IT8076 SOFTWARE TESTING L T P C 3 0 0 3

OBJECTIVES

The student should be made to:

- Expose the criteria for test cases
- Learn the design of test cases
- Be familiar with test management and test automation techniques
- Be exposed to test metrics and measurements

UNIT I INTRODUCTION 9

Testing as an Engineering Activity – Testing as a Process – Testing axioms – Basic definitions – Software Testing Principles – The Tester‘s Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design – Defect Examples – Developer/Tester Support of Developing a Defect Repository – Defect Prevention strategies.

UNIT II TEST CASE DESIGN 9

Test case Design Strategies – Using Black Bod Approach to Test Case Design – Random Testing – Requirements based testing – Boundary Value Analysis – Equivalence Class Partitioning – State-based testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Evaluating Test Adequacy Criteria.

UNIT III LEVELS OF TESTING 9

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing – Compatibility testing – Testing the documentation – Website testing.

UNIT IV TEST MANAGEMENT 9

People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – Test management – Test process – Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group.

UNIT V TEST AUTOMATION 9

Software test automation – skill needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.

TOTAL: 45 PERIODS

OUTCOMES:

At the end of the course the students will be able to

- Design test cases suitable for a software development for different domains.
- Identify suitable tests to be carried out
- Prepare test planning based on the document
- Document test plans and test cases designed
- Use of automatic testing tools.
- Develop and validate a test plan.

TEXT BOOKS:

1. Srinivasan Desikan and Gopalaswamy Ramesh, Software Testing – Principles and Practices, Pearson Education, 2006.
2. Ron Patton, Software Testing, Second Edition, Sams Publishing, Pearson Education, 2007.

REFERENCES:

1. Ilene Burnstein, Practical Software Testing, Springer International Edition, 2003.
2. Edward Kit, Software Testing in the Real World – Improving the Process, Pearson Education, 1995.
3. Boris Beizer, Software Testing Techniques – 2nd Edition, Van Nostrand Reinhold, New York, 1990.
4. Aditya P. Mathur, Foundations of Software Testing _ Fundamental Algorithms and Techniques, Dorling Kindersley(India) Pvt. Ltd., Pearson Education, 2008.

COURSE OUTCOMES:

On completion of the course, students will be able to

CO1	Understand fundamental concepts in software testing, origins of defects and prevention strategies.
CO2	Apply test case strategies and evaluate test criteria.
CO3	Analyze various levels of testing such as integration, regression, alpha, beta and website testing.
CO4	Understand the various roles in test planning and policy development.
CO5	Understand software test automation problems and solutions.
CO6	Create and validate a test plan

UNIT- I

INTRODUCTION

Testing as an Engineering Activity – Testing as a Process – Testing axioms – Basic definitions – Software Testing Principles – The Tester"s Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design – Defect Examples – Developer/Tester Support of Developing a Defect Repository – Defect Prevention strategies.

PART A

1. Define Software Engineering. (R) (MAY 2012)

Software Engineering is a discipline that produces error free software with in a time and budget.

2. Define Software Testing. (R) (DEC 2012), (May/June 2014)

Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.

3. List the elements of the Engineering disciplines. (R) (MAY 2012, Dec 2014)

- Basic principles
- Processes
- Standards
- Measurements
- Tools
- Methods
- Best practices
- Code of ethics
- Body of knowledge

4. Differentiate between Verification and Validation? (AN) (U.Q Nov/Dec 2009)

S.NO	Verification	Validation
1	Verification is the process of evaluating software system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.	Verification is the process of evaluating software system or component during or at the end of the, the development phase satisfy the conditions imposed at the start of that phase.
2	Verification is usually associated with activities such as inspections and reviews of the s/w deliverables.	Verification is usually associated with Traditional execution based testing i.e., Exercising the code with test cases.

5. Define the term Testing. (R) (DEC 2012)

- Testing is generally described as a group of procedures carried out to evaluate some aspect of a piece of software.
- Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.

6. Differentiate between testing and debugging. (AN) (U.Q Nov/Dec 2008)

Testing	Debugging
Testing as a dual purpose process Reveal defects and to evaluate quality attributes.	Debugging or fault localization is the process of Locating the fault or defect, Repairing the code, and Retesting the code.

7. Define process in the context of software quality. (R) (Nov/Dec 2009)

Process, in the software engineering domain, is a set of methods, practices, Standards, documents, activities, polices, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code, and manuals.

8. Define the term Debugging or fault localization. (R) (Nov 2011)

Debugging or fault localization is the process of Locating the fault or defect Repairing the code and Retesting the code.

9. List the levels of TMM(Testing Maturity Model). (R)(Apr-May 2018)

The testing maturity model or TMM contains five levels. They are

Level 1: Initial

Level 2: Phase definition

Level 3: Integration

Level 4: Management and Measurement

Level 5: Optimization /Defect prevention and Quality Control

10. List the members of the critical groups in a testing process (R) (Nov/Dec 2008)

- Manager
- Developer/Tester
- User/Client

11. Define Error. (R) (Nov 2011)

An error is mistake or misconception or misunderstanding on the part of a software developer.

12. Define Faults (Defects)., (R) (May/June 2014)

A fault is introduced into the software as the result of an error. It is an anomaly in the software that may cause nit to behave incorrectly, and not according to its specification.

13. Define Failures. (R) (MAY 2012 & May/June 2014)

A failure is the inability of a software or component to perform its required functions within specified performance requirements.

14. Define Test Cases. (R)

- A test case in a practical sense is attest related item which contains the following information.
- A set of test inputs. These are data items received from an external source by the code under test. The external source can be hardware, software, or human.
- Execution conditions. These are conditions required for running the test, for example, a certain state of a database, or a configuration of a hardware device.
- Expected outputs. These are the specified results to be produced by the code under test.

15. Distinguish between fault and failure. (AN) (May/June 2009)

Fault	Failure
A fault is introduced into the software as the result of an error. It is an anomaly in the software that may cause nit to behave incorrectly, and not according to its specification.	A failure is the inability of a software or componentto perform its required functions within specified performance requirements.

16. Write short notes on Test, Test Set, and Test Suite (U).

- A Test is a group of related test cases, or a group of related test cases and test procedure.
- A group of related test is sometimes referred to as a test set.
- A group of related tests that are associated with a database, and are usually run together, is sometimes referred to as a Test Suite.

17. Define Test Oracle. (R) (April/May 2013 & April/May 2015) (Apr-May 2018)

Test Oracle is a document, or a piece of software that allows tester to determine whether a test has been passed or failed.

18. Define Test Bed. (R) (April/May 2013 & April/May 2015)(Nov/Dec 2017)

A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.

19. Define Software Quality. (R)

- Quality relates to the degree to which a system, system component, or process meets specified requirements.
- Quality relates to the degree to which a system, system component, or process meets Customer or user needs, or expectations.

20. List the Quality Attributes. (R)

- Correctness
- Reliability
- Usability
- Integrity
- Portability
- Maintainability
- Interoperability

21. Define SQA group. (R)

The software quality assurance (SQA) group is a team of people with the necessary training and skills to ensure that all necessary actions are taken during the development process so that the resulting software confirms to established technical requirements.

22. Explain the work of SQA group. (U)

Testers to develop quality related policies and quality assurance plans for each project.

The group is also involved in measurement collection and analysis, record keeping, and Reporting. The SQA team members participate in reviews and audits, record and track Problems, and verify that corrections have been made.

23. Define reviews. (R)

A review is a group meeting whose purpose is to evaluate a software artifact or a set of Software artifacts. Review and audit is usually conducted by a SQA group.

24. List the sources of Defects or Origins of defects. Or list the classification of defect (May/June 2009) (R)

- Education
- Communication
- Oversight
- Transcription
- Process

25. Programmer A and Programmer B are working on a group of interfacing modules. Programmer A tends to be a poor communicator and does not get along well with Programmer B. Due to this situation, what types of defects are likely to surface in these interfacing modules?(AN)

Communication defects.

26. What is the role of process in software quality?(U) (Nov/Dec 2013)

Software process is a set of activities and associated results which produces a software product. These activities are mostly carried out by software engineers. There are four fundamental process activities which are common to all software processes. These activities are software specification, software development, software validation, software evolution.

27. What do you mean by software defect? Write one example. (U) (Nov/Dec 2013)

A software defect is a deficiency in a software product that causes it to perform unexpectedly. From a software user‘s perspective, a defect is anything that causes the software not to meet their expectations. In this context, a software user can be either a person or another piece of software.

Example:

User Expectation:	The software will help me accomplish a task
Software Defect:	Desired software functionality is missing

28. Why We Do Debugging? (U) (April/May 2013)

Debugging is done when the errors are found in testing process and those errors will be eliminated in debugging.

29. Mention the quality attributes of software (R) (April/May 2015)

- Correctness
- Reliability
- Adequacy
- Learnability
- Robustness
- Maintainability
- Readability
- Extensibility
- Testability
- Efficiency
- Portability

30. **Mention the objectives of Software Testing U) (Nov/Dec 2016), (May/June 2016)(Nov/Dec 2017)**

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

31. **Define defects with an example.(U). (Nov/Dec 2016) (Apr/May 2019)**
A **defect is an error or a bug**, in the application which is created. A programmer while designing and building the software can make mistakes or error. These mistakes or errors mean that there are flaws in the software. These are called defects.

Examples:

Scenario 1: The software will allow a user to make online payments using a debit card.
Defect: The option of selecting a debit card for making payments is missing.
Scenario 2: The software will help me in avoiding spelling mistakes.
Defect: The feature for detecting the spelling error is missing.

32. **Differentiate Error, defect and Failure. (ANALYZE) (May/June 2016)**

ERROR: An error is a mistake, misconception, or misunderstanding on the part of a software developer. In the category of developer we include software engineers, programmers, analysts, and testers. For example, a developer may misunderstand a de-sign notation, or a programmer might type a variable name incorrectly – leads to an Error. It is the one which is generated because of wrong login, loop or due to syntax. Error normally arises in software; it leads to change the functionality of the program.

DEFECT: It can be simply defined as a variance between expected and actual. Defect is an error found AFTER the application goes into production. It commonly refers to several troubles with the software products, with its external behavior or with its internal features. In other words Defect is the difference between expected and actual result in the context of testing. It is the deviation of the customer requirement.

