

# Software QA Job Interview Questions And Answers



**Interview Questions Answers**

<https://interviewquestionsanswers.org/>

## About Interview Questions Answers

**Interview Questions Answers . ORG** is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Software QA will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Software QA Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Software QA category. To ensure quality, each submission is checked by our team, before it becomes live. This [Software QA Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.  
[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter for latest Jobs and interview preparation guides.  
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

**Interview Questions Answers.ORG Team**  
[https://InterviewQuestionsAnswers.ORG/  
Support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG)



## Software QA Interview Questions And Answers Guide.

### Question - 1:

Explain a good test engineer?

#### Ans:

A good test engineer because he Has a "test to break" attitude, Takes the point of view of the customer, Has a strong desire for quality, Has an attention to detail, He's also Tactful and diplomatic and Has good a communication skill, both oral and written. And he Has previous software development experience, too. Good test engineers have a "test to break" attitude, they take the point of view of the customer, have a strong desire for quality and an attention to detail. Tact and diplomacy are useful in maintaining a cooperative relationship with developers and an ability to communicate with both technical and non-technical people. Previous software development experience is also helpful as it provides a deeper understanding of the software development process, gives the test engineer an appreciation for the developers' point of view and reduces the learning curve in automated test tool programming.

[View All Answers](#)

### Question - 2:

What is Constant?

#### Ans:

In software or software testing, a constant is a meaningful name that represents a number, or string, that does not change. Constants are variables that remain the same, i.e. constant, throughout the execution of a program.

Why do we, developers, use constants? Because if we have code that contains constant values that keep reappearing, or, if we have code that depends on certain numbers that are difficult to remember, we can improve both the readability and maintainability of our code, by using constants.

To give you an example, we declare a constant and we call it "Pi". We set it to 3.14159265, and use it throughout our code. Constants, such as Pi, as the name implies, store values that remain constant throughout the execution of our program.

Keep in mind that, unlike variables which can be read from and written to, constants are read-only. Although constants resemble variables, we cannot modify or assign new values to them, as we can to variables, but we can make constants public or private. We can also specify what data type they are.

[View All Answers](#)

### Question - 3:

What is TestDirector?

#### Ans:

TestDirector, also known as Mercury TestDirector, is a software tool made for software QA professionals. Mercury TestDirector, as the name implies, is a product made by Mercury Interactive Corporation, 379 North Whisman Road, Mountain View, California 94043 USA.

Mercury's other products include the Mercury QuickTest Professional, Mercury WinRunner, also known as WinRunner, and Mercury Business Process Testing.

[View All Answers](#)

### Question - 4:

What is Data integrity?

#### Ans:

Data integrity is one of the six fundamental components of information security. Data integrity is the completeness, soundness, and wholeness of the data that also complies with the intention of the creators of the data.

In databases, important data - including customer information, order database, and pricing tables - may be stored. In databases, data integrity is achieved by preventing accidental, or deliberate, or unauthorized insertion, or modification, or destruction of data.

[View All Answers](#)

### Question - 5:

What is PDR - Peer Design Review?

#### Ans:

PDR is an acronym. In the world of software QA or testing, it stands for "peer design review", informally known as "peer review".

[View All Answers](#)

**Question - 6:**

What is waiver?

**Ans:**

In software QA, a waiver is an authorization to accept software that has been submitted for inspection, found to depart from specified requirements, but is nevertheless considered suitable for use "as is", or after rework by an approved method.

[View All Answers](#)

**Question - 7:**

What is virtual address?

**Ans:**

In virtual storage systems, virtual addresses are assigned to auxiliary storage locations. The use of virtual addresses allow those locations to be accessed as though they were part of the main storage.

[View All Answers](#)

**Question - 8:**

What is version description document (VDD)?

**Ans:**

Version description document (VDD) is a document that accompanies and identifies a given version of a software product. Typically the VDD includes the description and identification of the software, identification of the changes incorporated into this version, and the installation and operating information unique to this version of the software.

[View All Answers](#)

**Question - 9:**

What is A document version?

**Ans:**

A document version is an initial release (or complete re-release) of a document, as opposed to a revision resulting from issuing change pages to a previous release.

[View All Answers](#)

**Question - 10:**

What is A variant?

**Ans:**

"Variants" are versions of a program. Variants result from the application of software diversity.

[View All Answers](#)

**Question - 11:**

What is Variable trace?

**Ans:**

1. "Variable trace" is a (computer) record of the names and the values of variables accessed and/or changed during the execution of a computer program.
2. "Value trace" is same as variable trace. It is a (computer) record of the names and values of variables accessed and/or changed during the execution of a computer program.

[View All Answers](#)

**Question - 12:**

What is a utility?

**Ans:**

"Utility" is a software tool designed to perform some frequently used support function. For example, one utility is a program to print files.

[View All Answers](#)

**Question - 13:**

What is Interface Analysis?

**Ans:**

Checks the interfaces between program elements for consistency and adherence to predefined rules or axioms.

[View All Answers](#)

**Question - 14:**

What is a user guide?

**Ans:**

The "user guide" is the same as the user manual. The user guide is a document that presents information necessary to employ a system or component to obtain the desired results. Typically, what is described are system and component capabilities, limitations, options, permitted inputs, expected outputs, error messages, and special instructions.

[View All Answers](#)

**Question - 15:**

What is A user friendly software?

**Ans:**

A computer program is "user friendly", when it is designed with ease of use, as one of the primary objectives of its design.

[View All Answers](#)

**Question - 16:**

What is User documentation?

**Ans:**

"User documentation" is a document that describes the way a software product or system should be used to obtain the desired results.

[View All Answers](#)

**Question - 17:**

What is Upwardly compatible software?

**Ans:**

"Upwardly compatible software" is software that is compatible with a later or more complex version of itself. For example, an upwardly compatible software is able to handle files created by a later version of itself.

[View All Answers](#)

**Question - 18:**

What is CMM and CMMI?

**Ans:**

CMM = 'Capability Maturity Model', now called the CMMI ('Capability Maturity Model Integration'), developed by the SEI. It's a model of 5 levels of process 'maturity' that determine effectiveness in delivering quality software. It is geared to large organizations such as large U.S. Defense Department contractors. However, many of the QA processes involved are appropriate to any organization, and if reasonably applied can be helpful. Organizations can receive CMMI ratings by undergoing assessments by qualified auditors.

Level 1 - characterized by chaos, periodic panics, and heroic efforts required by individuals to successfully complete projects. Few if any processes in place; successes may not be repeatable.

Level 2 - software project tracking, requirements management, realistic planning, and configuration management processes are in place; successful practices can be repeated.

Level 3 - standard software development and maintenance processes are integrated throughout an organization; a Software Engineering Process Group is in place to oversee software processes, and training programs are used to ensure understanding and compliance.

Level 4 - metrics are used to track productivity, processes, and products. Project performance is predictable, and quality is consistently high.

Level 5 - the focus is on continuous process improvement. The impact of new processes and technologies can be predicted and effectively implemented when required.

Perspective on CMM ratings: During 1997-2001, 1018 organizations were assessed. Of those, 27% were rated at Level 1, 39% at 2, 23% at 3, 6% at 4, and 5% at 5. (For ratings during the period 1992-96, 62% were at Level 1, 23% at 2, 13% at 3, 2% at 4, and 0.4% at 5.) The median size of organizations was 100 software engineering/maintenance personnel; 32% of organizations were U.S. federal contractors or agencies. For those rated at Level 1, the most problematical key process area was in Software Quality Assurance.

[View All Answers](#)

**Question - 19:**

Explain Configuration management?

**Ans:**

Configuration management (CM) covers the tools and processes used to control, coordinate and track code, requirements, documentation, problems, change requests, designs, tools, compilers, libraries, patches, changes made to them and who makes the changes. Rob Davis has had experience with a full range of CM tools and concepts, and can easily adapt to your software tool and process needs.

[View All Answers](#)

**Question - 20:**

What is A Test Configuration Manager?

**Ans:**

Test Configuration Managers maintain test environments, scripts, software and test data. Depending on the project, one person may wear more than one hat. For instance, Test Engineers may also wear the hat of a Test Configuration Manager.

