Faculty Name: TRUPTI DESHKAR **Year:** III **Sem:** I

Course: Software Testing Methodologies **Date:** 10-07-2025

UNIT-I

Introduction Of Software Testing Methodologies

❖ **1-Mark Questions**

S. No	Question	Bloom's Level
1.	Define software testing.	L1
2.	What is a bug?	L1
3.	What do you mean by "consequences of bugs"?	L2
4.	What is meant by path predicate?	L1
5.	List any two dichotomies in software testing.	L2
6.	What is the purpose of path instrumentation?	L2
7.	Define achievable path.	L1
8.	What is a flow graph?	L1
9.	What is taxonomy of bugs?	L2
10.	Define path sensitizing.	L1

❖ **5-Mark Questions and Aboveand above**

S. No	Question	Bloom's Level
1.	Explain the purpose of testing in software development.	L3
2.	Describe any 3 dichotomies used in software testing with examples.	L3

S. No	Question	Bloom's Level
3.	Analyze the different types of bugs based on their taxonomy.	L4
4.	Construct a simple flow graph and explain its nodes and edges.	L3
5.	Explain how path predicates and path sensitizing are used in testing.	L4
6.	Apply path instrumentation on a simple code snippet and explain the result.	L3
7.	Differentiate between basic and achievable paths in flow graph testing.	L4
8.	Discuss any two models for testing and their real-world application.	L3
9.	Analyze the impact of severe bugs in real-time systems.	L4
10.	Illustrate the steps involved in applying path testing.	L3
11.	Construct a flow graph for a login system and identify all path predicates, achievable paths, and sensitization steps.	L6
12.	Evaluate the role of path testing in improving software quality and compare it with other testing techniques.	L5
13.	Develop a bug taxonomy for a web application, categorizing bugs based on impact and origin.	L6
14.	Assess the importance of understanding consequences of bugs in safety-critical systems like healthcare or aviation.	L5
15.	Design a complete path testing approach for a function that processes banking transactions.	L6
16.	Critically analyze different models for testing and recommend one for agile environments with justification.	L5
17.	Create a detailed test case suite using flow graphs and path predicates for a student grading system.	L6
18.	Justify the need for modeling and instrumentation in path testing through a real-world example.	L5
19.	Create and explain a flow graph for a traffic light control system using all concepts of path testing.	L6

S. No	Question	Bloom's Level
20.	Examine the limitations of taxonomy-based bug classification and propose improvements.	L5

UNIT-II

Transaction Flow Testing

❖ **1-Mark Questions**

S.No	Question	Bloom's Level
1.	What is transaction flow testing?	L1
2.	Define a transaction.	L1
3.	What is a transaction flow graph?	L1
4.	What is the purpose of transaction flow testing?	L2
5.	Define dataflow testing.	L1
6.	What is a definition-use (DU) pair in dataflow testing?	L1
7.	Name any one strategy used in dataflow testing.	L1
8.	What is a def-clear path?	L2
9.	What kind of bugs can be detected using dataflow testing?	L2
10.	What is the difference between a predicate and a path predicate?	L2

◈ **5-Mark Questions and Above**

S.No	Question	Bloom's Level
1.	Explain transaction flow testing with the help of a sample flow graph.	L2
2.	Describe the techniques used in transaction flow testing.	L3
3.	Write short notes on path sensitizing and its importance in path testing.	L3
4.	Discuss the role of predicates and path predicates in transaction flow testing.	L3
5.	Compare control flow testing and transaction flow testing with examples.	L4
6.	Explain the basic concept of dataflow testing.	L2

S.No	Question	Bloom's Level
7.	Describe the main dataflow testing strategies with examples.	L3
8.	Identify different types of anomalies detected using dataflow testing.	L4
9.	Explain how path instrumentation is used in path testing.	L3
10.	Analyze the effectiveness of transaction flow testing in detecting logical errors.	L4
11.	Construct a transaction flow graph for a library management system and perform transaction flow testing.	L6
12.	Design a transaction flow for an online shopping cart and explain how you would apply transaction testing.	L6
13.	Develop a sample program and identify all DU paths using dataflow testing.	L6
14.	Create a comprehensive test plan using dataflow testing for a student registration module.	L6
15.	Evaluate the advantages and disadvantages of using dataflow testing in software maintenance.	L5
16.	Analyze a case study where transaction flow testing was applied. What were the outcomes?	L5
17.	Write a detailed comparison of path testing, control flow testing, and dataflow testing.	L5
18.	Explain the application of dataflow testing in real-time systems with examples.	L5
19.	Design a test strategy using both transaction flow and dataflow testing for a payment gateway.	L6
20.	Identify challenges in applying dataflow testing to large-scale enterprise applications and suggest solutions.	L6

UNIT-III

Domain Testing

❖ **1 Mark Questions**

S. No.	Question	Bloom's Level
1.	Define domain testing.	L1
2.	What is a domain in software testing?	L1
3.	What are "nice" domains?	L2
4.	What are "ugly" domains?	L2
5.	Define interface testing.	L1
6.	What is the meaning of testability?	L1
7.	Name two methods used in domain testing.	L1
8.	Mention one goal of domain testing.	L2
9.	Write an example of an input domain.	L2
10.	What is the relationship between domains and paths?	L2

❖ **5 Mark Questions and Above**

S. No.	Question	Bloom's Level
1.	Explain the characteristics of nice and ugly domains with an example.	L3
2.	Describe the role of domain and interface testing in quality assurance.	L3
3.	Discuss how path coverage is related to domain testing.	L4
4.	Differentiate between domain testing and boundary value analysis.	L4
5.	Illustrate how invalid inputs can be identified using domain testing.	L3
6.	Analyze the importance of testability in domain testing.	L4

