Repeated Questions:

1. **What is bug prevention?** (Repeated 2 times)
2. **What is the goal of software testing?** (Repeated 2 times)
3. **Define transaction.** (Repeated 2 times)
4. **What is Interface testing?** (Repeated 2 times)
5. **What is debugging?** (Repeated 2 times)
6. **What is meant by a bug?** (Repeated 2 times)
7. **Define domain.** (Repeated 2 times)
8. **What is data flow graph?** (Repeated 2 times)
9. **What is transition testing?** (Repeated 2 times)
10. **Define decision tables.** (Repeated 2 times)
11. **Define state testing.** (Repeated 2 times)
12. **What is path expression?** (Repeated 2 times)
13. **What is linguistics metrics?** (Repeated 2 times)

Repeated Questions

1. **What is bug prevention?**
    - Proactively identifying potential error sources.
    - Implementing preventive measures like code reviews.
    - Using tools to enforce coding standards.
    - Providing training to developers on best practices.
2. **What is the goal of software testing?**
    - Identify and fix defects in the software.
    - Ensure the software meets requirements.
    - Verify that the software functions as intended.
    - Improve software quality and reliability.
3. **Define transaction.**
    - A sequence of operations performed as a single logical unit.
    - Ensures data integrity during concurrent operations.
    - Must follow ACID properties (Atomicity, Consistency, Isolation, Durability).
    - Examples include database updates or online purchases.
4. **What is Interface testing?**
    - Testing interactions between software components or systems.
    - Ensures data transfer between modules is correct.
    - Validates input and output consistency.
    - Detects issues like integration gaps or communication errors.
5. **What is debugging?**
    - The process of locating and fixing software defects.
    - Involves analyzing the code and identifying errors.
    - Often performed after a failed test case.
    - Uses tools like debuggers for step-by-step code execution.
6. **What is meant by a bug?**
    - A defect or flaw in software causing unexpected results.

- Results from coding errors or misinterpretation of requirements.
- Can cause crashes, incorrect outputs, or security vulnerabilities.
- Requires debugging to resolve.

7. **Define domain.**
    - A specific area of knowledge or activity for a system.
    - Defines the scope and boundaries of the software.
    - Includes user requirements and business logic.
    - Examples: healthcare, e-commerce, banking domains.
8. **What is data flow graph?**
    - A graphical representation of data flow in a program.
    - Nodes represent operations or functions.
    - Edges represent data dependencies between operations.
    - Used to identify data usage and potential anomalies.
9. **What is transition testing?**
    - Tests state changes in software.
    - Focuses on system behavior when moving between states.
    - Validates state diagrams or state machines.
    - Ensures smooth transitions without unexpected outcomes.
10. **Define decision tables.**
    - A tabular method for representing decisions and rules.
    - Includes conditions, actions, and rules.
    - Ensures all possible scenarios are considered.
    - Simplifies complex logic for testing and validation.
11. **Define state testing.**
    - A testing technique for state-based systems.
    - Validates transitions and behaviors in different states.
    - Ensures system operates correctly in all states.
    - Uses state diagrams for coverage and completeness.
12. **What is path expression?**

- A representation of all possible execution paths in a program.
- Helps identify independent paths for testing.
- Derived from control flow or flow graphs.
- Useful for path-based test case generation.

13. **What is linguistics metrics?**
    - Metrics derived from natural language processing in requirements.
    - Analyze textual descriptions for clarity and consistency.
    - Identify ambiguities or incomplete specifications.
    - Weakness: Highly dependent on the quality of input text.

Unique Questions

1. **Name the types of bugs.**
    - Logical bugs.
    - Syntax bugs.
    - Performance bugs.
    - Security bugs.
2. **What is a flow of graphs?**
    - Representation of program control or data flow.
    - Nodes represent statements or conditions.
    - Edges represent the flow of control or data.
    - Used for path testing and analysis.
3. **What is the weakness of linguist metrics?**
    - Cannot fully understand complex domain-specific terms.
    - Relies on high-quality textual input.
    - Limited in identifying deeply hidden ambiguities.
    - Requires human intervention for context-specific issues.
4. **What is stable state?**

- A system condition where no further changes occur.
- All processes complete successfully without errors.
- No pending transitions or actions.
- Reflects system equilibrium or readiness.

5. **What are the components of decision tables?**
    - Conditions: Define inputs or scenarios.
    - Actions: Define outputs or decisions.
    - Rules: Map conditions to actions.
    - Entries: Specify conditions and actions for each rule.
6. **What is state graph?**
    - A diagram showing states and transitions of a system.
    - Nodes represent states.
    - Edges represent transitions between states.
    - Used in state-based testing and design validation.
7. **Write down any two principles of state testing.**
    - Test all possible transitions between states.
    - Validate outputs for each state and transition.
8. **Define verification.**
    - Ensures software meets specified requirements.
    - Involves static techniques like reviews and walkthroughs.
    - Confirms correct implementation of design.
    - Answers "Are we building the product right?"
9. **How is data flow measured?**
    - Analyzing variable usage in code.
    - Tracking definitions, uses, and lifetimes of data.
    - Identifying anomalies like unused variables or uninitialized data.
    - Using tools to automate data flow analysis.
10. **Define metrics.**
    - Quantitative measures of software quality or performance.