FAILURE: A failure is the inability of a software system or component to perform its required functions within specified performance requirements. When a defect reaches the end customer it is called a Failure. During development Failures are usually observed by testers.

33. **Mention the role of test engineer in software development organization (U) (APR/MAY 2017)**

Test engineers/QA testers/QC testers are responsible for:

- To read all the documents and understand what needs to be tested.
- Based on the information procured in the above step decide how it is to be tested.
- Inform the test lead about what all resources will be required for software testing.
- Develop test cases and prioritize testing activities.
- Execute all the test case and report defects, define severity and priority for each defect.
- Carry out regression testing every time when changes are made to the code to fix defects.

34. **What are the sources of defects. (R)(APR/MAY 2017)**
    - Miscommunication of requirements introduces error in code
    - Unrealistic time schedule for development
    - Lack of designing experience
    - Lack of coding practices experience
    - Human factors introduces errors in code
    - Lack of version control
    - Buggy third-party tools
    - Last minute changes in the requirement introduce error
    - Poor Software testing skill

35. **Why test cases should be developed for both valid and invalid inputs? (AN) (APR/MAY 2019)**
    The tester must consider both valid and invalid equivalence classes. Invalid classes represent erroneous or unexpected inputs.

36. **Mention the role of process in software quality? (R) (Nov/Dec 2018)**
    Process, in the software engineering domain, is the set of methods, practices, standards, documents, activities, policies, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code, and manuals.

37. **What is meant by Feature Defects? (R) (Nov/Dec 2018)**
    A Software feature can be defined as the changes made in the system to add new functionality or modify the existing functionality. Each feature is said to have a characteristics that is designed to be useful, intuitive and effective.
    Defect Severity is the impact that a defect has on either the development or execution of any program. It is the degree of impact that a defect has, on the application.

38. Mention the relationship between testing effectiveness and the quality of software system. (Apr/May-2021)

Testing focuses on system control and error detection, with product orientation and corrective actions. Testing is about checking the behavior of the application, while quality assurance is about making the overall quality level of the project better each day.

39. Who is called as a Test Specialist? (Apr/May-2021)

A test engineer is a professional who determines how to create a process that would best test a particular product in manufacturing and related disciplines, in order to assure that the product meets applicable specifications.

PART B

1. Write in brief about principles of software testing. **(R) (16) (May 2009) (Nov/Dec 2018)**

2. Explain: Testing as a process with small Example **(U) (16) (MAY 2012)**

3. Explain the role of process in software quality including components**. (R) (16)**

4. Discuss the origin of defects. **(R) (May 2009 & May 2013) (Nov/Dec 2018)**

5. Explain in detail about Tester‘s role in a software development organization. **(U) (16) (Nov/Dec 2017)**

6. Discuss in detail about the functions involved in Design defects. **(R) (16) (May 2010) (Nov/Dec 2017)**

7. Explain the process in coding defects. **(R) (16) (Dec 2011)**

8. How does a developer or tester support for developing a defect repository. Discuss. **(R) (May 2011) (Nov/Dec 2018)**

9. Explain the elements of the engineering disciplines. **(R) (16) (May 2010 & May 2011)**

10. Discuss in detail the internal structure of TMM Maturity Level. **(U) (16) (May 2009)**

11. Discuss about the role of process in software quality. **(U)**

12. Draw the 5-level structure of the testing maturity model and discuss about it. **(R)**

13. Explain in detail about the software testing principles. **(R) (May 2013)**

14. Give an example for defect classes and discuss them in detail**. (R) (Dec 2012 & Dec 2013)**

15. Explain the overview of the Testing Maturity Model (TMM) & the test related activities that should be done for V-model architecture.**(U) (May 2013 & Dec 2012) (Apr-May-2018)**

16. (i) Differentiate between validation and verification. Explain with them example. **(AN) (6)**

(ii) Write note on the following**.** **(R) (2+4+4) (Dec 2013)**

(1). Defect Classes

(2). Principles of software testing.

(3). Defect Test Design

17. Explain the role of tester in software development in detail with an example. Also

18. Explain the different phase in tester‘s mental life. **(R) (Dec 2013)**

19. (i)Explain the steps in developing defect repository. **(R) (Dec 2009)**

(ii)Describe the defect classes in detail with example. **(R) (April/May 2015)**

20. (i) Discuss the origins of defects and explain the defect repository. **(R) (Dec 2009) (Apr/May 2019)**

(ii)Explain the principles of software testing and describe the role of tester in software development organization. **(R)**

21. Explain the role of test and debug cycle in development process of software.

(U) (May/June 2014)

22. –Developer as a tester – Write the psychological views related to it. **(AP) (May/June 2014)**

23. Write the origin of defects with defect classes. **(R) (May/June 2014)**

24. Outline the goal of software testing and write a note on the software testing principles? **(R) (Apr/May 2019)**

25. How defect repository helps in test design and management? Explain. **(R) (May/June 2014)**

26. Discuss different testing principles being followed in Software Testing.

(AN) (April/May 2015)

27. Elaborate on the principles of software testing and summarize the tester role in software development organization. **(U) (Nov/Dec 2016) (Apr-May-2018)**

28. Explain in detail processing and monitoring of the defects with defect repository**.(15) (U) Nov/Dec 2016) (Apr-May-2018)**

29. i. State and explain in detail the various software testing principles. (8) **(R) (May/June 2016)**

ii. Explain the developer and tester support for the development of a defect repository.(8)

(R) (May/June 2016)

30. i. Define defect and illustrate the various origin of defects.(8) **(R) (May/June 2016)**

ii. What approach would you use to solve the concepts of defects with the coin problem?

(APPLY)(May/June 2016)

31. State and explain all Software Testing principles. (16) **(R)(APR/MAY 2017) (Nov/Dec 2017)**

32. What are the typical origins of defects? Explain the major classes of defects in the software artifacts.(16**) (R) (APR/MAY 2017)**

33. (i)Discuss in detail about various types of testing axioms **(5)(Apr/May-2021)**

(ii)Discuss in detail Tester's role in software development organization(8) **(Apr/May-2021)**

34. Explain in detail about software testing principles. **(5)(Apr/May-2021)**

35. Elaborate on various defect classes with examples. **(8)(Apr/May-2021)**

UNIT - II

TEST CASE DESIGN

Test case Design Strategies – Using Black Bod Approach to Test Case Design – Random Testing – Requirements based testing – Boundary Value Analysis – Equivalence Class Partitioning – State-based testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Evaluating Test Adequacy Criteria.

PART- A

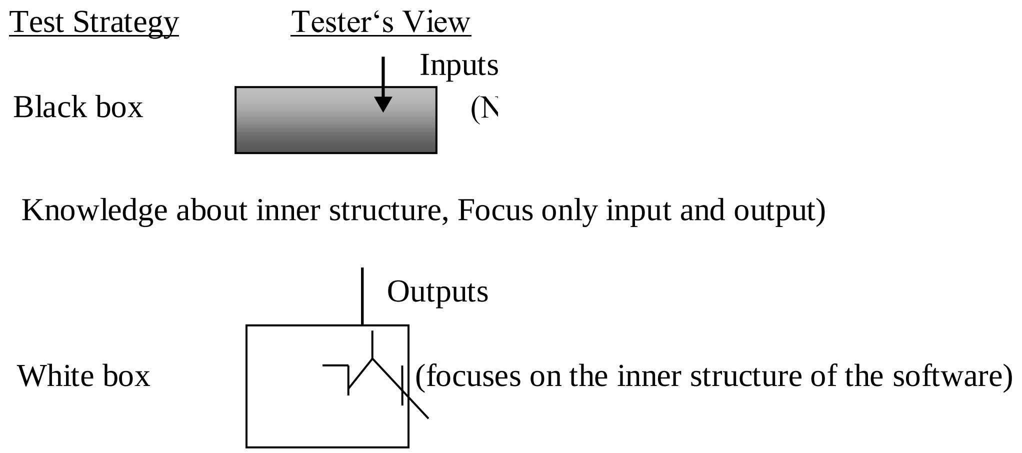
1. Define Smart Tester. (U) (DEC 2012)

Software must be tested before it is delivered to users. It is responsibility of the testers to Design tests that
(i) reveal defects
(ii) It can be used to evaluate software performance, usability and reliability. To achieve these goals, tester must select a finite no. of test cases (i/p, o/p, & conditions).

2. Compare black box and white box testing. (AN) (MAY 2012 & 2013, Dec 2014) (Nov/Dec 2017) (Nov/Dec 2018) (Apr/May 2019)

Black box testing	White box Testing
Black box testing , the tester is no Knowledge of its inner structure(i.e. how it works)The tester only has knowledge of what it does(Focus only input & output)	The White box approach focuses on the inner structure of the software to be tested.
Black box approach is usually applied Large size piece of software.	White box approach is usually applied Small size piece of software.
Black box testing sometimes called functional or specification testing.	White box sometimes called clear or glass box testing.

3. Draw the tester's view of black box and white box testing. (R) (Nov 2011)


Test Strategy
Tester‘s View
Inputs
Black box
(N
Knowledge about inner structure, Focus only input and output)
Outputs
White box
(focuses on the inner structure of the software)

4. Write short notes on Random testing and Equivalence class portioning.(U)

Each software module or system has an input domain from which test input data is selected. If a tester randomly selects inputs from the domain, this is called random testing. In equivalence class partitioning the input and output is divided in to equal classes or partitions.

5. What is CFG (control flow graph) with its merits? (R) (Dec 2013)

Static code analysis tools can provide control-flow and data-flow information. The control-flow information, presented in terms of a CFG, is helpful to the inspection team in that it allows the determination of the flow of control under different conditions. A CFG can be annotated with data-flow information to make a data-flow graph. For example, to each node of a CFG one can append the list of variables defined and used. This information is valuable to the inspection team in understanding the code as well as pointing out possible defects. Note that a static analysis tool might itself be able to discover several data-flow-related defects.