[View All Answers](#)

**Question - 21:**

What is A Database Administrator?

**Ans:**

Test Build Managers, System Administrators and Database Administrators deliver current software versions to the test environment, install the application's software and apply software patches, to both the application and the operating system, set-up, maintain and back up test environment hardware. Depending on the project, one person may wear more than one hat. For instance, a Test Engineer may also wear the hat of a Database Administrator.

[View All Answers](#)

**Question - 22:**

What is a Test Build Manager?



**Ans:**

Test Build Managers deliver current software versions to the test environment, install the application's software and apply software patches, to both the application and the operating system, set-up, maintain and back up test environment hardware.

Depending on the project, one person may wear more than one hat. For instance, a Test Engineer may also wear the hat of a Test Build Manager.

[View All Answers](#)

**Question - 23:**

What is QA engineer?

**Ans:**

QA engineers, are test engineers but we do more than just testing. Good QA engineers understand the entire software development process and how it fits into the business approach and the goals of the organization. Communication skills and the ability to understand various sides of issues are important. We, QA engineers, are successful if people listen to us, if people use our tests, if people think that we're useful, and if we're happy doing our work. I would love to see QA departments staffed with experienced software developers who coach development teams to write better code. But I've never seen it. Instead of coaching, we, QA engineers, tend to be process people.

[View All Answers](#)

**Question - 24:**

What is Process and procedures?

**Ans:**

Detailed and well-written processes and procedures ensure the correct steps are being executed to facilitate a successful completion of a task. They also ensure a process is repeatable.

[View All Answers](#)

**Question - 25:**

Test report?

**Ans:**

A document describing the conduct and results of the testing carried out for a system or system component.

[View All Answers](#)

**Question - 26:**

What is Test Objective?

**Ans:**

An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation.

[View All Answers](#)

**Question - 27:**

What is Test Executive?

**Ans:**

Another term for test harness.

[View All Answers](#)

**Question - 28:**

What is Technical Review?

**Ans:**

A review that refers to content of the technical material being reviewed.

[View All Answers](#)

**Question - 29:**

Explain Test plan?

**Ans:**

1. Documentation specifying the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, responsibilities, required, resources, and any risks requiring contingency planning.

or

A formal or informal plan to be followed to assure the controlled testing of the product under test.

2. A software project test plan is a document that describes the objectives, scope, approach and focus of a software testing effort. The process of preparing a test plan is a useful way to think through the efforts needed to validate the acceptability of a software product. The completed document will help people outside the test group understand the why and how of product validation. It should be thorough enough to be useful, but not so thorough that none outside the test group will be able to read it.

[View All Answers](#)

**Question - 30:**

What is Test log?

**Ans:**



A chronological record of all relevant details about the execution of a test.

[View All Answers](#)

### Question - 31:

What is Test incident report?

#### Ans:

A document reporting on any event that occurs during testing that requires further investigation.

[View All Answers](#)

### Question - 32:

What is Test documentation?

#### Ans:

Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report.

[View All Answers](#)

### Question - 33:

Explain Good design?

#### Ans:

Design could mean to many things, but often refers to functional design or internal design. Good functional design is indicated by software functionality can be traced back to customer and end-user requirements. Good internal design is indicated by software code whose overall structure is clear, understandable, easily modifiable and maintainable; is robust with sufficient error handling and status logging capability; and works correctly when implemented.

[View All Answers](#)

### Question - 34:

What is Test scenario?

#### Ans:

The terms "test scenario" and "test case" are often used synonymously. Test scenarios are test cases or test scripts, and the sequence in which they are to be executed. Test scenarios are test cases that ensure that all business process flows are tested from end to end. Test scenarios are independent tests, or a series of tests that follow each other, where each of them dependent upon the output of the previous one. Test scenarios are prepared by reviewing functional requirements, and preparing logical groups of functions that can be further broken into test procedures. Test scenarios are designed to represent both typical and unusual situations that may occur in the application. Test engineers define unit test requirements and unit test scenarios. Test engineers also execute unit test scenarios. It is the test team that, with assistance of developers and clients, develops test scenarios for integration and system testing. Test scenarios are executed through the use of test procedures or scripts. Test procedures or scripts define a series of steps necessary to perform one or more test scenarios. Test procedures or scripts may cover multiple test scenarios.

[View All Answers](#)

### Question - 35:

Explain Test case?

#### Ans:

1. Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.

A test case is a document that describes an input, action, or event and an expected response, to determine if a feature of an application is working correctly. A test case should contain particulars such as test case identifier, test case name, objective, test conditions/setup, input data requirements, steps, and expected results.

Note that the process of developing test cases can help find problems in the requirements or design of an application, since it requires completely thinking through the operation of the application. For this reason, it's useful to prepare test cases early in the development cycle if possible.

or

The definition of test case differs from company to company, engineer to engineer, and even project to project. A test case usually includes an identified set of information about observable states, conditions, events, and data, including inputs and expected outputs.

2. A test case is a document that describes an input, action, or event and its expected result, in order to determine if a feature of an application is working correctly. A test case should contain particulars such as a...

- \* Test case identifier;
- \* Test case name;
- \* Objective;
- \* Test conditions/setup;
- \* Input data requirements/steps, and
- \* Expected results.

Please note, the process of developing test cases can help find problems in the requirements or design of an application, since it requires you to completely think through the operation of the application. For this reason, it is useful to prepare test cases early in the development cycle, if possible.

[View All Answers](#)

### Question - 36:

What is System Test?

#### Ans:

An activity in which a system or component is executed under specified conditions, the results are observed or recorded and an evaluation is made of some aspect of the system or component.

[View All Answers](#)

### Question - 37:



What is Software Review?

**Ans:**

An evaluation of software elements to ascertain discrepancies from planned results and to recommend improvement. This evaluation follows a formal process.

[View All Answers](#)

**Question - 38:**

What is Software Risk?

**Ans:**

A measure of the probability and severity of undesired effects. Often taken as the simple product of probability and consequence.

[View All Answers](#)

**Question - 39:**

What is Software fault?

**Ans:**

Software faults are hidden programming errors. Software faults are errors in the correctness of the semantics of computer programs.

[View All Answers](#)

**Question - 40:**

What is System?

**Ans:**

A collection of people, machines, and methods organized to accomplish a set of specified functions.

[View All Answers](#)

**Question - 41:**

What is Software failure?

**Ans:**

Software failure occurs when the software does not do what the user expects to see.

[View All Answers](#)

**Question - 42:**

What is Exception?

**Ans:**

An event that causes suspension of normal program execution. Types include addressing exception, data exception, operation exception, overflow exception, protection exception, and underflow exception.

[View All Answers](#)

**Question - 43:**

What is Error guessing?

**Ans:**

Test data selection technique. The selection criterion is to pick values that seem likely to cause errors.

[View All Answers](#)

**Question - 44:**

What is Crash?

**Ans:**

The sudden and complete failure of a computer system or component.

[View All Answers](#)

**Question - 45:**

Explain Inspection?

**Ans:**

An inspection is more formalized than a 'walkthrough', typically with 3-8 people including a moderator, reader, and a recorder to take notes. The subject of the inspection is typically a document such as a requirements spec or a test plan, and the purpose is to find problems and see what's missing, not to fix anything. Attendees should prepare for this type of meeting by reading thru the document; most problems will be found during this preparation. The result of the inspection meeting should be a written report. Thorough preparation for inspections is difficult, painstaking work, but is one of the most cost effective methods of ensuring quality. Employees who are most skilled at inspections are like the 'eldest brother' in the parable in 'Why is it often hard for management to get serious about quality assurance?'. Their skill may have low visibility but they are extremely valuable to any software development organization, since bug prevention is far more cost-effective than bug detection.

or

1) A formal evaluation technique in which software requirements, design, or code are examined in detail by a person or group other than the author to detect faults, violations of development standards, and other problems. 2) A quality improvement process for written material that consists of two dominant components: product (document) improvement and process improvement (document production and inspection). Instrument: To install or insert devices or instructions into hardware or software to monitor the operation of a system or component.