- Examples include code coverage, defect density.
- Help track progress and evaluate effectiveness.
- Used for process improvement and decision-making.

(Continued...)

11. **Name any four software testing tools.**
    - Selenium.
    - JUnit.
    - TestNG.
    - Postman.
12. **Define states.**
    - Conditions of a system or component at a specific time.
    - Defined by the current values of variables and outputs.
    - Change occurs due to events or inputs.
    - Examples: Ready, Processing, Completed.
13. **Give any two applications of decision tables.**
    - Representing complex business rules for testing.
    - Designing test cases for all possible scenarios.
14. **What are decision tables?**
    - A structured format to represent conditions and actions.
    - Helps identify missing conditions or logic.
    - Ensures comprehensive test coverage.
    - Simplifies validation of decision-based logic.
15. **What is the purpose of debugging?**
    - Identify and fix software defects.
    - Validate correctness of logic and algorithms.
    - Enhance software stability and performance.
    - Prevent similar errors in future development.
16. **What is path testing?**

- A testing technique to cover independent paths in a program.
- Derived from control flow graphs.
- Focuses on decision outcomes and loops.
- Ensures all paths are tested at least once.

17. **Define software quality.**
    - Degree to which software meets requirements.
    - Includes attributes like reliability, maintainability, and usability.
    - Ensures the product delivers value to users.
    - Measured using standards and metrics.
18. **Define data flow testing.**
    - Testing technique based on variable usage.
    - Identifies anomalies in data definitions and uses.
    - Focuses on paths where data is manipulated.
    - Ensures data is used correctly throughout the program.
19. **What is domain testing?**
    - Testing inputs and outputs within specific ranges.
    - Validates boundary and edge conditions.
    - Ensures software handles all domain-defined scenarios.
    - Helps identify input-related defects.
20. **Write the objectives of syntax testing.**
    - Validate adherence to language syntax rules.
    - Detect syntactical errors in inputs or code.
    - Ensure correct processing of valid inputs.
    - Identify handling of invalid inputs.
21. **Define path expressions.**
    - Mathematical representation of all paths in a program.
    - Used for coverage analysis and test case generation.
    - Combines control flow and decision-making.
    - Helps ensure path-wise correctness.

22. **What do you mean by state graph?**
    - A visual model of states and transitions.
    - Represents system behavior over time.
    - Aids in designing and testing state-based systems.
    - Identifies unreachable or invalid states.
23. **What are processing bugs?**
    - Errors in logic or calculations within a program.
    - Examples include incorrect algorithms or data handling.
    - Impact program outputs and functionality.
    - Common in complex computations or business logic.
24. **Define flow graph.**
    - Graphical representation of control flow in a program.
    - Nodes represent statements or blocks of code.
    - Edges represent the control flow between nodes.
    - Used for path testing and control flow analysis.
25. **What is meant by path instrumentation?**
    - Technique to track execution of specific paths in code.
    - Uses tools or code instrumentation to monitor paths.
    - Helps identify untested or partially tested paths.
    - Useful for dynamic testing and coverage analysis.
26. **List the strategies for data flow testing.**
    - All-definitions strategy: Test all variable definitions.
    - All-uses strategy: Test all usages of variables.
    - All-du-paths strategy: Test all definition-use paths.
    - Subset strategies for efficiency and critical paths.
27. **Define test cases.**
    - A set of inputs, execution conditions, and expected results.
    - Designed to validate specific functionality or requirements.
    - Ensures system behaves as intended under defined scenarios.

- Documented for reuse and traceability.

28. **What is software quality?**
    - A measure of how well software meets customer expectations.
    - Includes functional, performance, and security aspects.
    - Assessed through testing and user feedback.
    - Ensures product reliability and efficiency.
29. **What is flow chart?**
    - A graphical representation of a process or algorithm.
    - Uses symbols like rectangles (process) and diamonds (decision).
    - Helps visualize program logic and workflow.
    - Useful for planning, debugging, and documentation.
30. **Define simple path segment.**
    - A sequence of program nodes without loops.
    - Represents a linear control flow.
    - Used for straightforward path testing.
    - Ensures coverage of non-repeating program segments.
31. **What do you mean by transition?**
    - Movement from one state to another in a system.
    - Triggered by events or conditions.
    - Defined by state diagrams or tables.
    - Essential for testing dynamic system behaviors.