6. List the Knowledge Sources & Methods of black box and white box testing. (R) (Dec 2012)

Test Strategy	**Knowledge Sources**	**Methods**
Black box	1. Requirements document 2. Specifications 3. Domain Knowledge 4. Defect analysis data	1. Equivalence class partitioning (ECP) 2. Boundary value analysis (BVA) 3. State Transition testing.(STT) 4. Cause and Effect Graphing. 5. Error guessing
	1. High level design 2. Detailed design	1. Statement testing 2. Branch testing

White box	3. Control flow graphs 4. Cyclomatic complexity	3. Path testing 4. Data flow testing 5. Mutation testing 6. Loop testing

7. Define State. (R)

A state is an internal configuration of a system or component. It is defined in terms of the values assumed at a particular time for the variables that characterize the system or component.

8. Define Finite-State machine. (R)

A finite-state machine is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states.

9. Define Error Guessing. (R)

The tester/developer is sometimes able to make an educated –guess' as to which type of defects may be present and design test cases to reveal them. Error Guessing is an ad-hoc approach to test design in most cases

10. Define COTS Components. (R) (Nov 2011) (Nov/Dec 2018)

The reusable component may come from a code reuse library within their org or, as is most likely, from an outside vendor who specializes in the development of specific types of software components. Components produced by vendor org are known as commercial off-the shelf, or COTS, components.

11. Define usage profiles and Certification. (R)

Usage profiles are characterizations of the population of intended uses of the software in its intended environment. Certification refers to third party assurance that a product, process or service meets a specific set of requirements.

12. Write the application scope of adequacy criteria? (R)

- Helping testers to select properties of a program to focus on during test.
- Helping testers to select a test data set for a program based on the selected properties.
- Supporting testers with the development of quantitative objectives for testing
- Indicating to testers whether or not testing can be stopped for that program.

13. What are the factors affecting less than 100% degree of coverage?(AN) (Apr-May 2018)

- The nature of the unit
  - ➢ Some statements/branches may not be reachable.
  - ➢ The unit may be simple, and not mission, or safety, critical, and so complete coverage is thought to be unnecessary.
- The lack of resources

- ➢ The time set aside for testing is not adequate to achieve complete coverage for all of the units.
- ➢ There is a lack of tools to support complete coverage

- Other project related issues such as timing, scheduling. And marketing constraints.

14. What are the basic primes for all structured program. (R) (MAY 2012 &May 2013) (Nov/Dec 2017)

- Sequential (e.g., Assignment statements)
- Condition (e.g., if/then/else statements)
- Iteration (e.g., while, for loops)

15. Define path. (R)

A path is a sequence of control flow nodes usually beginning from the entry node of a graph through to the exit node.

16. Write the formula for cyclomatic complexity? (R) (Apr-May 2018)

The complexity value is usually calculated from control flow graph(G) by the formula.

(1) **V(G) = E-N+2**

Where The value E is the number of edges in the control flow graph. The value N is the number of nodes.

(2) **V(G) = P + 1**

Where P = Number of predicate nodes (node that contains condition)

17. List the various iterations of Loop testing. (R) (DEC 2012)

Zero iteration of the loop
One iteration of the loop
Two iterations of the loop
K iterations of the loop where k<n
n-1 iterations of the loop
n+1 iterations of the loop

18. Define test set. (R)

A test set T is said to be mutation adequate for program p provided that for every in equivalent mutant pi of p there is an element t in T such that pi[t] is not equal to p[t].

19. What are the errors uncovered by black box testing?(U)

Incorrect or missing functions
Interface errors
Errors in data structures
Performance errors
Initialization or termination error

20. Write the role of the path static test. (R) (Dec 2013)

Static testing is carried out without executing the application under test. This is in contrast to dynamic testing that requires one or more executions of the application under test. Static testing is useful in that it may lead to the discovery of faults in the application, as well as ambiguities and errors in requirements and other application-related documents, at a relatively low cost

21. What is positive and negative testing (R) (May/June 2014)

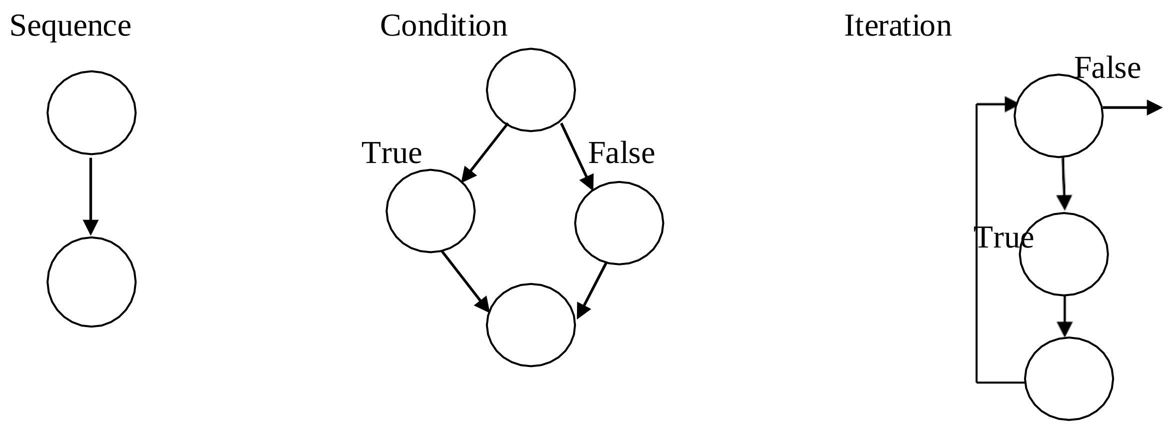
Positive testing tries to prove that a given product does what it is supposed to do. When a test case verifies the requirements of the product with a set of expected output, it is called positive test case.

Negative testing is done to show that the product does not fail when an unexpected input is given. The purpose of negative testing is to try and break the system.

22. What are the basic primes for all structured program. (R) (MAY 2012, DEC 2014)

- Sequential (e.g., Assignment statements)
- Condition (e.g., if/then/else statements)
- Iteration (e.g., while, for loops)

The graphical representation of these three primes are given

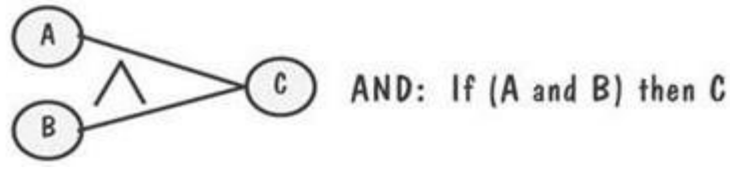


23. Define Test Adequacy Criteria. (R) (April/May 2015) (Apr/May 2019)

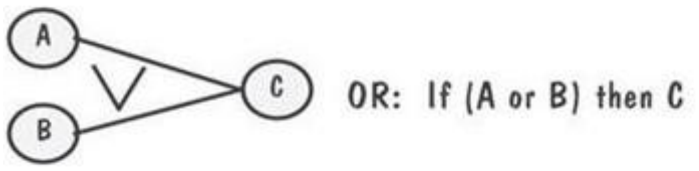
A software test adequacy criterion is a predicate that. Defines what properties of a program must be exercised to constitute a thorough test. Proposed test criteria: Reliability-producing consistent results.

24. Draw the notations used in cause effect graph. (R) (April/May 2015)

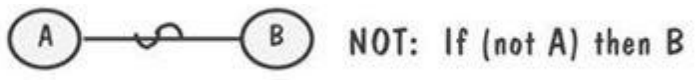
AND - For effect C to be true, both the causes A and B should be true



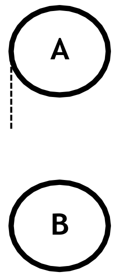
OR - For effect C to be true, either of causes A OR B should be true.



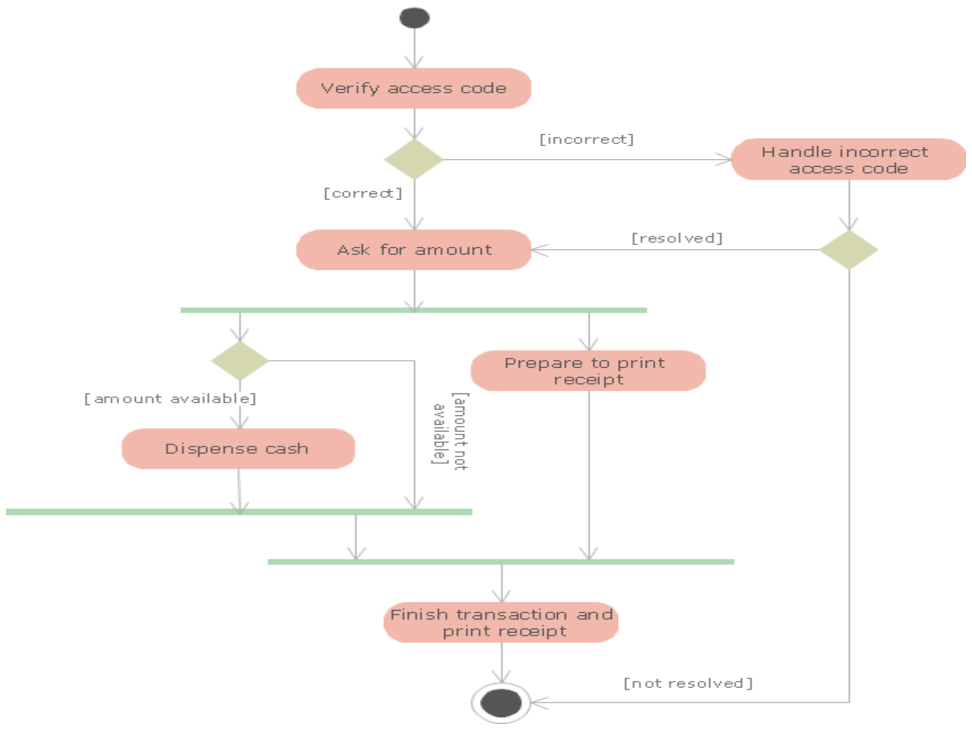
NOT – For Effect B to be True, Causes A should be false.



MUTUALLY EXCLUSIVE – When only one of the cause will hold true.



25. Sketch the control flow graph for an ATM withdrawal system. (R) (Nov/Dec 2016)



26. Give a note on the procedure to compute cyclomatic complexity. (R) (Nov/Dec 2016)

Cyclomatic complexity is a software metric (measurement), used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code. It was developed by Thomas J. McCabe, Sr. in 1976.