An inspection is a formal meeting, more formalized than a walk-through and typically consists of 3-10 people including a moderator, reader (the author of whatever is being reviewed) and a recorder (to make notes in the document). The subject of the inspection is typically a document, such as a requirements document or a test plan. The purpose of an inspection is to find problems and see what is missing, not to fix anything. The result of the meeting should be documented in a written report. Attendees should prepare for this type of meeting by reading through the document, before the meeting starts; most problems are found during this preparation. Preparation for inspections is difficult, but is one of the most cost-effective methods of ensuring quality, since bug prevention is more cost effective than bug detection.

[View All Answers](#)

### Question - 46:

What is Bottom-Up Strategy?

#### Ans:

Bottom-up approach, as the name suggests, is the opposite of the Top-down method.

This process starts with building and testing the low level modules first, working its way up the hierarchy.

Because the modules at the low levels are very specific, we may need to combine several of them into what is sometimes called a cluster or build in order to test them properly.

Then to test these builds, a test driver has to be written and put in place.

The advantage of Bottom-up integration is that there is no need for program stubs as we start developing and testing with the actual modules.

Starting at the bottom of the hierarchy also means that the critical modules are usually build first and therefore any errors in these modules are discovered early in the process.

As with Top-down integration, there are some drawbacks to this procedure.

In order to test the modules we have to build the test drivers which are more complex than stubs. And in addition to that they themselves have to be tested. So more effort is required.

A major disadvantage to Bottom-up integration is that no working model can be presented or tested until many modules have been built.

[View All Answers](#)

### Question - 47:

Explain bug life cycle?

#### Ans:

Bug life cycles are similar to software development life cycles. At any time during the software development life cycle errors can be made during the gathering of requirements, requirements analysis, functional design, internal design, documentation planning, document preparation, coding, unit testing, test planning, integration, testing, maintenance, updates, re-testing and phase-out.

Bug life cycle begins when a programmer, software developer, or architect makes a mistake, creates an unintentional software defect, i.e. bug, and ends when the bug is fixed, and the bug is no longer in existence.

What should be done after a bug is found? When a bug is found, it needs to be communicated and assigned to developers that can fix it. After the problem is resolved, fixes should be re-tested.

Additionally, determinations should be made regarding requirements, software, hardware, safety impact, etc., for regression testing to check the fixes didn't create other problems elsewhere.

If a problem-tracking system is in place, it should encapsulate these determinations. A variety of commercial, problem-tracking, management software tools are available. These tools, with the detailed input of software test engineers, will give the team complete information so developers can understand the bug, get an idea of its severity, reproduce it and fix it.

[View All Answers](#)

### Question - 48:

Explain Equivalence Partitioning?

#### Ans:

Input data of a program is divided into different categories so that test cases can be developed for each category of input data. The goal of equivalence partitioning is to come out with test cases so that errors are uncovered and test cases can be carried out more efficiently. The different categories of input data are called Equivalence Classes.

[View All Answers](#)

### Question - 49:

Explain Boundary value analysis?

#### Ans:

A selection technique in which test data are chosen to lie along "boundaries" of the input domain [or output range] classes, data structures, procedure parameters, etc. Choices often include maximum, minimum, and trivial values or parameters. This technique is often called stress testing.

or

A test data selection technique in which values are chosen to lie along data extremes. Boundary values include maximum, mini-mum, just inside/outside boundaries, typical values, and error values.

[View All Answers](#)

### Question - 50:

Explain Audit?

#### Ans:

(1)An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria.

(2)To conduct an independent review and examination of system records and activities in order to test the adequacy and effectiveness of data security and data integrity procedures, to ensure compliance with established policy and operational procedures, and to recommend any necessary changes.

[View All Answers](#)

### Question - 51:



What is software life cycle?

**Ans:**

Software life cycle begins when a software product is first conceived and ends when it is no longer in use. It includes aspects such as initial concept, requirements analysis, functional design, internal design, documentation planning, test planning, coding, document preparation, integration, testing, maintenance, updates, retesting, phase-out, and other aspects.

[View All Answers](#)

**Question - 52:**

What is software verification?

**Ans:**

In general the demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

[View All Answers](#)

**Question - 53:**

What is Validation and verification testing?

**Ans:**

Used as an entity to define a procedure of review, analysis, and testing throughout the software life cycle to discover errors, determine functionality, and ensure the production of quality software.

[View All Answers](#)

**Question - 54:**

Explain Walk-through?

**Ans:**

A walk-through (in software QA) is an informal meeting for evaluation or informational purposes. A walk-through is also a process at an abstract level. It's the process of inspecting software code by following paths through the code (as determined by input conditions and choices made along the way).

The purpose of code walk-throughs (in software development) is to ensure the code fits the purpose. Walk-throughs also offer opportunities to assess an individual's or team's competency.

A walk-through is also a static analysis technique in which a programmer leads participants through a segment of documentation or code, and the participants ask questions, and make comments about possible errors, violations of development standards, and other issues.

[View All Answers](#)

**Question - 55:**

What is SEI - Software Engineering Institute?

**Ans:**

SEI = 'Software Engineering Institute' at Carnegie-Mellon University; initiated by the U.S. Defense Department to help improve software development processes.

[View All Answers](#)

**Question - 56:**

What is IEEE - Institute of Electrical and Electronics Engineers?

**Ans:**

IEEE = 'Institute of Electrical and Electronics Engineers' - among other things, creates standards such as 'IEEE Standard for Software Test Documentation' (IEEE/ANSI Standard 829), 'IEEE Standard of Software Unit Testing (IEEE/ANSI Standard 1008)', 'IEEE Standard for Software Quality Assurance Plans' (IEEE/ANSI Standard 730), and others.

[View All Answers](#)

**Question - 57:**

What is Syntax?

**Ans:**

1) The relationship among characters or groups of characters independent of their meanings or the manner of their interpretation and use;

2) the structure of expressions in a language; and

3) the rules governing the structure of the language.

[View All Answers](#)

**Question - 58:**

Explain Process?

**Ans:**

The work effort that produces a product. This includes efforts of people and equipment guided by policies, standards, and procedures.

[View All Answers](#)

**Question - 59:**

What is Procedure?

**Ans:**

The step-by-step method followed to ensure that standards are met.



[View All Answers](#)

**Question - 60:**

What is Problem?

**Ans:**

Any deviation from defined standards. Same as defect.

[View All Answers](#)

**Question - 61:**

What is Peer Reviews?

**Ans:**

A methodical examination of software work products by the producer's peers to identify defects and areas where changes are needed.

[View All Answers](#)

**Question - 62:**

What is Path Analysis?

**Ans:**

Program analysis performed to identify all possible paths through a program, to detect incomplete paths, or to discover portions of the program that are not on any path.

[View All Answers](#)

**Question - 63:**

Explain Metric?

**Ans:**

A measure of the extent or degree to which a product possesses and exhibits a certain quality, property, or attribute.

[View All Answers](#)

**Question - 64:**

What is Non-intrusive Testing?

**Ans:**

Testing that is transparent to the software under test; i.e., testing that does not change the timing or processing characteristics of the software under test from its behavior in a real environment. Usually involves additional hardware that collects timing or processing information and processes that information on another platform.

[View All Answers](#)

**Question - 65:**

What is Control Structure Testing?

**Ans:**

Because basis path testing alone is insufficient, other techniques should be utilized.

Condition testing can be utilized to design test cases which examine the logical conditions in a program. It focuses on all conditions in the program and includes testing of both relational expressions and arithmetic expressions.

This can be accomplished using branch testing and/or domain testing methods. Branch testing executes both true and false branches of a condition. Domain testing utilizes values on the left-hand side of the relation by making them greater than, equal to and less than the right-hand side value. This method tests both values and the relation operators in the expression. Data flow testing method is effective for error protection because it is based on the relationship between statements in the program according to the definition and uses of variables.

Loop testing method concentrates on validity of the loop structures.

[View All Answers](#)

**Question - 66:**

What is Random Testing?

**Ans:**

This is one of methods used to exercise a class. It is based on developing a random test sequence that tries the minimum number of operations typical to the behavior of the class.

[View All Answers](#)

**Question - 67:**

What is Intrusive Testing?

