of design patterns in software engineering?**
    Answer: Design patterns provide proven, reusable solutions to common software design problems, improving code quality and reducing development time.
29. **What are the types of design patterns?**
    Answer: Design patterns are categorized into:
    - Creational (e.g., Singleton, Factory)
    - Structural (e.g., Adapter, Composite)
    - Behavioral (e.g., Observer, Strategy)
30. **What is a Class diagram?**
    Answer: A class diagram in UML (Unified Modeling Language) shows the structure of a system by representing its classes, their attributes, methods, and relationships.
31. **What is a Use Case diagram?**
    Answer: A use case diagram represents the interactions between users (actors) and the system, defining the system's functional requirements.
32. **What is the difference between system architecture and system design?**
    Answer: System architecture refers to the high-level structure of the system, while system design focuses on detailed components and how they function.
33. **What is the importance of performance metrics in software development?**
    Answer: Performance metrics help evaluate the effectiveness and efficiency of the software in terms of speed, resource usage, and scalability, ensuring it meets user expectations.

Object-Oriented Programming (OOP)

34. **What are the four pillars of OOP?**
    Answer: The four pillars of OOP are:
    - Encapsulation
    - Inheritance
    - Polymorphism
    - Abstraction
35. **What is encapsulation in OOP?**
    Answer: Encapsulation is the concept of hiding the internal details of an object and exposing only necessary functionality through methods.

36. **What is inheritance in OOP?**
    Answer: Inheritance allows a class to inherit properties and behaviors (methods) from another class, enabling code reuse and establishing relationships.
37. **What is polymorphism in OOP?**
    Answer: Polymorphism allows objects of different classes to be treated as objects of a common superclass, supporting method overriding and overloading.
38. **What is abstraction in OOP?**
    Answer: Abstraction hides complex implementation details and exposes only essential features of an object or system.
39. **What is the difference between a class and an object?**
    Answer: A class is a blueprint or template for creating objects, while an object is an instance of a class.
40. **What is a constructor in OOP?**
    Answer: A constructor is a special method that is called when an object is instantiated, initializing the object's properties.
41. **What is a destructor in OOP?**
    Answer: A destructor is a special method that is called when an object is destroyed or goes out of scope, used to clean up resources.
42. **What is method overloading?**
    Answer: Method overloading allows multiple methods with the same name but different parameter lists to exist within a class.
43. **What is method overriding?**
    Answer: Method overriding occurs when a subclass provides a specific implementation for a method that is already defined in its superclass.
44. **What is the difference between static and dynamic binding?**
    Answer: Static binding (or early binding) occurs at compile time, while dynamic binding (or late binding) occurs at runtime.
45. **What is the significance of the “this” keyword in OOP?**
    Answer: The "this" keyword refers to the current instance of a class, allowing access to its methods and properties.
46. **What is the purpose of an abstract class?**
    Answer: An abstract class is a class that cannot be instantiated and is meant to be subclassed, providing a base for other classes.
47. **What is an interface in OOP?**
    Answer: An interface is a contract that defines a set of methods that a class must implement, but it doesn't provide method implementations.
48. **What is the difference between an interface and an abstract class?**
    Answer: An interface defines a contract without any implementation, while an abstract class can have both abstract and implemented methods.
49. **What is multiple inheritance and how is it implemented in Java or C++?**
    Answer: Multiple inheritance occurs when a class inherits from more than one class. In Java, this is achieved using interfaces, while C++ supports it directly.
50. **What is the difference between public, private, and protected access modifiers in OOP?**
    Answer:
    - **Public**: Accessible from anywhere.
    - **Private**: Accessible only within the class.

- **Protected**: Accessible within the class and its subclasses.

This covers the first 50 questions. Would you like me to continue with the remaining 100 questions?

4o mini

You said:

yes

ChatGPT said:

ChatGPT

Here's the continuation of the **150 software engineering viva questions and answers**, covering additional areas such as testing, databases, version control, software architecture, and project management.