Cyclomatic complexity is computed using the control flow graph of the program: the nodes of the graph correspond to indivisible groups of commands of a program, and a directed edge connects two nodes if the second command might be executed immediately after the first command. Cyclomatic complexity may also be applied to individual functions, modules, methods or classes within a program.

27. What are the errors uncovered by black box testing?(U) (May/June 2016)

Black-box testing attempts to find errors in the following categories:
1. Incorrect or missing functions.
2. Interface errors.
3. Errors in data structures or external database access.
4. Behavior or performance errors, and
5. Initialization and termination errors.

28. Mention the ways by which test cases may be generated. Generate a test case for a scenario.(APPLY) (APR/MAY 2017)

How to write test cases for software:

1. Use a Strong Title.
2. Include a Strong Description.
3. Include Assumptions and Preconditions.
4. Keep the Test Steps Clear and Concise.
5. Include the Expected result.
6. Make it Reusable.
7. Title: Login Page – Authenticate Successfully on gmail.com.

Sample of a Test Case

Title: Login Page – Authenticate Successfully on gmail.com
Description: A registered user should be able to successfully login at gmail.com.
Precondition: the user must already be registered with an email address and password.

Assumption: a supported browser is being used.

Test Steps:

1. Navigate to gmail.com
2. In the 'email' field, enter the email of the registered user.
3. Click the _Next' button.
4. Enter the password of the registered user
5. Click _Sign In'

Expected Result: A page displaying the gmail user's inbox should load, showing any new message at the top of the page.

29. Error Vs Defect Vs Failure. Discuss.(ANALYZE) (APR/MAY 2017)

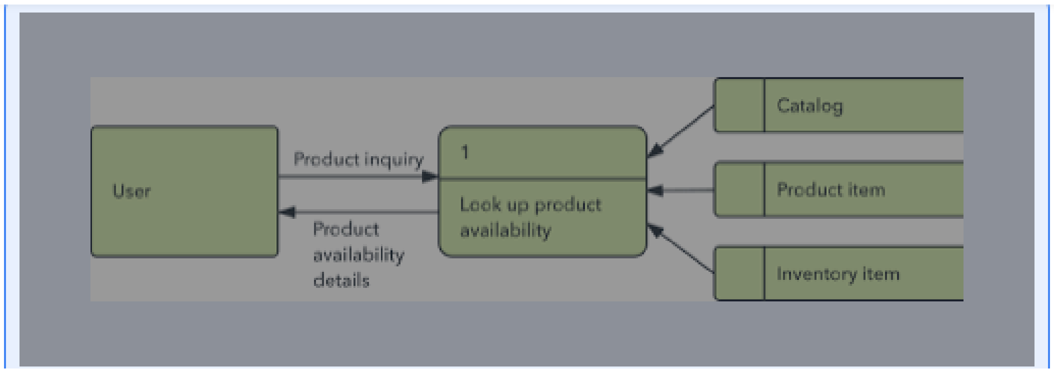
ERROR: An error is a mistake, misconception, or misunderstanding on the part of a software developer. In the category of developer we include software engineers, programmers, analysts, and testers. For example, a developer may misunderstand a de-sign notation, or a programmer might type a variable name incorrectly – leads to an Error. It is the one which is generated because of wrong login, loop or due to syntax. Error normally arises in software; it leads to change the functionality of the program.

DEFECT: It can be simply defined as a variance between expected and actual. Defect is an error found AFTER the application goes into production. It commonly refers to several troubles with the software products, with its external behavior or with its internal features. In other words Defect is the difference between expected and actual result in the context of testing. It is the deviation of the customer requirement.

FAILURE: A failure is the inability of a software system or component to perform its required functions within specified performance requirements. When a defect reaches the end customer it is called a Failure. During development Failures are usually observed by testers.

30.Give the diagrammatic representation of various components of engineered processes.(Apr/May-2021)

All data flow diagrams include four main elements: **entity, process, data store and data flow**.



31. What is code complexity testing? .(Apr/May-2021)

It is a quantitative measure of the number of linearly independent paths through a program's source code. It was developed by Thomas J. McCabe, Sr. in 1976. ... Cyclomatic complexity may also be applied to individual functions, modules, methods or classes within a program.

PART - B

1. Explain the Equivalence class partitioning of Black Box testing with example.**(R) (Dec 2011)**

2. Explain the Boundary value analysis of Black Box testing with example.**(U) (Dec 2011)**

3. Write a note on COTS Components. **(R) (May 2010)**

4. Explain Random Testing. **(R)**

5. Discuss the cause and effect graphing of black box testing. **(R) (May 2011)**

6. Write a note on state transition testing. **(R) (May 2011) (Apr-May-2018)**

7. Explain how to evaluate test adequacy criteria in white box test approach. **(AN) (May 2013)**

8. Write a note on following **(R) (16)**

(i) Loop Testing (ii) Mutation Testing

9.(i) Explain briefly about path and cyclomatic complexity.**(R) (May 2010 & Dec 2009)**

(ii) Explain the test factors that must be followed to design a customized test strategy.**(R)**

10. Explain with an example how to represent the control flow graph.**(R)**

11. Explain The Boundary Value Analysis and Equivalence Class Partitioning with example. **(U) (May 2013) (Nov/Dec 2017)**

12. Explain in detail about the Equivalence class partitioning. **(R)**

13. Discuss the various approaches in Black Box test design. **(R)**

14. Describe the difference between the white box and black box testing strategies.**(AN)**

15. What is a control flow graph? How is it used in white box test design? **(R)**

16. Explain the differences between random testing and testing using error guessing. **(AN)**

17. (i) Explain positive and negative testing in detail with an example**(R) (8) (Dec 2013)**

(ii)Compare and contrast between static testing and structural testing with an example.**(8) (R)**

18. Write note on the following. **(R) (4*4) (Dec 2013)**

(i) Smarter Test.

(ii)Decision tables

(iii)Code complexity testing

(iv)Desk checking

19. Describe the role of Oaths in white box testing and explain any two white box design approaches. **(U) (Dec 2009)**

20. How test cases are generated using the following black box testing approaches.

i. Boundary value analysis

ii. Equivalence Partitioning

21. Cause-Effect Graphing **(R) (May/June 2014)**

22. Consider the following program code. Apply white box testing based on control flow graph to generate test case from it.**(AP) (May/June 2014)**

```
begin
        int x,y,z;
        input ( x, y );
if ( x > 0 ) && ( y > 0 )
        z=pow ( x, y )
else if ( x < 0 ) && ( y > 0 )
z=pow ( ( -x ), y )
     else if ( x > 0 ) && ( y < 0 )
     z=pow ( x,( -y ) );
     else if ( x < 0 ) && ( y < 0 )
z=0;
while( y > 0 ) {
z + = 1;
y -- ;}
if(z > 0 ){
z - =1; }
```

23. Consider an application *App* that takes two inputs *name* and *age* where *name* is a nonempty string containing at most 20 alphabetic characters and age is an integer that must satisfy the constraint 0 age 80. The App is required to display an error message if the input value provided for age is out of range. The application truncates any name that is more than 20 characters in length and generates an error message if an empty string is supplied for name.

Construct test date for App using the

(i) uni-dimensional equivalent partitioning

(ii) multi- dimensional equivalent partitioning

(iii) boundary value analysis technique **(AP) (April/May 2015)**

24. Define White Box testing. Draw CFG for the program P. Identify distinct paths and calculate cyclomatic complexity of P. Write suitable test cases to satisfy all distinct paths. **(AP) (April/May 2015)**

Program P

```
1  begin
2  int num, product
3  bool done;
4  product = 1;
5  input(done);
6  while(!done){
7  input(num)
8  if(num>0)
9          product=product*num;
10 input(done);
11 }
12 output(product);
13 end.
```

25. Demonstrate the various black box test cases using Equivalence class partitioning and boundary value analysis to test a module for payroll system. **(Analyze) (Nov/Dec 2016) (Apr-May-2018)**

26. Explain the various white box techniques with suitable test cases. **(U) (Nov/Dec 2016)**

27. Discuss in detail about code coverage testing**.(U) (Nov/Dec 2016)**

28. i. Explain the significance of control flow graph and cyclomatic complexity in white box testing with a pseudo code for sum of n positive numbers. Also mention the independent paths with test cases. (8) **(APPLY) (May/June 2016) (Nov/Dec 2017)**

ii. Briefly explain the Weyuker's eleven axioms that allow testers to evaluate test adequacy criteria**.(8) (U) (May/June 2016)**

29. i. Demonstrate the various black box test cases using equivalence class partitioning and boundary value analysis to test a module for an ATM**.(8). (U)(May/June 2016)**

ii. Explain how black box testing is performed in COTS components. **(8).(U) (May/June 2016)**

30. Illustrate with an example the following black box testing techniques **(U)(APR/MAY 2017)**

- **i.** Equivalence Class Portioning.**(8)**
- **ii.** Boundary Value Analysis. **(8)**

31. Suppose you are testing defect coin problem artifacts, Identify the causes of various defects. What steps could have been taken to prevent the various classes of defects. (16) **(ANALYZE)(APR/MAY 2017)**

32. Explain the following methods of black box testing with example. (U) (Nov/Dec 2018)

(i). Equivalence class partitioning
(ii). Boundary value analysis

33. Discuss in detail about static testing and structural testing. Also write the difference between these testing concepts. (U) (Nov/Dec 2018)

34. Explain boundary value analysis, equivalence class partitioning and state based testing with an example? (U) (Apr/May 2019)

35. Outline the steps in constructing a control flow graph and computing cyclomatic complexity with an example? (AN) (Apr/May 2019).

36. Mention at least four test case design strategies and explain them in detail.(**13)Apr/May-2021**)

Discuss below testing methods with examples. (13) **Apr/May-2021**)

(i)Compatibility Testing.(ii) User documentation Testing (iii)Domain Testing (iv)Random Testing

UNIT – III

LEVELS OF TESTING

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing – Compatibility testing – Testing the documentation – Website testing.

PART A

1. **List the levels of Testing or Phases of testing. (R) (MAY 2012 & 2013, Dec 2014) (Nov/Dec 2018)**
   - Unit Test
   - Integration Test
   - System Test
   - Acceptance Test

2. **Define Unit Test and characterized the unit test. (R) (Nov 2011) (Nov/Dec 2017)**
   At a unit test a single component is tested. A unit is the smallest possible testable software component.
   It can be characterized in several ways
   - A unit in a typical procedure oriented software systems.
   - It performs a single cohesive function.
   - It can be compiled separately.
   - It contains code that can fit on a single page or a screen.