**Ans:**

Testing that collects timing and processing information during program execution that may change the behavior of the software from its behavior in a real environment. Usually involves additional code embedded in the software being tested or additional processes running concurrently with software being tested on the same platform.

[View All Answers](#)



### Question - 68:

What is Walk-through?

#### Ans:

A walk-through is an informal meeting for evaluation or informational purposes.

[View All Answers](#)

### Question - 69:

What is Parameter?

#### Ans:

In software QA or software testing, a parameter is an item of information - such as a name, number, or selected option - that is passed to a program, by a user or another program. By definition, in software, a parameter is a value on which something else depends. Any desired numerical value may be given as a parameter. In software development, we use parameters when we want to allow a specified range of variables. We use parameters when we want to differentiate behavior or pass input data to computer programs or their subprograms. Thus, when we are testing, the parameters of the test can be varied to produce different results, because parameters do affect the operation of the program receiving them.

Example 1: We use a parameter, such as temperature, that defines a system. In this definition, it is temperature that defines the system and determines its behavior.

Example 2: In the definition of function  $f(x) = x + 10$ ,  $x$  is a parameter. In this definition,  $x$  defines the  $f(x)$  function and determines its behavior. Thus, when we are testing,  $x$  can be varied to make  $f(x)$  produce different values, because the value of  $x$  does affect the value of  $f(x)$ .

When parameters are passed to a function subroutine, they are called arguments.

[View All Answers](#)

### Question - 70:

What is Data validity?

#### Ans:

Data validity is the correctness and reasonableness of data. Reasonableness of data means that, for example, account numbers falling within a range, numeric data being all digits, dates having a valid month, day and year, and spelling of proper names. Data validity errors are probably the most common, and most difficult to detect (data-related) errors.

What causes data validity errors? Data validity errors are usually caused by incorrect data entries, when a large volume of data is entered in a short period of time. For example, a data entry operator enters 12/25/2010 as 13/25/2010, by mistake, and this data is therefore invalid. How can you reduce data validity errors? You can use one of the following two, simple field validation techniques.

Technique 1: If the date field in a database uses the MM/DD/YYYY format, then you can use a program with the following two data validation rules: "MM" should not exceed "12", and "DD" should not exceed "31".

Technique 2: If the original figures do not seem to match the ones in the database, then you can use a program to validate data fields. You can compare the sum of the numbers in the database data field to the original sum of numbers from the source. If there is a difference between the two figures, it is an indication of an error in at least one data element.

[View All Answers](#)

### Question - 71:

Explain Exit criteria?

#### Ans:

The "exit criteria" is a checklist, sometimes known as the "PDR sign-off sheet". It is a list of peer design review related tasks that have to be done by the facilitator or attendees of the PDR, either during or near the conclusion of the PDR.

By having a checklist, and by going through the checklist, the facilitator can verify that A) all attendees have inspected all the relevant documents and reports, B) all suggestions and recommendations for each issue have been recorded, and C) all relevant facts of the meeting have been recorded.

The facilitator's checklist includes the following questions:

- \* Have we inspected all the relevant documents, code blocks, or products?
- \* Have we completed all the required checklists?
- \* Have I recorded all the facts relevant to this peer review?
- \* Does anyone have any additional suggestions, recommendations, or comments?
- \* What is the outcome of this peer review?

As the end of the PDR, the facilitator asks the attendees to make a decision as to the outcome of the PDR, i.e. "What is our consensus... are we accepting the design (or document or code)?" Or, "Are we accepting it with minor modifications?" Or, "Are we accepting it after it has been modified and approved through e-mails to the attendees?" Or, "Do we want another peer review?" This is a phase, during which the attendees work as a committee, and the committee's decision is final.

[View All Answers](#)

### Question - 72:

What is Waterfall model?

#### Ans:

Waterfall is a model of the software development process in which the concept phase, requirements phase, design phase, implementation phase, test phase, installation phase, and checkout phase are performed in that order, probably with overlap, but with little or no iteration.

[View All Answers](#)

### Question - 73:

Explain virtual memory?

#### Ans:

Virtual memory relates to virtual storage. In virtual storage, portions of a user's program and data are placed in auxiliary storage, and the operating system automatically swaps them in and out of main storage as needed.

[View All Answers](#)

### Question - 74:



What is A vertical microinstruction?

**Ans:**

A vertical microinstruction is a microinstruction that specifies one of a sequence of operations needed to carry out a machine language instruction. Vertical microinstructions are short, 12 to 24 bit instructions. They're called vertical because they are normally listed vertically on a page. These 12 to 24 bit microinstructions are required to carry out a single machine language instruction. In addition to vertical microinstructions, there are horizontal and diagonal microinstructions as well.

[View All Answers](#)

**Question - 75:**

What is VDD - version description document?

**Ans:**

"VDD" is an acronym that stands for "version description document".

[View All Answers](#)

**Question - 76:**

What is A software version?

**Ans:**

A software version is an initial release (or re-release) of a software associated with a complete compilation (or recompilation) of the software.

[View All Answers](#)

**Question - 77:**

Explain A variable?

**Ans:**

"Variables" are data items in a program whose values can change. There are local and global variables. One example is a variable we have named "capacitor\_voltage\_10000", where "capacitor\_voltage\_10000" can be any whole number between -10000 and +10000.

[View All Answers](#)

**Question - 78:**

What is Utilization?

**Ans:**

"Utilization" is the ratio of time a system is busy (i.e. working for us), divided by the time it is available. For example, if a system was available for 160 hours and busy for 40 hours, then utilization was  $(40/160 =) 25$  per cent. Utilization is a useful measure in evaluating computer performance.

[View All Answers](#)

**Question - 79:**

What is A user interface?

**Ans:**

"User interface" is the interface between a human user and a computer system. It enables the passage of information between a human user and hardware or software components of a computer system.

[View All Answers](#)

**Question - 80:**

Explain Interface?

**Ans:**

A shared boundary. An interface might be a hardware component to link two devices, or it might be a portion of storage or registers accessed by two or more computer programs.

[View All Answers](#)

**Question - 81:**

What is A user friendly document?

**Ans:**

A document is user friendly, when it is designed and written with ease of use, as one of the primary objectives of its design.

[View All Answers](#)

**Question - 82:**

What is A user manual?

**Ans:**

A "user manual" is a document that presents information necessary to employ software or a system to obtain the desired results. Typically, what is described are system and component capabilities, limitations, options, permitted inputs, expected outputs, error messages, and special instructions.

[View All Answers](#)

**Question - 83:**



What is Upward compression?

**Ans:**

In software design, "upward compression" means a form of demodularization in which a subordinate module is copied into the body of a superior module.

[View All Answers](#)

**Question - 84:**

What is Up time?

**Ans:**

"Up time" is the time period when a system is operational and in service. Up time is the sum of busy time and idle time. For example, if, out of 168 hours, a system has been busy for 50 hours, idle for 110 hours, and down for 8 hours, then the busy time is 50 hours, idle time is 110 hours, and up time is (110 + 50 =) 160 hours.

[View All Answers](#)

**Question - 85:**

What is Software configuration management?

**Ans:**

Software Configuration Management (SCM) is the control and the recording of changes that are made to the software and documentation throughout the software development life cycle (SDLC).

SCM covers the tools and processes used to control, coordinate and track code, requirements, documentation, problems, change requests, designs, tools, compilers, libraries, patches, and changes made to them, and to keep track of who makes the changes.

[View All Answers](#)

**Question - 86:**

Explain the Requirements?

**Ans:**

Requirement specifications are important and one of the most reliable methods of insuring problems in a complex software project is to have poorly documented requirement specifications. Requirements are the details describing an application's externally perceived functionality and properties. Requirements should be clear, complete, reasonably detailed, cohesive, attainable and testable. A non-testable requirement would be, for example, "user-friendly", which is too subjective. A testable requirement would be something such as, "the product shall allow the user to enter their previously-assigned password to access the application". Care should be taken to involve all of a project's significant customers in the requirements process. Customers could be in-house or external and could include end-users, customer acceptance test engineers, testers, customer contract officers, customer management, future software maintenance engineers, salespeople and anyone who could later derail the project. If his/her expectations aren't met, they should be included as a customer, if possible. In some organizations, requirements may end up in high-level project plans, functional specification documents, design documents, or other documents at various levels of detail. No matter what they are called, some type of documentation with detailed requirements will be needed by test engineers in order to properly plan and execute tests. Without such documentation there will be no clear-cut way to determine if a software application is performing correctly.