S. No.	Question	Bloom's Level
7.	Explain domains and paths in context of decision tables.	L3
8.	Why is it important to test domain boundaries? Give examples.	L4
9.	Write short notes on "domains and testability" with practical insights.	L3
10.	Compare and contrast interface testing and domain testing.	L4
11.	Design a domain-based test plan for a flight reservation system.	L6
12.	Evaluate the effectiveness of domain testing in reducing defects in large-scale applications.	L5
13.	Create a strategy to handle ugly domains and complex interfaces in a banking software application.	L6
14.	Assess the limitations of domain testing. Suggest improvements or alternatives.	L5
15.	Develop a domain testing framework for a web-based student result management system.	L6
16.	Analyze the interrelation between domains, paths, and testability in mission-critical systems.	L5
17.	Design test cases to handle both valid and invalid domains in a login system.	L6
18.	Critically evaluate how domain and interface testing detect boundary and integration errors.	L5
19.	Develop a detailed test plan using domain testing for a medical diagnostic tool.	L6
20.	Analyze a real-world system failure that could have been avoided using proper domain testing.	L5

UNIT-IV

Paths, Path Products and Regular Expressions

❖ **1 Mark Questions**

S. No.	Question	Bloom's Level
1.	Define path in software testing.	L1
2.	What is a path product?	L1
3.	What is the purpose of the reduction procedure?	L1
4.	Define regular expression.	L1
5.	What is flow anomaly?	L2
6.	Name one application of regular expressions in software testing.	L1
7.	What is the role of decision tables in logic-based testing?	L2
8.	What is a KV chart used for?	L2
9.	Define logic-based testing.	L1
10.	What is a specification in software testing?	L2

❖ **5 Mark Questions and Above**

S. No.	Question	Bloom's Level
1.	Explain the concept of path expressions with an example.	L3
2.	Describe the reduction procedure in path testing with steps.	L3
3.	Differentiate between path products and path expressions.	L4
4.	Illustrate how regular expressions are used to detect flow anomalies.	L4
5.	Apply logic-based testing to a simple decision-making scenario using a decision table.	L3

S. No.	Question	Bloom's Level
6.	Draw and explain a basic KV chart.	L3
7.	Describe any two real-world applications of path testing.	L4
8.	How can logic-based testing help improve the accuracy of specifications?	L4
9.	Apply path expression techniques to represent simple program flow.	L3
10.	Explain how decision tables are used in logic-based testing.	L4
11.	Design a test case using path products for a program that contains nested decision statements.	L6
12.	Evaluate the effectiveness of reduction procedures in simplifying control flow graphs.	L5
13.	Develop a complex regular expression and explain how it detects multiple flow anomalies.	L6
14.	Critically evaluate the use of decision tables in logic-based testing.	L5
15.	Create a KV chart for a sample logic condition and interpret the result.	L6
16.	Analyze a real-world software bug that could have been prevented using logic-based testing techniques.	L5
17.	Generate path expressions from a complex flowchart and simplify them using reduction rules.	L6
18.	Compare the utility of path expressions and regular expressions in representing execution paths.	L5
19.	Design a logic-based testing framework for a decision support system.	L6
20.	Evaluate the limitations of KV charts and propose enhancements for modern systems.	L5

UNIT-V

State, State Graphs and Transition Testing

❖ 1-Mark Questions

S. No.	Question	Bloom's Level
1.	Define a state in state transition testing.	L1
2.	What is a state graph?	L1
3.	What is meant by a good state graph?	L2
4.	What defines a bad state graph?	L2
5.	Name one objective of state testing.	L1
6.	What is the role of testability in state-based testing?	L2
7.	What is a graph matrix?	L1
8.	What is the power of a matrix in graph theory?	L1
9.	What is the use of the node reduction algorithm?	L2
10.	Give one example of a transition in a state graph.	L2

❖ 5-Mark Questions and Above

S. No.	Question	Bloom's Level
1.	Explain with example the components of a state graph.	L3
2.	Draw and identify characteristics of a good and bad state graph.	L3
3.	Describe the process of state testing with a simple example.	L3
4.	List and explain any three testability tips relevant to transition testing.	L4
5.	Construct a graph matrix for a given small state diagram.	L4

S. No.	Question	Bloom's Level
6.	Apply the node reduction algorithm on a sample state graph and show the steps.	L4
7.	Differentiate between state testing and transition testing.	L4
8.	Explain the motivational overview behind using graph matrices in testing.	L3
9.	Analyze how the power of a matrix helps determine path length in a graph.	L4
10.	Give examples of testing tools that support state graph modeling.	L3
11.	Create a state graph for a user login system and identify potential flaws.	L6
12.	Evaluate the importance of distinguishing good vs bad state graphs in software test design.	L5
13.	Design a test suite based on a given state transition diagram.	L6
14.	Critically analyze how graph matrices assist in visualizing and optimizing state transitions.	L5
15.	Use the node reduction algorithm to find the minimum path for a complex state graph.	L6
16.	Evaluate the effectiveness of testability tips in reducing redundant transitions in state testing.	L5
17.	Generate graph matrices and calculate their power to interpret system behavior paths.	L6
18.	Develop a model showing how building tools utilize graph-based logic in test automation.	L6
19.	Discuss and evaluate challenges in implementing transition testing in real-time systems.	L5
20.	Design and simulate a test case using state transition diagrams for a vending machine.	L6