3. **List the phases of unit test planning. (R)**
   Unit test planning having set of development phases.
   - Phase1: Describe unit test approach and risks.
   - Phase 2: Identify unit features to be tested.
   - Phase 3: Add levels of detail to the plan.

4. **List the work of test planner. (R) (DEC 2012)**
   - Identifies test risks.
   - Describes techniques to be used for designing the test cases for the units.
   - Describe techniques to be used for data validation and recording of test results.
   - Describe the requirement for test harness and other software that interfaces with the unit to be tested, for ex, any special objects needed for testing object oriented.

5. **Define integration Test. (R)**
   At the integration level several components are tested as a group and the tester investigates component interactions.

6. **Define System test. (R)**
   When integration test are completed a software system has been assembled and its major subsystems have been tested. At this point the developers /testers begin to test it as a whole. System test planning should begin at the requirements phase.

7. **Define Alpha and Beta Test. (R) (MAY 2012)**
   Alpha test developer‘s to use the software and note the problems.
   Beta test who use it under real world conditions and report the defect to the Developing organization.

8. **What are the approaches are used to develop the software? (R)**
   There are two major approaches to software development
   - Bottom-Up
   - Top _ Down
   - These approaches are supported by two major types of programming languages.
   - They are
   - Procedure _ oriented
   - Object _ oriented

9. **List the issues of class testing**. **(R) (Nov 2011)**
   - Issue1: Adequately Testing classes
   - Issue2: Observation of object states and state changes.
   - Issue3: The retesting of classes-I
   - Issue4: The retesting of classes-II

10. **Define test Harness. (R) (DEC 2012 & MAY 2013) (Apr/May 2019)**
    The auxiliary code developed into support testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.

11. **Define Test incident report. (R) (Nov/Dec 2018)**
    The tester must determine from the test whether the unit has passed or failed the test. If the test is failed, the nature of the problem should be recorded in what is sometimes called a test incident report.

12. **What is the role of the Test Summary Report. (R) (Nov/Dec 2017)**
    The causes of the failure should be recorded in the test summary report, which is the summary of testing activities for all the units covered by the unit test plan.

13. **Goals of Integration test. (R)**
    To detects defects that occur on the interface of the units.
    To assemble the individual units into working subsystems and finally a completed system that ready for system test.

14. **What are the Integration strategies? (R) (Nov 2011)**
    Top_ Down: In this strategy integration of the module begins with testing the upper level modules.
    Bottom_ Up: In this strategy integration of the module begins with testing the lowest level modules.

15. **What is Cluster? (R)**
    A cluster consists of classes that are related and they may work together to support a required functionality for the complete system.

16. **List the different types of system testing. (R)**
    - Functional testing
    - Performance testing
    - Stress testing
    - Configuration testing
    - Security testing
    - Recovery testing

    The other types of system Testing are
    Reliability & Usability testing.

17. **Define load generator and Load. (R)**
    An important tool for implementing system tests is a load generator. A load generator is essential for testing quality requirements such as performance and stress

    A load is a series of inputs that simulates a group of transactions. A transaction is a unit of work seen from the system user‘s view. A transaction consists of a set of operation that may be performed by a person, s/w system or device that is outside the system.

18. **Define functional Testing. (R) (MAY 2012)**
    Functional tests at the system level are used ensure that the behavior of the system adheres to the requirement specifications.

19. **What are the two major requirements in the Performance testing. (R)**
    Functional Requirement: User describe what functions the software should perform. We test for compliance of the requirement at the system level with the functional based system test.
    Quality Requirement: They are nonfunctional in nature but describe quality levels expected for the software.

20. **Define stress Testing. (R) (Apr/May 2019)**
    When a system is tested with a load that causes it to allocate its resources in maximum amounts .It is important because it can reveal defects in real-time and other types of systems.

21. **Define Breaking the System. (R)**
    The goal of stress test is to try to break the system; Find the circumstances under which it will crash. This is sometimes called –breaking the system‖.

22. **What are the steps for top down integration?(U)**
    Main control module is used as a test driver and stubs are substituted for all components directly subordinate to the main module.
    Depending on integration approach (Depth or breadth first) subordinate stubs are replaced one at a time with actual components.

Tests are conducted as each component is integrated.
The completion of each set of tests another stub is replaced with real component
Regression testing may be conducted to ensure that new errors have not been introduced.

31. What is meant by regression testing? (R) (Nov/Dec 2018)

Regression testing is used to check for defects propagated to other modules by changes made to existing program. Thus, regression testing is used to reduce the side effects of the changes.

32. What is ad-hoc testing? (R) (May/June 2014)

Testing done without using any formal testing technique is called adhoc testing. Adhoc testing is done to explore the undiscovered areas in the product by using intuition, previous experience in working with the product, expert knowledge of the platform or technology and experience of testing a similar product.

33. How alpha and beta testing differs?(U)

Alpha test takes place at the developer‘s site. A cross-section of potential users and members of the developer‘s organization are invited to use the software. Developers observe the users and note problems.

Beta test sends the software to a cross-section of users who install it and use it under real world working conditions. The users send records of problems with the software to the development organization where the defects are repaired sometimes in time for the current release

34. Write the major activities followed in internationalization testing. (R) (April / May 2015)

- Testing to check if the product works across settings.
- Verifying the installation using various settings.
- Verify if the product works across language settings and currency settings

35. List out types of system testing. (R) (Nov/Dec 2016)

- Functional testing
- Performance testing
- Stress testing
- Configuration testing
- Security testing
- Recovery testing

The other types of system Testing are Reliability & Usability testing.

36. Compare and contrast Alpha testing and Beta Testing. (Analyze) (Nov/Dec 2016)

Alpha test takes place at the developer‘s site. A cross-section of potential users and members of the developer‘s organization are invited to use the software. Developers observe the users and note problems.

Beta test sends the software to a cross-section of users who install it and use it under real world working conditions. The users send records of problems with the software to the development organization where the defects are repaired sometimes in time for the current release

37. Why is it important to design test harness for unit testing? (May/June 2016)& (APR/MAY 2017) (Nov/Dec 2017)

- To supply inputs to the software being tested;
- To receive outputs generated by the software being tested;
- To execute a set of tests within the framework or using the test harness;
- To record the pass/fail results of each test (framework tools);
- To store tests (framework tools);
- Provide support for debugging (framework tools);
- To do coverage measurement at code level (framework tools).

38. What are the issues in testing object oriented systems? (May/June 2016)

- Encapsulation of attributes and methods in class may create obstacles while testing. As methods are invoked through the object of corresponding class, testing cannot be accomplished without object. In addition, the state of object at the time of invocation of method affects its behavior. Hence, testing depends not only on the object but on the state of object also, which is very difficult to acquire.
- Inheritance and polymorphism also introduce problems that are not found in traditional software. Test cases designed for base class are not applicable to derived class always (especially, when derived class is used in different context). Thus, most testing methods require some kind of adaptation in order to function properly in an OO environment.

39. Differentiate Black box with white box testing. (APR/MAY 2017)

Criteria	**Black Box Testing**	**White Box Testing**
Definition	Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester	White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.
Levels Applicable To	Mainly applicable to higher levels of testing,Acceptance Testing System Testing	Mainly applicable to lower levels of testing:Unit Testing Integration Testing
Responsibility	Generally, independent Software Testers	Generally, Software Developers
Programming Knowledge	Not Required	Required
Implementation Knowledge	Not Required	Required
Basis for Test Cases	Requirement Specifications	Detail Design

40. What is the advantage of Bottom up integration?(U) (Apr-May 2018)

- Appropriate for applications where bottom up design methodology is used.
- Advantageous if major flaws occur towards the bottom of the program.
- Test conditions are easier to create.
- Observation of test results is easier.
- Always starting at the bottom of the hierarchy again means that the critical modules are generally built and tested first and therefore any errors or mistakes in these forms of modules are identified early in the process.

41. Give the examples of Security Testing. (R) (Apr-May-2018)

This is an example of a very basic security test which anyone can perform on a web site/application:

- Log into the web application.
- Log out of the web application.
- Click the BACK button of the browser (Check if you are asked to log in again or if you are provided the logged-in application.)

Most types of security testing involve complex steps and out-of-the-box thinking but, sometimes, it is simple tests like the one above that help expose the most severe security risks.

Sample Test scenarios to give you a glimpse of security test cases -

- Password should be in encrypted format
- Application or System should not allow invalid users
- Check cookies and session time for application
- For financial sites, Browser back button should not work.

42. Brief the importance of test plan in software testing**.(Apr/May-2021)**

A test plan is the foundation of every testing effort. It helps set out how the software will be checked, what specifically will be tested, and who will be performing the test. By creating a clear test plan all team members can follow, everyone can work together effectively

43. List the features of risk matrix test tool. **.(Apr/May-2021)**

Risk assessment matrix is the probability impact matrix. It provides the project team with a quick view of the risks and the priority with which each of these risks needs to be addressed. Probability is the measure of the chance for an uncertain event will occur. **Exposure in terms of time, proximity and repetition**.