Software Testing

51. **What is software testing?**
    Answer: Software testing is the process of evaluating a software application to ensure it functions correctly and meets the specified requirements.
52. **What is the difference between verification and validation?**
    Answer: Verification ensures the software is built correctly according to specifications, while validation ensures the software meets the user's needs and requirements.
53. **What is unit testing?**
    Answer: Unit testing involves testing individual components or functions of a software program to ensure they work as expected.
54. **What is integration testing?**
    Answer: Integration testing focuses on testing the interaction between integrated components or systems to verify they work together as expected.
55. **What is system testing?**
    Answer: System testing tests the complete software system to ensure it meets the specified requirements and works correctly in the intended environment.
56. **What is acceptance testing?**
    Answer: Acceptance testing is performed by the client to verify that the software meets business needs and is ready for deployment.
57. **What is black-box testing?**
    Answer: Black-box testing focuses on testing the functionality of the software without knowledge of the internal code or logic.
58. **What is white-box testing?**
    Answer: White-box testing involves testing the internal structure and code of the software, focusing on the logic and flow of the program.

59. **What is regression testing?**
    Answer: Regression testing ensures that recent changes or updates to the software have not negatively affected existing functionality.
60. **What is load testing?**
    Answer: Load testing checks how the system performs under expected user loads to ensure it can handle normal traffic.
61. **What is stress testing?**
    Answer: Stress testing determines the software's behavior under extreme conditions, beyond its expected capacity.
62. **What is performance testing?**
    Answer: Performance testing measures the software's speed, responsiveness, and stability under various conditions.
63. **What is the difference between alpha testing and beta testing?**
    Answer: Alpha testing is performed by internal teams before releasing the product to external users, while beta testing is performed by a select group of external users to gather feedback.
64. **What is the importance of test cases?**
    Answer: Test cases define the conditions under which the software should be tested, ensuring all functionalities are covered and the software behaves as expected.
65. **What is the difference between functional and non-functional testing?**
    Answer: Functional testing verifies the software's functionality, while non-functional testing checks aspects like performance, security, and usability.
66. **What is boundary value analysis in testing?**
    Answer: Boundary value analysis tests the software at the boundaries of input ranges, where errors are more likely to occur.
67. **What is equivalence partitioning in testing?**
    Answer: Equivalence partitioning divides input data into equivalent classes, assuming that all data within a class will be treated the same.
68. **What is a test plan?**
    Answer: A test plan is a document outlining the testing strategy, objectives, resources, schedule, and scope for testing a software application.
69. **What is a test case?**
    Answer: A test case is a set of conditions or steps used to verify a particular feature or functionality in the software.
70. **What is a bug/defect?**
    Answer: A bug or defect is a flaw or issue in the software that prevents it from functioning as expected.
71. **What is debugging?**
    Answer: Debugging is the process of identifying, isolating, and fixing bugs in the software.
72. **What is a test suite?**
    Answer: A test suite is a collection of test cases designed to test specific functionalities or features of a software application.
73. **What is the role of a Quality Assurance (QA) engineer?**
    Answer: A QA engineer is responsible for ensuring the software meets quality standards, performing testing, and identifying areas for improvement.

74. **What is exploratory testing?**
Answer: Exploratory testing involves simultaneously learning about the software, designing tests, and executing them, often in an ad-hoc and informal manner.
75. **What is a mock object in testing?**
Answer: A mock object is a simulated object used in testing to mimic the behavior of real objects in controlled ways.
76. **What is the difference between system and acceptance testing?**
Answer: System testing evaluates the entire system, while acceptance testing ensures the system meets business requirements and is acceptable to the client.