[View All Answers](#)

**Question - 87:**

What is A Technical Analyst?

**Ans:**

Technical Analysts perform test assessments and validate system/functional test requirements. Depending on the project, one person may wear more than one hat. For instance, Test Engineers may also wear the hat of a Technical Analyst.

[View All Answers](#)

**Question - 88:**

What is A System Administrator?

**Ans:**

Test Build Managers, System Administrators, Database Administrators deliver current software versions to the test environment, install the application's software and apply software patches, to both the application and the operating system, set-up, maintain and back up test environment hardware.

Depending on the project, one person may wear more than one hat. For instance, a Test Engineer may also wear the hat of a System Administrator.

[View All Answers](#)

**Question - 89:**

What is Test/QA Team Lead?

**Ans:**

The Test/QA Team Lead coordinates the testing activity, communicates testing status to management and manages the test team.

[View All Answers](#)

**Question - 90:**

Test engineer?

**Ans:**

We, test engineers are engineers who specialize in testing. We, test engineers, create test cases, procedures, scripts and generate data. We execute test procedures and scripts, analyze standards of measurements, evaluate results of system/integration/regression testing.

We also...

\* Speed up the work of your development staff;

\* Reduce your organization's risk of legal liability;



- \* Give you the evidence that your software is correct and operates properly;
- \* Improve your problem tracking and reporting;
- \* Maximize the value of your software;
- \* Maximize the value of the devices that use it;
- \* Assure the successful launch of your product by discovering bugs and design flaws, before users get discouraged, before shareholders lose their cool and before employees get bogged down;
- \* Help the work of your development staff, so the development team can devote its time to build up your product;
- \* Promote continual improvement;
- \* Provide documentation required by FDA, FAA, other regulatory agencies and your customers;
- \* Save money by discovering defects 'early' in the design process, before failures occur in production, or in the field;
- \* Save the reputation of your company by discovering bugs and design flaws; before bugs and design flaws damage the reputation of your company.

[View All Answers](#)

### Question - 91:

What is Test result analyzer?

#### Ans:

A software tool used to test output data reduction, formatting, and printing.

[View All Answers](#)

### Question - 92:

What is Benchmark?

#### Ans:

A Benchmark is a standard to measure against. If you benchmark an application, all future application changes will be tested and compared against the benchmarked application.

[View All Answers](#)

### Question - 93:

Explain Test Harness?

#### Ans:

A software tool that enables the testing of software components that links test capabilities to perform specific tests, accept program inputs, simulate missing components, compare actual outputs with expected outputs to determine correctness, and report discrepancies.

[View All Answers](#)

### Question - 94:

What is Test Development?

#### Ans:

The development of anything required to conduct testing. This may include test requirements (objectives), strategies, processes, plans, software, procedures, cases, documentation, etc.

[View All Answers](#)

### Question - 95:

Explain Test Procedure?

#### Ans:

The formal or informal procedure that will be followed to execute a test. This is usually a written document that allows others to execute the test with a minimum of training.

[View All Answers](#)

### Question - 96:

What is Test phase?

#### Ans:

The period of time in the software life cycle in which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied.

[View All Answers](#)

### Question - 97:

What is Test item?

#### Ans:

A software item which is the object of testing.

[View All Answers](#)

### Question - 98:

Explain Test driver?

#### Ans:

A software module used to invoke a module under test and, often, provide test inputs, control and monitor execution, and report test results.



[View All Answers](#)

**Question - 99:**

What is Test design?

**Ans:**

Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests.

[View All Answers](#)

**Question - 100:**

What is Scenario-based Testing?

**Ans:**

This form of testing concentrates on what the user does. It basically involves capturing the user actions and then simulating them and similar actions during the test. These tests tend to find interaction type of errors

[View All Answers](#)

**Question - 101:**

What is Test case generator?

**Ans:**

A software tool that accepts as input source code, test criteria, specifications, or data structure definitions; uses these inputs to generate test input data; and, sometimes, determines expected results.

[View All Answers](#)

**Question - 102:**

Explain Testability?

**Ans:**

- (1) The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.
- (2) The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met.

[View All Answers](#)

**Question - 103:**

Explain Static analysis?

**Ans:**

- (1) Analysis of a program that is performed without executing the program.
- (2) The process of evaluating a system or component based on its form, structure, content, documentation. Contrast with dynamic analysis.

[View All Answers](#)

**Question - 104:**

What is Risk Assessment?

**Ans:**

A comprehensive evaluation of the risk and its associated impact.

[View All Answers](#)

**Question - 105:**

What is Review?

**Ans:**

A process or meeting during which a work product or set of work products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Types include code review, design review, formal qualification review, requirements review, test readiness review. Contrast with audit, inspection.

[View All Answers](#)

**Question - 106:**

What is System Simulation?

**Ans:**

Another name for prototyping.

[View All Answers](#)

**Question - 107:**

Explain Fault?

**Ans:**

An incorrect step, process, or data definition in a computer program which causes the program to perform in an unintended or unanticipated manner.



[View All Answers](#)

### Question - 108:

Explain Failure?

#### Ans:

The inability of a system or component to perform its required functions within specified performance requirements.

[View All Answers](#)

### Question - 109:

What is Error seeding?

#### Ans:

error seeding. (IEEE) The process of intentionally adding known faults to those already in a computer program for the purpose of monitoring the rate of detection and removal, and estimating the number of faults remaining in the program. Contrast with mutation analysis.

[View All Answers](#)

### Question - 110:

What is Criticality?

#### Ans:

The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system. Syn: severity.

[View All Answers](#)

### Question - 111:

What is Coverage analysis?

#### Ans:

Determining and assessing measures associated with the invocation of program structural elements to determine the adequacy of a test run. Coverage analysis is useful when attempting to execute each statement, branch, path, or iterative structure in a program. Tools that capture this data and provide reports summarizing relevant information have this feature.

[View All Answers](#)

### Question - 112:

What is Big-Bang Strategy?

#### Ans:

Big-Bang approach is very simple in its philosophy where basically all the modules or builds are constructed and tested independently of each other and when they are finished, they are all put together at the same time.

The main advantage of this approach is that it is very quick as no drivers or stubs are needed, thus cutting down on the development time.

However, as with anything that is quickly slapped together, this process usually yields more errors than the other two. Since these errors have to be fixed and take more time to fix than errors at the module level, this method is usually considered the least effective.

Because of the amount of coordination that is required it is also very demanding on the resources. Another drawback is that there is really nothing to demonstrate until all the modules have been built and integrated.

[View All Answers](#)

### Question - 113:

What is Top-Down Strategy?

#### Ans:

Top down integration is basically an approach where modules are developed and tested starting at the top level of the programming hierarchy and continuing with the lower levels.

It is an incremental approach because we proceed one level at a time. It can be done in either "depth" or "breadth" manner.

- Depth means we proceed from the top level all the way down to the lowest level.

- Breadth, on the other hand, means that we start at the top of the hierarchy and then go to the next level. We develop and test all modules at this level before continuing with another level.

Either way, this testing procedure allows us to establish a complete skeleton of the system or product.

The benefits of Top-down integration are that, having the skeleton, we can test major functions early in the development process.

At the same time we can also test any interfaces that we have and thus discover any errors in that area very early on. But the major benefit of this procedure is that we have a partially working model to demonstrate to the clients and the top management. This of course builds everybody's confidence not only in the development team but also in the model itself. We have something that proves our design was correct and we took the correct approach to implement it.

However, there are some drawbacks to this procedure as well:

Using stubs does not permit all the necessary upward data flow. There is simply not enough data in the stubs to feed back to the calling module.

As a result, the top level modules can not be really tested properly and every time the stubs are replaced with the actual modules, the calling modules should be re-tested for integrity again.

[View All Answers](#)

### Question - 114:

What is Software Bug?

#### Ans:

(1) A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, fault.