PART – B

1. How would you define a software unit? In terms of your definition, what constitutes a unit for procedural code; for object-oriented code? **(U) (May 2011)**
2. Discuss the issues that arise in class testing. **(U) (May 2011)**
3. Why is it so important to design a test harness for reusability? **(U) (May 2012)**

4. What are the key differences in integrating procedural-oriented systems as compared to object-oriented systems? **(AN) (May 2010)**
5. Describe the activities/Tasks and responsibilities for developer/testers in support of multilevel testing. **(U)**
6. Discuss the importance of following **(U)**

   (i) Security Testing. **(8)** (ii) Alpha Testing, Regression Testing **(8)**
7. Discuss **(U)**

   (i) Beta Testing, Recovery Testing (8) (ii) Acceptance Testing (8)
8. List and explain types of system test. **(R)**
9. Discuss the needs for various levels of testing. **(R) (DEC 2012)**
10. Explain the various units in a program considered for unit testing. **(R)**
11. Explain the planning of unit tests. **(R) (Dec 2011)**
12. Explain the design process in unit test. **(R) (Dec 2011)**
13. Explain the Execution process of unit test. **(R)**
14. Describe the Integration strategies for procedures and functions.**(U) (May 2010)**
15. Define Integration test with its design planning procedures along with its goals. **(U) (May 2012)**
16. Explain the different types of system tests? **(R) (May 2013)**
17. i) What is need for preparation of unit test?**(6) (U) (May 2013)**

    ii) Explain the Phases Involved in Unit Test Planning and How Will You Design It? **(U)**
18. .i) Write the needs of levels of testing. **(R) (4) (Dec 2013)**

    ii) Explain the levels of testing in detail. **(R) (12)**
19. i). Compare and contrast between regression testing and ad-hoc testing. **(AN) (6) (Dec 13)**

    ii) Explain how do you perform the unit test and record the result in detail with a suitable illustration. **(AN) (10)**
20. Explain the integration and its design and planning. **(R) (Dec 2009)**
21. What is regression testing? Give its types. When is it necessary to perform regression testing and how is it done? **(R) (Dec 2009)**
22. How unit testing is done in the initial stages of code testing? Explain the various activities involved in it. **(U) (May/June 2014)**
23. Explain –defect bash elimination‖ process involved in designing integration tests. **(U) (May/June 2014)**
24. How –Alpha-Omega‖ method is used for testing classes? Explain. **(R) (May/June 2014)**

25. Discuss in detail about different types of integration testing. **(R) (April/May 2015)**
26. Discuss the levels of testing adapted to test OO systems. **(U) (April/May 2015)**
27. Explain the different integration testing strategies for procedures and functions with suitable diagrams. **(R) (Nov/Dec 2016) (Nov/Dec 2017)**
28. How would you identify the hardware and software for configuration testing and explain what testing techniques applied for website testing? **(R)(Apply) (Nov/Dec 2016)**
29. i. Define a unit. Explain why test planning is so important for developing a repeatable and managed testing process? **(U) (8) (May/June 2016)**

    ii. Tabulate the key differences in integrating procedural oriented systems as compared to object oriented systems. **(ANALYZE) (8) (May/June 2016)**
30. Explain the different integration testing strategies for procedures and functions with suitable diagrams. **(R) (16) (May/June 2016)**
31. Explain the significance of Control flow graph and cyclomatic complexity in white box testing with a pseudo code for sum of positive numbers. Also mention the independent paths with test cases. **(APPLY) (16) (APR/MAY 2017)**
32. With examples explain the following black box techniques to testing **(U) (APR/MAY 2017)**
    i. Requirements based testing(4)
    ii. Positive and Negative testing.(4)
    iii. State based testing.(4)
    iv. User documentation and compatibility.(4)
33. Differentiate Alpha testing from beta testing and discuss in detail about the phases in which alpha and beta testing is done. **(16) (AN)(Nov/Dec 2017)**
34. Write the importance of security testing and explain the consequences of security breaches, also write the various areas which has to be focused on during security testing**.(7)(AN) (Apr-May-2018)**
35. State the need for integration testing in procedural code.**(6)(R) (Apr-May-2018)**
36. Explain about the unit test planning. **(7)(R) (Apr-May-2018)**
37. Explain about configuration testing and its objectives.**(6)(U) (Apr-May-2018)**
38. State unit test and describe about planning and designing of unit test. **(U) (Nov/Dec 2018)**
39. Explain elaborately about the various types of system testing. **(R) (Nov/Dec 2018)**
40. (i). Outline the importance of system testing with an example? **(U) (Apr/May 2019)**

    (ii). What is regression testing? Outline the issues to be addressed for developing test cases to perform regression testing?

41. Present the outline of testing object oriented systems? **(R) (Apr/May 2019)**

42. Discuss in detail about various levels of testing along with planning and designing test cases in each one of them**.(13) (Apr/May-2021)**

43. Write a short note on below testing methods **(13)(Apr/May-2021)**

    (i) Alpha Testing (ii) Beta Testing (iii) OO Testing (iv) Website Testing.

44. Differentiate end to end testing and functional testing. When to apply end to end testing and detail various methods of end to end testing**.(15))(Apr/May-2021)**

UNIT - IV

TEST MANAGEMENT

People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group.

PART – A

1. **Write the different types of goals? (R) (DEC 2012)**
   Business goal: To increase market share 10% in the next 2 years in the area of financial software
   Technical Goal: To reduce defects by 2% per year over the next 3 years.
   Business/technical Goal: To reduce hotline calls by 5% over the next 2 years
   Political Goal: To increase the number of women and minorities in high management positions by 15% in the next 3 years.

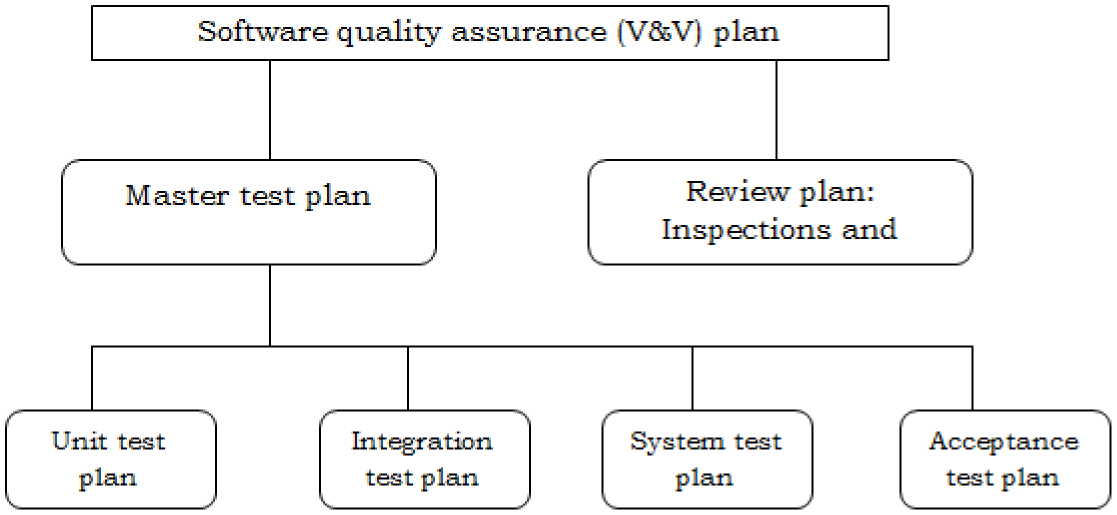
2. **Define Goal and Policy(R) (MAY 2012)**
   A goal can be described as a statement of intent (or) a statement of a accomplishment that an individual or an org wants to achieve.

   A Policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an org.

3. **List the Test plan components. (R)**
   - Test plan identifier
   - Introduction
   - Items to be tested
   - Features to be tested
   - Approach

- Pass/fail criteria
- Suspension and resumption criteria
- Test deliverables
- Testing Tasks
- Test environment
- Responsibilities
- Staffing and training needs
- Scheduling
- Risks and contingencies
- Testing costs.

4. Draw a hierarchy of test plans. (R) (DEC 2012)



5. Define a Work Breakdown Structure.(WBS) (R) (MAY 2012) (Apr-May-2018)

A Work Breakdown Structure (WBS) is a hierarchical or treelike representation of all the tasks that are required to complete a project.

6. Write the approaches to test cost Estimation? (R)

The COCOMO model and heuristics

- Use of test cost drivers
- Test tasks
- Tester/developer ratios
- Expert judgment

7. Write short notes on Cost driver. (R)

- A Cost driver can be described as a process or product factor that has an impact on overall project costs. Cost drivers for project the include
- Product attributes such as the required level of reliability

- Hardware attributes such as memory constraints.
- Personnel attributes such as experience level.
- Project attributes such as tools and methods.

8. Write the WBS elements for testing. (R)

- Project startup
- Management coordination
- Tool selection
- Test planning
- Test design
- Test development
- Test execution
- Test measurement, and monitoring
- Test analysis and reporting
- Test process improvement

9. What is the function of Test Item Transmittal Report or Locating Test Items(R) (Apr-May-2018)

Suppose a tester is ready to run tests on the data described in the test plan. He needs to be able to locate the item and have knowledge of its current status. This is the function of the Test Item Transmittal Report. Each Test Item Transmittal Report has a unique identifier.

10. What is the information present in the Test Item Transmittal Report or Locating Test Items (U) (May 2013)

a. Version/revision number of the item
b. Location of the item
c. Person responsible for the item (the developer)
d. References to item documentation and test plan it is related to.
e. Status of the item
f. Approvals – space for signatures of staff who approve the transmittal.

11. What are the three critical groups in testing planning and test plan policy? (R)

Managers: Task forces, policies, standards, planning Resource allocation, support for education and training, Interact with users/Clients

Developers/Testers:

Apply Black box and White box methods, test at all levels, Assst with test planning, Participate in task forces.

Users/Clients:

Specify requirement clearly, Support with operational profile, Participate in acceptance test planning

12. Define Procedure. (R) (DEC 2012)

A procedure in general is a sequence of steps required to carry out a specific task.

13. List down the skills needed by test specialist? (R) (April/ May 2015) (Nov/Dec 2017)

Personal and managerial Skills

Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively.

Technical Skills

General software engineering principles and practices, understanding of testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.

14. Write the test term hierarchy? (R)

Test Manager
Test leader
Test Engineer
Junior Test Engineer

15. Define Test incident Report. (R) (Nov 2011)

The tester should record in attest incident report (sometimes called a problem report) any event that occurs during the execution of the tests that is unexpected , unexplainable, and that requires a follow- up investigation.

16. Define Plan. (R)

A plan is a document that provides a framework or approach for achieving a set of goals.

17. Define Milestones. (R) (Nov 2011)

Milestones are tangible events that are expected to occur at a certain time in the Project‘s lifetime. Managers use them to determine project status.

18. Write the steps in forming a test group? (R) (May 2013)

i. Organizing
ii. Staffing
iii. Directing

19. List out any two organizational issues in testing. (R) (Dec 2013)

i. Organization type
ii. Geographic distribution

20. Write the business impacts of globalization. (U) (Dec 2013)

Since the markets for software are global, the needs that the product must satisfy are increasing exponentially.