Databases and SQL

77. **What is a database?**
Answer: A database is an organized collection of structured information or data, typically stored and accessed electronically.
78. **What is DBMS (Database Management System)?**
Answer: DBMS is software that provides an interface for users to interact with databases, manage data, and ensure its integrity.
79. **What is the difference between DBMS and RDBMS?**
Answer: DBMS is a general database system, while RDBMS (Relational DBMS) stores data in tables and enforces relationships between them using primary and foreign keys.
80. **What are the different types of DBMS models?**
Answer: The main DBMS models are:

- Hierarchical model
- Network model
- Relational model
- Object-oriented model

81. **What is SQL?**
Answer: SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases.
82. **What are the types of SQL commands?**
Answer: SQL commands are categorized into:

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
- DCL (Data Control Language)
- TCL (Transaction Control Language)

83. **What is a primary key?**
Answer: A primary key is a unique identifier for each record in a database table, ensuring that no two rows have the same key.

84. **What is a foreign key?**
    Answer: A foreign key is a field in one table that uniquely identifies a row of another table, establishing a relationship between them.
85. **What is a unique key?**
    Answer: A unique key ensures that all values in a column or set of columns are distinct, preventing duplicate entries.
86. **What is normalization?**
    Answer: Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity.
87. **What are the different normal forms?**
    Answer: The normal forms are:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)

88. **What is denormalization?**
    Answer: Denormalization is the process of combining normalized tables to improve query performance at the expense of potential data redundancy.
89. **What is a join in SQL?**
    Answer: A join combines rows from two or more tables based on a related column, allowing data to be queried across tables.
90. **What are the different types of joins?**
    Answer: The main types of joins are:

- INNER JOIN
- LEFT JOIN (OUTER)
- RIGHT JOIN (OUTER)
- FULL JOIN (OUTER)

91. **What is the difference between UNION and UNION ALL in SQL?**
    Answer: UNION combines results from multiple queries, removing duplicates, while UNION ALL includes all records, even duplicates.
92. **What is a subquery in SQL?**
    Answer: A subquery is a query nested inside another query, typically used to retrieve data for use in the outer query.
93. **What is indexing in SQL?**
    Answer: Indexing improves query performance by creating a data structure that allows faster search and retrieval of records.
94. **What is a transaction in SQL?**
    Answer: A transaction is a sequence of one or more SQL operations that are executed as a single unit, ensuring consistency and reliability.
95. **What are ACID properties in a database?**
    Answer: ACID stands for:

- Atomicity

- Consistency
- Isolation
- Durability

96. **What is SQL injection?**
    Answer: SQL injection is a security vulnerability that allows an attacker to execute arbitrary SQL code by manipulating input data.
97. **What is a trigger in SQL?**
    Answer: A trigger is a set of SQL instructions that automatically execute in response to certain events on a table or view.
98. **What is a view in SQL?**
    Answer: A view is a virtual table based on the result of a SQL query, used to simplify complex queries and enhance security.
99. **What is a stored procedure in SQL?**
    Answer: A stored procedure is a precompiled collection of SQL statements that can be executed as a unit, improving performance and reusability.
100. **What is the difference between DELETE and TRUNCATE commands in SQL?**
    Answer: DELETE removes rows from a table based on a condition and can be rolled back, while TRUNCATE removes all rows and cannot be rolled back.

Version Control and Git

101. **What is version control?**
    Answer: Version control is a system that manages changes to code or documents, allowing multiple versions to be tracked and modified over time.
102. **What is Git?**
    Answer: Git is a distributed version control system that helps track changes to files, collaborate with others, and maintain project history.
103. **What is the difference between Git and GitHub?**
    Answer: Git is a version control system, while GitHub is a platform that hosts Git repositories and provides collaboration tools for developers.
104. **What is a commit in Git?**
    Answer: A commit in Git is a snapshot of the changes made to files in the repository, marking a point in the project's history.
105. **What is a branch in Git?**
    Answer: A branch in Git is a separate line of development that allows developers to work on features or fixes without affecting the main codebase.
106. **What is the purpose of the "merge" command in Git?**
    Answer: The merge command combines changes from two different branches into a single branch.
107. **What is a pull request in GitHub?**
    Answer: A pull request is a way to propose changes to a repository, allowing others to review and discuss the changes before merging them into the main branch.