(2) A bug is a glitch in computer software or hardware (where something doesn't do what it is supposed to do). Since computers and computer software are very



complicated to design, human beings will make mistakes in the design. Unfortunately, in the rush to market, many of these mistakes are not found until after a product has shipped. This is why fixes (also called patches) are often posted on web sites. When considering the quality of a product, one must consider not only the number of bugs, but also the value of the features of a program, since a feature-rich program is likely to have more bugs than a "plain-vanilla" program. 3)A design flaw that will result in symptoms exhibited by some object (the object under test or some other object) when an object is subjected to an appropriate test.

[View All Answers](#)

### Question - 115:

What is Branch coverage?

#### Ans:

A test coverage criteria which requires that for each decision point each possible branch be executed at least once. Syn: decision coverage. Contrast with condition coverage, multiple condition coverage, path coverage, statement coverage.

[View All Answers](#)

### Question - 116:

What is Boundary value?

#### Ans:

(1)A data value that corresponds to a minimum or maximum input, internal, or output value specified for a system or component.  
(2)A value which lies at, or just inside or just outside a specified range of valid input and output values.

[View All Answers](#)

### Question - 117:

What is SDLC - software development life cycle?

#### Ans:

SDLC is an acronym. It stands for "software development life cycle".

[View All Answers](#)

### Question - 118:

What is Life Cycle?

#### Ans:

The period that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically includes a requirements phase, design phase, implementation (code) phase, test phase, installation and checkout phase, operation and maintenance phase, and a retirement phase.

[View All Answers](#)

### Question - 119:

What is V&V?

#### Ans:

"V&V" is an acronym that stands for verification and validation.

"Validation: are we building the product right"

"Verification: are we building the right product"

Verification and validation (V&V) is a process that helps to determine if the software requirements are complete, correct; and if the software of each development phase fulfills the requirements and conditions imposed by the previous phase; and if the final software complies with the applicable software requirements.

[View All Answers](#)

### Question - 120:

Explain Traceability matrix?

#### Ans:

(2)A matrix that records the relationship between two or more products; e.g., a matrix that records the relationship between the requirements and the design of a given software component. See: traceability, traceability analysis.

(1)Traceability Matrix is for mapping the requirements to Test cases. To verify whether all the test cases covering all the stated requirements or not. The purpose of the Traceability Matrix is to identify all business requirements and to trace each requirement through the project's completion.

[View All Answers](#)

### Question - 121:

What is ANSI - American National Standards Institute?

#### Ans:

ANSI = 'American National Standards Institute', the primary industrial standards body in the U.S.; publishes some software-related standards in conjunction with the IEEE and ASQ (American Society for Quality). Other software development/IT management process assessment methods besides CMMI and ISO 9000 include SPICE, Trillium, TickIT, Bootstrap, ITIL, MOF, and CobiT.

[View All Answers](#)

### Question - 122:

What is ISO - International Organisation for StandardizationO?

#### Ans:

ISO = 'International Organisation for Standardization' - The ISO 9001:2000 standard (which replaces the previous standard of 1994) concerns quality systems that are



assessed by outside auditors, and it applies to many kinds of production and manufacturing organizations, not just software. It covers documentation, design, development, production, testing, installation, servicing, and other processes. The full set of standards consists of: (a)Q9001-2000 - Quality Management Systems: Requirements; (b)Q9000-2000 - Quality Management Systems: Fundamentals and Vocabulary; (c)Q9004-2000 - Quality Management Systems: Guidelines for Performance Improvements. To be ISO 9001 certified, a third-party auditor assesses an organization, and certification is typically good for about 3 years, after which a complete reassessment is required. Note that ISO certification does not necessarily indicate quality products - it indicates only that documented processes are followed. Also see <http://www.iso.ch/> for the latest information. In the U.S. the standards can be purchased via the ASQ web site at <http://e-standards.asq.org/>

[View All Answers](#)

### Question - 123:

What is ISSUES?

#### Ans:

Invariable there will be issues with software testing under both models. This is simply because both environments are dynamic and have to deal with ongoing changes during the life cycle of the project. That means changes in specifications, analysis, design and development. All of these of course affect testing. However, we will concentrate on possible problem areas within the testing strategies and methods. We will examine how these issues pertain to each environment.

[View All Answers](#)

### Question - 124:

What is Supplier?

#### Ans:

An individual or organization that supplies inputs needed to generate a product, service, or information to an end user.

[View All Answers](#)

### Question - 125:

Explain Test Bed?

#### Ans:

- 1) An environment that contains the integral hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test of a logically or physically separate component.
- 2) A suite of test programs used in conducting the test of a component or system.

[View All Answers](#)

### Question - 126:

What is Policy?

#### Ans:

Managerial desires and intents concerning either process (intended objectives) or products (desired attributes).

[View All Answers](#)

### Question - 127:

What is Path Coverage Testing?

#### Ans:

A test method satisfying coverage criteria that each logical path through the program is tested. Paths through the program often are grouped into a finite set of classes; one path from each class is tested.

[View All Answers](#)

### Question - 128:

What is Outputs?

#### Ans:

Products, services, or information supplied to meet end user needs.

[View All Answers](#)

### Question - 129:

Explain Operational Testing?

#### Ans:

Testing performed by the end user on software in its normal operating environment.

[View All Answers](#)

### Question - 130:

Explain Basis Path Testing?

#### Ans:

Basis path testing is a white-box technique. It allows the design and definition of a basis set of execution paths. The test cases created from the basis set allow the program to be executed in such a way as to examine each possible path through the program by executing each statement at least once.

To be able to determine the different program paths, the engineer needs a representation of the logical flow of control. The control structure can be illustrated by a flow graph. A flow graph can be used to represent any procedural design.

Next a metric can be used to determine the number of independent paths. It is called cyclomatic complexity and it provides the number of test cases that have to be designed. This insures coverage of all program statements.



[View All Answers](#)

**Question - 131:**

What is Class Level Methods?

**Ans:**

As mentioned above, a class (and its operations) is the module most concentrated on in OO environments. From here it should expand to other classes and sets of classes. Just like traditional models are tested by starting at the module first and continuing to module clusters or builds and then the whole program

[View All Answers](#)

**Question - 132:**

What is Operational Requirements?

**Ans:**

Qualitative and quantitative parameters that specify the desired operational capabilities of a system and serve as a basis for determining the operational effectiveness and suitability of a system prior to deployment.

[View All Answers](#)

**Question - 133:**

What is Infeasible Path?

**Ans:**

Program statement sequence that can never be executed.

[View All Answers](#)

**Question - 134:**

What is Hybrid Testing?

**Ans:**

A combination of top-down testing combined with bottom-up testing of prioritized or available components.

[View All Answers](#)

**Question - 135:**

What is Heuristics Testing?

**Ans:**

Another term for failure-directed testing.

[View All Answers](#)

**Question - 136:**

What is Flowchart?

**Ans:**

A diagram showing the sequential steps of a process or of a workflow around a product or service. Formal Review: A technical review conducted with the end user, including the types of reviews called for in the standards.

[View All Answers](#)

**Question - 137:**

What is Fault Tree Analysis?

**Ans:**

A form of safety analysis that assesses hardware safety to provide failure statistics and sensitivity analyses that indicate the possible effect of critical failures.

[View All Answers](#)

**Question - 138:**

What is Failure-directed Testing?

**Ans:**

Testing based on the knowledge of the types of errors made in the past that are likely for the system under test.

[View All Answers](#)

**Question - 139:**

What is Execution?

**Ans:**

The process of a computer carrying out an instruction or instructions of a computer. Exhaustive Testing: Executing the program with all possible combinations of values for program variables.

[View All Answers](#)

**Question - 140:**



What is Usability?

**Ans:**  
"Usability" means ease of use; the ease with which a user can learn to operate, prepare inputs for, and interpret the outputs of a software product.

[View All Answers](#)

**Question - 141:**

What is Desk Checking?

**Ans:**  
A form of manual static analysis usually performed by the originator. Source code documentation, etc., is visually checked against requirements and standards.

[View All Answers](#)

**Question - 142:**

Explain Defect?