21. List the internal and external dependencies for executing WBS. (R) (April /May 2015)

External Dependencies

- Availability of the product from developers;
- Training
- Acquisition of hardware / software required for training.
- Availability of translated message files for testing.

Internal Dependencies

- Completing the test specification
- Coding / scripting the tests
- Executing the tests

22. Discuss on the role of manager in a test group.(U) (Nov/Dec 2016)

Effective software test managers not only understand the discipline of testing, but they are also able to manage and implement a testing process in their organizations. That requires team leading skills, communication skills, and being able to measure the testing team's return on investment.

23. What are the issues in testing object orient system? (U) (Nov/Dec 2016)

1. Encapsulation of attributes and methods in class may create obstacles while testing. As methods are invoked through the object of corresponding class, testing cannot be accomplished without object. In addition, the state of object at the time of invocation of method affects its behavior. Hence, testing depends not only on the object but on the state of object also, which is very difficult to acquire.
2. Inheritance and polymorphism also introduce problems that are not found in traditional software. Test cases designed for base class are not applicable to derived class always (especially, when derived class is used in different context). Thus, most testing methods require some kind of adaptation in order to function properly in an OO environment.

24. Make distinctions between structures of single product and multi product companies. (Analyze) (May/June 2016)

- As a large company, with only a single product, the focus is on: further development, creative new applications or editions and effective marketing on your product. Take Crocs shoes for example. The shoes have the same basic structure but you can purchase them in different colour, size and with individual stickers.
- When you are a small company your chances are half-chance. Either the focus is on growth or the company is at a level where it earns enough and is able to pay all costs.

25. Mention the reasons to create a WBS. (U)(May/June 2016)

1. The WBS helps ensure that all required scope will be reflected in the detailed schedule.
2. The WBS is a risk-mitigation tool.
3. The WBS creates the resource allocation and control structure for the project.

4. The WBS is an excellent communication tool.
5. The WBS enables traceability between project documents.

26. State the limitations of statement coverage.(U)(APR/MAY 2017)

- It cannot test the false conditions.
- It does not report that whether the loop reaches its termination condition.
- It does not understand the logical operators.

27. Differentiate decision and condition coverage.(Analyze) (APR/MAY 2017)

- Decision coverage - It checks whether every edge of the program is covered or not i.e. each edge of every control structure(i.e. IF and CASE statements)are covered or not.
- Condition coverage - It checks whether every conditional statements are covered or not. It evaluates for only true and false condition for each conditional statement not every edge of those conditional statements.
-

28. Outline the need for a test plan? (R) (Apr/May 2019)

A test plan serves as a roadmap to the testing process that has all the necessary details related to the process. It serves a means of communication between the team members and stakeholders and keeps a record of what was tested in a particular release, along with any comments or conversation notes.

Test plans often help testers generate more bugs than exploratory testing. They are able to test all sections and all use cases. A benefit of test plans is that exploratory testing generates good and valuable bugs. They generally imply “out of the box” thinking, but as for quantity, nothing beats a good test plan.

29. What is a test log? (R) (Apr/May 2019)

Test Log is a document which consists of information about the test cases.Means whether the test case is Passed or Failed. Test log is nothing but the addition of 2 fields namely 'Actual result' and 'Pass/Fail Criteria' to the test case i.e., already populated with

test case id
test description
test pre-requesties
test steps
expected result.

30. Mention the duties of component-wise testing teams? (R) (Nov/Dec 2018)

This type of software testing is performed by developers. Component testing is carried out after completing unit testing. Component testing involves testing a group of units as code together as a whole rather than testing individual functions, methods.

31. Mention few aspects of software program which can be validated by performance testing**.(Apr/May-2021)**

Speed – Determines whether the application responds quickly.

Scalability – Determines maximum user load the software application can handle.

Stability – Determines if the application is stable under varying loads.

32. Give short note on cost estimation**.(Apr/May-2021)**
Cost estimation in project management is the process of forecasting the cost and other resources needed to complete a project within a defined scope. Cost estimation accounts for each element required for the project and calculates a total amount that determines a project's budget.

PART – B

1. Why is testing planning so important for developing a repeatable and managed testing process? **(U) (Dec 2011)**

2. Why is it so important to integrate testing activities into the software life cycle? **(U) (May 2011)**

3. What role do managers play in support of a test group? **(R)**

4. Discuss in detail about the test plan components. **(R) (May 2010)**

5. Explain the steps in forming a test group. **(R) (Nov/Dec 2017)**

6. Explain in brief about test cost impact items. **(U) (May 2009)**

7. Explain elaborately about the basic test plan components as described in IEEE829 - 1983. **(R) (May 2010)**

8. What role do user/client play in the development of test plan for a project? Should they be present at any of the test plan reviews? Justify? **(U) (May 2009)**

9. Explain the test plan components and attachments? **(R) (May 2012 & Dec 2012)**

10. Explain with neat diagram the hierarchy of test plans? **(R)**

11. Explain the activities in reporting test results? **(R) (May 2011)**

12. Explain the role of the three critical groups in testing planning and test policy development. **(R) (May 2013)**

13. Write the various approaches used to test cost estimation. **(R) (May 2013)**

14. (i) Explain the various issues of people and organizational testing. **(R) (6) (Dec 2013)**

(ii) What are the three groups in test planning? Explain each group and its function in detail. **(10) (R)**

15. Explain the organization structure of testing team and present the testing services. **(R) (10)**

16. (i) State and discuss the various stages that a test plan will consist of**(R)**

(ii) Explain the role of testing. **(R) (Dec 2009)**

17. (i) Explain the challenges and issues faced in testing services organization. Also write how we can eliminate challenges. **(AN) (Dec 2009)**

(ii) How can we build a test group? **(R)**

18. Explain the perceptions and misconception about testing. **(U) (May/June 2014)**

19. Which groups are involved in test planning and policy development? Explain each group‘s functionalities. **(R) (May/June 2014)**

20. Discuss the roles and responsibilities of testing services organization with suitable organization structure. **(U) (April/May 2015)**

21. Discuss the different test process activities of software testing in detail. **(R) (April/May 2015)**

22. What are the skills needed for a test specialist? **(U) (Dec 2013), (May/June 2014), (Nov/Dec 2016) (Nov/Dec 2017) (Apr-May-2018)**

23. Explain the organizational structure for testing teams in single product companies. **(R) (Nov/Dec 2016) (Apr-May-2018)**

24. Explain the components of test plan in detail.**(R) (Nov/Dec 2016) (Nov/Dec 2017) (Apr-May-2018) (Nov/Dec 2018)**

25. Compare and contrast the role of debugging goals and policies in testing. **(Analyze) (Nov/Dec 2016)**

26. Explain the various impacts of globalization and geographically distributed teams on product testing. **(U) (16) (May/June 2016)**

27. Explain the different challenges and issues faced in the testing service organization. Discuss how challenges can be addressed. **(U) (16) (May/June 2016)**

28. i. How data flow testing aid in identifying defects in variable declaration and its use.**(U) (8) (APR/MAY 2017)**

ii. Explain mutation testing with an example**. (R) (8) (APR/MAY 2017)**

29. Explain Weyuker‘s eleven axioms that allow testers to evaluate test adequacy criteria**.(U) (16) (APR/MAY 2017)**

30. Name the reports of test results and the contents available in each test reports.**(6) (R) (Apr-May-2018)**

31. Explain the concepts of test planning in detail. Also mention the way od defining test plan. **(R) (Nov/Dec 2018)**

32. Describe the concepts of building a test group. **(R) (Nov/Dec 2018)**

33. (i). Discuss the advantages and disadvantages of having an independent test group, that is, one that is a separate organizational entity with its own reporting structure. **(U) (Apr/May 2019)**

(ii). Why is it so important to integrate testing activities into the software life cycle?

34. Explain the following test related documents and its components. **(R) (Apr/May 2019)**

(i). Test case specification
(ii). Test incident report

35. Assume you are working in an online fast food restaurant system. The system reads customer orders, relays orders to the kitchen, calculates the customer's bill, and gives change. It also maintains inventory information. Each wait-person has a terminal. Only authorized wait-persons and a system administrator can access the system. Describe the tests that are suitable to test the application. **(AN) (Apr/May 2019)**

36. Suppose you are developing an online system for a specific vendor of electronic equipment with all the necessary features to run the shop. Write down a detailed test plan by including the necessary components. **(C) (Apr/May 2019)**

37. Explain in detail about the organization structures of testing teams in single product and multi product companies along with their advantages **(13)(Apr/May-2021)**

38. Elaborate different tools and techniques used for testing a software. **(13)(Apr/May-2021)**

39. Discuss in detail the testing methodology which validates software system against functional requirements and specifications. Also explain the complete process with a neat diagram**.(15)(Apr/May-2021)**

UNIT - V

TEST AUTOMATION

Software test automation – skill needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.

PART A

1. **Define Project monitoring or tracking. (R) (MAY 2012)**
   Project monitoring refers to the activities and tasks managers engage into periodically check the status of each project .Reports are prepared that compare the actual work done to the work that was planned.

2. **Define Project Controlling. (R) (DEC 2012)**
   It consists of developing and applying a set of corrective actions to get a project on track when monitoring shows a deviation from what was planned.

3. **Define Milestone. (R) (Nov 2011)**
   Milestones are tangible events that are expected to occur at a certain time in the projects life time .Managers use them to determine project status.

4. **Define Base line. (R)**
   Base lines are formally reviewed and agreed upon versions of software artifacts, from which all changes are measured. They serve as the basis for further development and can be changed only through formal change procedures.
5. **Differentiate version control and change control. (AN) (Nov 2011)**
   Version Control combines procedures and tools to manage different versions of configuration objects that are created during software process.
   Change control is a set of procedures to evaluate the need of change and apply the changes requested by the user in a controlled manner.

6. **What is Testing? (R) (DEC 2012)**
   Testing is generally described as a group of procedures carried out to evaluate some aspect of a piece of software. It used for revealing defect in software and to evaluate degree of quality.

7. **Define Review. (R) (MAY 2012 & Dec 2013)**
   Review is a group meeting whose purpose is to evaluate a software artifact or a set of software artifacts.