108. **What is the difference between "git pull" and "git fetch"?**
Answer: "git fetch" retrieves updates from the remote repository without applying them, while "git pull" retrieves and merges the changes.

109. **What is a conflict in Git?**
Answer: A conflict occurs when Git is unable to automatically merge changes from different branches, requiring manual intervention to resolve.

110. **What is the difference between "git clone" and "git init"?**
Answer: "git clone" creates a copy of an existing repository, while "git init" initializes a new repository in the current directory.

Software Architecture and Design

111. **What is software architecture?**
Answer: Software architecture is the high-level structure of a software system, including its components, their relationships, and how they interact.

112. **What is a layered architecture?**
Answer: A layered architecture divides a system into distinct layers, each responsible for a specific function (e.g., presentation, business logic, data access).

113. **What is microservices architecture?**
Answer: Microservices architecture structures an application as a collection of loosely coupled, independently deployable services that communicate via APIs.

114. **What is monolithic architecture?**
Answer: Monolithic architecture is a traditional approach where all components of an application are tightly integrated and run as a single unit.

115. **What is the MVC design pattern?**
Answer: The MVC (Model-View-Controller) design pattern separates an application into three components:

- Model (data and business logic)
- View (user interface)
- Controller (handles input and updates the model and view)

116. **What is the Singleton design pattern?**
Answer: The Singleton pattern ensures that a class has only one instance and provides a global point of access to that instance.

117. **What is the Factory design pattern?**
Answer: The Factory pattern provides an interface for creating objects, but allows subclasses to alter the type of objects created.

118. **What is the Observer design pattern?**
Answer: The Observer pattern defines a one-to-many dependency, where changes to one object automatically notify and update all dependent objects.

119. **What is the Strategy design pattern?**
Answer: The Strategy pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable, allowing the algorithm to be chosen at runtime.

120. **What is Dependency Injection in software design?**
Answer: Dependency Injection is a design pattern where an object's dependencies are provided from the outside, rather than the object creating them internally.

Project Management

121. **What is project management in software engineering?**
Answer: Project management in software engineering involves planning, organizing, and overseeing the execution of software projects to ensure timely delivery, within budget, and meeting requirements.

122. **What is the role of a project manager in software development?**
Answer: A project manager is responsible for planning, executing, and closing projects, managing resources, risks, and stakeholders, ensuring project success.

123. **What are the main phases of project management?**
Answer: The phases of project management are:

- Initiation
- Planning
- Execution
- Monitoring and Control
- Closing

124. **What is a Gantt chart?**
Answer: A Gantt chart is a visual representation of a project schedule, showing tasks, their duration, and dependencies over time.

125. **What is Agile project management?**
Answer: Agile project management focuses on delivering small, incremental improvements to a product, emphasizing collaboration, flexibility, and customer feedback.

126. **What is Scrum?**
Answer: Scrum is an Agile framework for managing and completing complex projects, dividing work into small iterations called sprints, with defined roles such as Scrum Master and Product Owner.

127. **What is Kanban?**
Answer: Kanban is a visual workflow management method that helps teams visualize their work, limit work in progress, and improve efficiency.

128. **What is the difference between Agile and Waterfall methodologies?**
Answer: Agile is iterative, flexible, and focuses on customer collaboration, while Waterfall is linear, sequential, and follows a fixed process.

129. **What is risk management in software engineering?**
Answer: Risk management involves identifying, assessing, and mitigating potential risks to ensure the successful delivery of a software project.

130. **What is the significance of a project scope?**
Answer: Project scope defines the boundaries of the project, including deliverables,

timelines, and resources, helping to ensure project goals are clearly understood and achieved.