**Ans:**  
NOTE: Operationally, it is useful to work with two definitions of a defect:  
1) From the producer's viewpoint: a product requirement that has not been met or a product attribute possessed by a product or a function performed by a product that is not in the statement of requirements that define the product.  
2) From the end user's viewpoint: anything that causes end user dissatisfaction, whether in the statement of requirements or not.

[View All Answers](#)

**Question - 143:**

What is Defect Analysis?

**Ans:**  
Using defects as data for continuous quality improvement. Defect analysis generally seeks to classify defects into categories and identify possible causes in order to direct process improvement efforts.

[View All Answers](#)

**Question - 144:**

What is Data Flow Analysis?

**Ans:**  
Consists of the graphical analysis of collections of (sequential) data definitions and reference patterns to determine constraints that can be placed on data values at various points of executing the source program.

[View All Answers](#)

**Question - 145:**

What is Customer (end user)?

**Ans:**  
The individual or organization, internal or external to the producing organization, that receives the product.

[View All Answers](#)

**Question - 146:**

What is Walkthrough?

**Ans:**  
Usually, a step-by-step simulation of the execution of a procedure, as when walking through code, line by line, with an imagined set of inputs. The term has been extended to the review of material that is not procedural, such as data descriptions, reference manuals, specifications, etc.

[View All Answers](#)

**Question - 147:**

What is V- Diagram (model)?

**Ans:**  
a diagram that visualizes the order of testing activities and their corresponding phases of development

[View All Answers](#)

**Question - 148:**

What is Control Chart?

**Ans:**  
A statistical method for distinguishing between common and special cause variation exhibited by processes.

[View All Answers](#)

**Question - 149:**

What is Checksheet?

**Ans:**

A form used to record data as it is gathered.

[View All Answers](#)

**Question - 150:**

What is Cause-and-Effect (Fishbone) Diagram?

**Ans:**

A tool used to identify possible causes of a problem by representing the relationship between some effect and its possible cause.

[View All Answers](#)

**Question - 151:**

What is Affinity Diagram?

**Ans:**

A group process that takes large amounts of language data, such as a list developed by brainstorming, and divides it into categories.

[View All Answers](#)

**Question - 152:**

Explain Cause effect graph?

**Ans:**

A Boolean graph linking causes and effects. The graph is actually a digital-logic circuit (a combinatorial logic network) using a simpler notation than standard electronics notation.

[View All Answers](#)

**Question - 153:**

Explain Glossary of Software QA/Testing?

**Ans:**

A value derived by adding several qualities and dividing the sum by the number of these quantities.

[View All Answers](#)

**Question - 154:**

What is Code review?

**Ans:**

A meeting at which software code is presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Contrast with code audit, code inspection, code walkthrough.

[View All Answers](#)

**Question - 155:**

Explain Validation?

**Ans:**

Establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specifications and quality attributes. Validation typically involves actual testing and takes place after verifications are completed.

Validation (Product Oriented)

Validation is concerned with whether the right functions of the program have been properly implemented, and that this function will properly produce the correct output given some input value.

The process of evaluating software to determine compliance with specified requirements.

Validation ensures that functionality, as defined in requirements, is the intended behavior of the product; validation typically involves actual testing and takes place after verifications are completed.

[View All Answers](#)

**Question - 156:**

Explain Automated testing?

**Ans:**

Automated testing is a formally specified and controlled method of formal testing approach.

or

That part of software testing that is assisted with software tool(s) that does not require operator input, analysis, or evaluation.

[View All Answers](#)

**Question - 157:**

What is Stochastic testing?

**Ans:**

Stochastic testing is the same as "monkey testing", but stochastic testing is a lot more technical sounding name for the same testing process.

Stochastic testing is black box testing, random testing, performed by automated testing tools. Stochastic testing is a series of random tests over time. The software under test typically passes the individual tests, but our goal is to see if it can pass a large number of individual tests.



[View All Answers](#)

**Question - 158:**

What is Partition Testing?

**Ans:**

This method categorizes the inputs and outputs of a class in order to test them separately. This minimizes the number of test cases that have to be designed. To determine the different categories to test, partitioning can be broken down as follows:

- State-based partitioning - categorizes class operations based on how they change the state of a class
- Attribute-based partitioning - categorizes class operations based on attributes they use
- Category-based partitioning - categorizes class operations based on the generic function the operations perform

[View All Answers](#)

**Question - 159:**

What is Top-down Testing?

**Ans:**

An integration testing technique that tests the high-level components first using stubs for lower-level called components that have not yet been integrated and that stimulate the required actions of those components.

[View All Answers](#)

**Question - 160:**

What is Reliability testing?

**Ans:**

Reliability testing is designing reliability test cases, using accelerated reliability techniques - for example step-stress, test / analyze / fix, and continuously increasing stress testing techniques - AND testing units or systems to failure, in order to obtain raw failure time data for product life analysis.

The purpose of reliability testing is to determine product reliability, and to determine whether the software meets the customer's reliability requirements.

In the system test phase, or after the software is fully developed, one reliability testing technique we use is a test / analyze / fix technique, where we couple reliability testing with the removal of faults.

When we identify a failure, we send the software back to the developers, for repair. The developers build a new version of the software, and then we do another test iteration.

Then we track failure intensity - for example failures per transaction, or failures per hour - in order to guide our test process, and to determine the feasibility of the software release, and to determine whether the software meets the customer's reliability requirements.

[View All Answers](#)

**Question - 161:**

What is Comparison testing?

**Ans:**

Comparison testing is testing that compares software weaknesses and strengths to those of competitors' products.

[View All Answers](#)

**Question - 162:**

What is Disaster recovery testing?

**Ans:**

"Disaster recovery testing" is testing how well a system recovers from disasters, crashes, hardware failures, or other catastrophic problems.

[View All Answers](#)

**Question - 163:**

What is Security/penetration testing?

**Ans:**

Security/penetration testing is testing how well the system is protected against unauthorized internal or external access, or willful damage. This type of testing usually requires sophisticated testing techniques.

[View All Answers](#)

**Question - 164:**

Explain Sanity testing?

**Ans:**

Sanity testing is performed whenever cursory testing is sufficient to prove the application is functioning according to specifications. This level of testing is a subset of regression testing.

It normally includes a set of core tests of basic GUI functionality to demonstrate connectivity to the database, application servers, printers, etc.

[View All Answers](#)

**Question - 165:**

What is Bottom-up testing?

**Ans:**

Bottom-up testing is a technique of integration testing. A test engineer creates and uses test drivers for components that have not yet been developed, because, with



bottom-up testing, low-level components are tested first. The objective of bottom-up testing is to call low-level components first, for testing purposes.

or

An integration testing technique that tests the low-level components first using test drivers for those components that have not yet been developed to call the low-level components for test.

[View All Answers](#)

**Question - 166:**

Explain Ad hoc testing?

**Ans:**

Ad hoc testing is a testing approach; it is the least formal testing approach.

[View All Answers](#)

**Question - 167:**

What is OO Function Testing?

**Ans:**

Function testing of OO software is no different than validation testing of procedural software. Client involvement is usually part of this testing stage. In OO environment use cases may be used. These are basically descriptions of how the system is to be used.

[View All Answers](#)

**Question - 168:**

What is Closed box testing?

**Ans:**

Closed box testing is same as black box testing. Black box testing a type of testing that considers only externally visible behavior. Black box testing considers neither the code itself, nor the "inner workings" of the software.

[View All Answers](#)

**Question - 169:**

Explain Usability Testing?

**Ans:**

1. Tests designed to evaluate the machine/user interface. Are the communication device(s) designed in a manner such that the information is displayed in a understandable fashion enabling the operator to correctly interact with the system?
2. Usability testing is testing for 'user-friendliness'. Clearly this is subjective and depends on the targeted end-user or customer. User interviews, surveys, video recording of user sessions and other techniques can be used. Programmers and developers are usually not appropriate as usability testers.

[View All Answers](#)

**Question - 170:**

Explain Structural Testing?

**Ans:**

(1) Testing that takes into account the internal mechanism [structure] of a system or component. Types include branch testing, path testing, statement testing.

(2) Testing to insure each program statement is made to execute during testing and that each program statement performs its intended function.

Structural testing is white box testing, not black box testing, since black boxes are considered opaque and do not permit visibility into the code.