8. What are the goals of Reviewers? (R) (Apr-May-2018)

- Identify problem components or components in the software artifact that need improvement.
- Identify components of the software artifact that do not need improvement.
- Identify specific errors or defects in the software artifact.
- Ensure that the artifact confirms to organizational standards.

9. What are the benefits of a Review program? (R)

- Higher quality software
- Increased productivity
- Increased awareness of quality issues
- Reduced maintenance costs
- Higher customer satisfaction

10. What are the various types of Reviews? (R)

- Inspections
- Walk Throughs

11. What is Inspections? (R)

It is a type of review that is formal in nature and requires pre-review preparation on the part of the review team. The Inspection leader prepares is the checklist of items that serves as the agenda for the review.

12. What is Walk Throughs? (R) (Apr-May-2018)

Code Walkthrough is a form of peer review in which a programmer leads the review process and the other team members ask questions and spot possible errors against development standards and other issues.

13. What are the uses of WalkThroughs?(U) (Nov/Dec 2017)

The main purpose of walkthrough is to enable learning about the content of the document under review to help team members gain an understanding of the content of the document and also to find defects.

14. List out the members present in the Review Team. (R)

- SQA(Software Quality Assurance) staff
- Testers
- Developers
- Users /Clients.
- Specialists.

15. List the components of review plans. (R)

- Review Goals

- Items being reviewed
- Preconditions for the review.
- Rolls,Team size,participants.
- Training requirements.
- Review steps.
- Time requirements

16. Define SCM (Software Configuration management). (R) (May 2013)
Software Configuration Management is a set of activities carried out for identifying, organizing and controlling changes throughout the lifecycle of computer software.

17. What is the scope of automation? (R) (Dec 2013)

Test Automation demands considerable investments of money and resources. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required . This improved ROI of Test Automation.
Goal of Automation is to reduce number of test cases to be run manually and not eliminate manual testing all together.

18. Define the following terms with respect to software quality evaluation

a)Compatibility (b)Interoperability (R) (May 2013)
Compatibility testing is one of the several types of software testing performed on a system that is built based on certain criteria and which has to perform specific functionality in an already existing setup/environment. Compatibility of a system/application being developed with, for example, other systems/applications, OS, Network, decide many things such as use of the system/application in that environment, demand of the system/application etc.

Interoperability testing involves testing whether a given software program or technology is compatible with others and promotes cross-use functionality. This kind of testing is now important as many different kinds of technology are being built into architectures made up of many diverse parts, where seamless operation is critical for developing a user base.

18. Write the different types of reviews practiced by software industry. (R) (April /May 2015)
Software reviews may be divided into three categories
- Software peer reviews
- Software management reviews
- Software audit reviews

19. Differentiate effort and schedule (R) (April /May 2015)

Effort :

- Testing effort can also be taken as a percentage of development effort which can be found out from the experience in different projects in an organization. It is generally in the range of 20% - 40%
- Provide a summary of test estimates (cost or effort) and/or provide a link to the detailed estimation.

Schedule : Provide a summary of the schedule, specifying key test milestones, and/or provide a link to the detailed schedule.

20. Mention the criteria for selecting test tool. (U) (Nov/Dec 2016)

- Assessment of the organization's maturity (e.g. readiness for change);
- Identification of the areas within the organization where tool support will help to improve testing processes;
- Evaluation of tools against clear requirements and objective criteria;
- Proof-of-concept to see whether the product works as desired and meets the requirements and objectives defined for it;
- Evaluation of the vendor (training, support and other commercial aspects) or open-source network of support;
- Identifying and planning internal implementation (including coaching and mentoring for those new to the use of the tool).

21. Distinguish between milestone and deliverable. (Analyze) (Nov/Dec 2016)

- Deliverables can be specific small tasks
- Deliverables can also be strategic goals or parameters
- Milestones are composed of many smaller tasks and mark important phases of a project/development, such as new software release, and thus could be composed of many smaller Deliverables.

22. State any two generic requirements for test tool and framework. (May/June 2016)

The main technical requirement for a test tool framework is to find a general way of connecting the test tools to the nodes in the System Under Test (SUT). Ideally the whole platform, for example TSP, should be viewed as a single unit with one single access point for connecting test tools to the nodes to be tested. The second technical requirement for a test tool framework is to centralize functions common to different test tools, as a base for building new test tools. The test tool framework could, for example, have support functions for logging and debugging. Another possibility is a common Graphical User Interface (GUI) that the test tools can use.

23. What are the skills needed in automation? (May/June 2016)

1. Configuration Management (CM) Software Experience
2. Troubleshooting
3. Development Methodology
4. Coding and Scripting Expertise
5. Certifications

24. Mention the challenges in Test automation. (APR/MAY 2017) (Nov/Dec 2017)

1. Testing the complete application
2. Misunderstanding of company processes
3. Relationship with developers
4. Regression testing
5. Lack of skilled tester
6. Testing always under time constraint
7. Which tests to execute first?
8. Understanding the requirements
9. Automation testing
10. Decision to stop the testing

25. Mention the types of testing amenable to automation. (R) (APR/MAY 2017)

Stress, reliability, scalability and performance tests.

26. Name any two software testing tools? (R) (Apr/May 2019)

Selenium, TestingWhiz, TestComplete, Ranorex, Appium etc.

27. Outline the need for test metrics? (R) (Apr/May 2019)

Helps to analyze the risks associated in a deeper way. Helps to analyze metrics in every phase of testing to improve defect removal efficiency. Improves Test Automation ROI over a time period. Enforces better relationship between testing coverage, risks and complexities of the systems.

28. What is the need for Automated Testing? (U) (Nov/Dec 2018)

Automated software testing is important due to the following reasons: Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming. It is difficult to test for multilingual sites manually. Automation does not require Human intervention.

29. Define Progress Metrics? (R) (Nov/Dec 2018)

Progress Metrics Usually when we track *progress*, it is related to time, or other unit that indicates a schedule. Most often we use *progress metrics* to track planned versus actual over time.

30. List various productivity metrics in software testing. (Apr/May-2021)

- Test Case Productive Preparation. = Total test steps / effort (hours) ...
- Test Execution Summary. Summarize your reports with the following parameters such as. ...
- Test Case Coverage. ...
- Defect Acceptance. ...

- Defect Rejection. ...
- Test Efficiency. ...
- Effort Variance. ...
- Schedule Variance.

31. What are the various challenges in Automation. **(Apr/May-2021)**

- Effective Communicating and Collaborating in Team. This is perhaps a challenge not just in test automation but also in manual testing teams. ...
- Selecting a Right Tool. ...
- Demanding Skilled Resources. ...
- Selecting a Proper Testing Approach. ...
- High Upfront Investment Cost.

PART -B

1. Discuss in detail about the controlling and monitoring: three critical views. **(R) (May 2009)**
2. Explain in detail about the role of reviews in testing software deliverables. **(R) (May 2011)**
3. Discuss in detail about the components of review plans. **(R) (Dec 2009)**
4. Explain in detail about the software configuration management. **(R) (Dec 2009)**
5. Explain about the various types of reviews. **(R)**
6. Suppose you are a test manager, what are the milestone you should select for unit test plan an integration test plan, and a system test plan? **(C) (May 2011)**
7. Suggest appropriate measurements for monitoring defects/faults and failure.**(U) (May 2011)**
8. Which groups do you think should contribute to membership of a configuration control board? Why?**(AN) (May 2010)**
9. What is a role of a tester in supporting the monitoring and controlling of testing? **(U)**
10. Discuss the critical views in controlling and monitoring?**(U) (May 2010)**
11. Explain the major activities of SCM? **(R)**
12. What should be included in a milestone report for testing? Who should be included on distribution list?**(U) (Dec 2011)**
13. Discuss in detail about review types and the need for review policies? **(R) (Dec 2011)**
14. How will you report Review results? **(R)**
15. Define review? Write the roles of reviews in testing software deliverables. How will you develop deliverable program. **(R) (May 2013)**
16. Write need for testing maturity model.(4) **(U) (May 2013)**
17. Explain the types of reviews and how to develop the review program in detail with suitable illustration. **(R) (Dec 2013)**
18. i) Discuss the design and architecture of automation in detail**. (10) (R) (Dec 2013, May/June 2014)**
19. ii) Explain the components of review plans and control issues**.(6) (R) (Dec 2009)**

i. Test metrics. ii. Testing tools

20. Explain the scope of automation in software testing in detail. **(R) (May/June 2014)**

21. List the steps for tool selection and deployment**. (R) (May/June 2014)**

22. Explain W-Model and its phases in test automation. **(R) (May/June 2014)**

23. Elaborate different types of S/W metrics and measurement used **(U) (April/May 2015)**

24. Explain the design and architecture for test automation with examples. **(R) (April/May 2015) (Nov/Dec 2017)**

25. Explain the design and architecture for automation and outline the challenges**.(U) (Nov/Dec 2016)**

26. What are metrics and measurements? Illustrate the types of product metrics. (R) (**Nov/Dec 2016)**

27. Explain the various types of Progress and Productivity metrics based on what they measure and what area they focus on. (16) (U) **(May/June 2016)**

28. Explain the design and architecture for software test automation.(16) (R**)(May/June 2016)**

29. Explain the various generations of automation and the required skills for each.(16)(U) **(APR/MAY 2017)**

30. Explain the different types of Test defect metrics under Progress metrics based on what they measure and what area they focus on. (16)(U) **(APR/MAY 2017)**

31. Discuss various metrics and measurements in software testing. Explain various types of progress metrics.(16) (R) **(Dec 2013, May/June 2014)**

32. Discuss the types of reviews. Explain various components of review plans.**(13)(U) (Apr-May-2018)**

33. Narrate about the metrics or parameters to be considered for evaluating the software quality**.(13) (AP) (Apr-May-2018)**

34. Write short notes on the following. **(R) (Nov/Dec 2018)**

(i). Classifications of automation testing
(ii). Scope of automation

35. Discuss in detail about selecting the test tool in test automation. **(U) (Nov/Dec 2018)**

36. Developing software to test the software is called automation. Test automation can help address several problems, justify. Draw the framework for test automation. **(AN) (Apr/May 2019)**

37. Outline project, product, and productivity metrics with relevant examples. (R) (Apr/May 2019).

38. Explain the structure of test group with component wise testing team ? **(13)(Apr/May-2021)**