Structural testing is also known as clear box testing, also known as glass box testing.

Structural testing is a way to test software with knowledge of the internal workings of the code being tested.

[View All Answers](#)

**Question - 171:**

What is Load testing?

**Ans:**

1. Load testing is testing an application under heavy loads, such as the testing of a web site under a range of loads to determine at what point the system response time will degrade or fail.

2. Load testing simulates the expected usage of a software program, by simulating multiple users that access the program's services concurrently. Load testing is most useful and most relevant for multi-user systems, client/server models, including web servers. For example, the load placed on the system is increased above normal usage patterns, in order to test the system's response at peak loads

[View All Answers](#)

**Question - 172:**

Explain Inputs?

**Ans:**

Products, services, or information needed from suppliers to make a process work.

[View All Answers](#)

**Question - 173:**

What is Incremental Analysis?

**Ans:**



Incremental analysis occurs when (partial) analysis may be performed on an incomplete product to allow early feedback on the development of that product.

[View All Answers](#)

**Question - 174:**

What is Histogram?

**Ans:**

A graphical description of individual measured values in a data set that is organized according to the frequency or relative frequency of occurrence. A histogram illustrates the shape of the distribution of individual values in a data set along with information regarding the average and variation.

[View All Answers](#)

**Question - 175:**

What is Function Points?

**Ans:**

A consistent measure of software size based on user requirements. Data components include inputs, outputs, etc. Environment characteristics include data communications, performance, reusability, operational ease, etc. Weight scale: 0 = not present; 1 = minor influence, 5 = strong influence.

[View All Answers](#)

**Question - 176:**

What is Fault-based Testing?

**Ans:**

1. Testing that employs a test data selection strategy designed to generate test data capable of demonstrating the absence of a set of pre-specified faults, typically, frequently occurring faults.
2. This type of testing allows for designing test cases based on the client specification or the code or both. It tries to identify plausible faults (areas of design or code that may lead to errors). For each of these faults a test case is developed to "flush" the errors out. These tests also force each line of code to be executed

[View All Answers](#)

**Question - 177:**

What is Fault?

**Ans:**

A manifestation of an error in software. A fault, if encountered, may cause a failure.

[View All Answers](#)

**Question - 178:**

What is Failure?

**Ans:**

The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered.

[View All Answers](#)

**Question - 179:**

Evaluation?

**Ans:**

The process of examining a system or system component to determine the extent to which specified properties are present.

[View All Answers](#)

**Question - 180:**

What is Dynamic Analysis?

**Ans:**

The process of evaluating a program based on execution of that program. Dynamic analysis approaches rely on executing a piece of software with selected test data.

[View All Answers](#)

**Question - 181:**

Explain Error?

**Ans:**

- 1) A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition; and
- 2) a mental mistake made by a programmer that may result in a program fault.

[View All Answers](#)

**Question - 182:**

What is Defect Density?

**Ans:**

Ratio of the number of defects to program length (a relative number).



[View All Answers](#)

**Question - 183:**

Explain Debugging?

**Ans:**

The act of attempting to determine the cause of the symptoms of malfunctions detected by testing or by frenzied user complaints.

[View All Answers](#)

**Question - 184:**

Explain Cyclomatic complexity?

**Ans:**

- (1)The number of independent paths through a program.
- (2)The cyclomatic complexity of a program is equivalent to the number of decision statements plus 1.

[View All Answers](#)

**Question - 185:**

What is Integration?

**Ans:**

The process of combining software components or hardware components, or both, into an overall system.

[View All Answers](#)

**Question - 186:**

Explain Verification?

**Ans:**

The process of evaluating the products of a given software development activity to determine correctness and consistency with respect to the products and standards provided as input to that activity.

[View All Answers](#)

**Question - 187:**

What is Glossary of Software QA/Testing?

**Ans:**

The end user that actually uses the product received.

[View All Answers](#)

**Question - 188:**

Explain Client?

**Ans:**

The end user that pays for the product received, and receives the benefit from the use of the product.

[View All Answers](#)

**Question - 189:**

Explain Cause-effect Graphing?

**Ans:**

A testing technique that aids in selecting, in a systematic way, a high-yield set of test cases that logically relates causes to effects to produce test cases. It has a beneficial side effect in pointing out incompleteness and ambiguities in specifications.

[View All Answers](#)

**Question - 190:**

What is Brainstorming?

**Ans:**

A group process for generating creative and diverse ideas.

[View All Answers](#)

**Question - 191:**

What is Cause-effect Graphing?

**Ans:**

A testing technique that aids in selecting, in a systematic way, a high-yield set of test cases that logically relates causes to effects to produce test cases. It has a beneficial side effect in pointing out incompleteness and ambiguities in specifications.

[View All Answers](#)

**Question - 192:**



What is Measurement?

**Ans:**

The act or process of measuring. A figure, extent, or amount obtained by measuring.

[View All Answers](#)

**Question - 193:**

Explain Code walkthrough?

**Ans:**

A manual testing [error detection] technique where program [source code] logic [structure] is traced manually [mentally] by a group with a small set of test cases, while the state of program variables is manually monitored, to analyze the programmer's logic and assumptions. Contrast with code audit, code inspection, code review.

[View All Answers](#)

**Question - 194:**

Explain Code inspection?

**Ans:**

A manual [formal] testing [error detection] technique where the programmer reads source code, statement by statement, to a group who ask questions analyzing the program logic, analyzing the code with respect to a checklist of historically common programming errors, and analyzing its compliance with coding standards. Contrast with code audit, code review, code walkthrough. This technique can also be applied to other software and configuration items.

[View All Answers](#)

**Question - 195:**

Explain Smoke testing?

**Ans:**

Smoke testing is a relatively simple check to see whether the product "smokes" when it runs. Smoke testing is also known as ad hoc testing, i.e. testing without a formal test plan.

With many projects, smoke testing is carried out in addition to formal testing. If smoke testing is carried out by a skilled tester, it can often find problems that are not caught during regular testing. Sometimes, if testing occurs very early or very late in the software development life cycle, this can be the only kind of testing that can be performed.

Smoke testing, by definition, is not exhaustive, but, over time, you can increase your coverage of smoke testing.

A common practice at Microsoft, and some other software companies, is the daily build and smoke test process. This means, every file is compiled, linked, and combined into an executable file every single day, and then the software is smoke tested.

Smoke testing minimizes integration risk, reduces the risk of low quality, supports easier defect diagnosis, and improves morale. Smoke testing does not have to be exhaustive, but should expose any major problems. Smoke testing should be thorough enough that, if it passes, the tester can assume the product is stable enough to be tested more thoroughly. Without smoke testing, the daily build is just a time wasting exercise. Smoke testing is the sentry that guards against any errors in development and future problems during integration. At first, smoke testing might be the testing of something that is easy to test. Then, as the system grows, smoke testing should expand and grow, from a few seconds to 30 minutes or more.

[View All Answers](#)

**Question - 196:**

Explain Mutation testing?

**Ans:**

Mutation testing is testing where our goal is to make mutant software fail, and thus demonstrate the adequacy of our test case. How do we perform mutation testing?

Step one: We create a set of mutant software. In other words, each mutant software differs from the original software by one mutation, i.e. one single syntax change made to one of its program statements, i.e. each mutant software contains one single fault.

Step two: We write and apply test cases to the original software and to the mutant software.

Step three: We evaluate the results, based on the following set of criteria: Our test case is inadequate, if both the original software and all mutant software generate the same output. Our test case is adequate, if our test case detects faults in our software, or, if, at least, one mutant software generates a different output than does the original software for our test case.

[View All Answers](#)

**Question - 197:**

Explain Monkey testing?

**Ans:**

Monkey testing is random testing performed by automated testing tools (after the latter are developed by humans). These automated testing tools are considered "monkeys", if they work at random. We call them "monkeys" because it is widely believed that if we allow six monkeys to pound on six typewriters at random, for a million years, they will recreate all the works of Isaac Asimov.

There are "smart monkeys" and "dumb monkeys". "Smart monkeys" are valuable for load and stress testing; they will find a significant number of bugs, but are also very expensive to develop. "Dumb monkeys", on the other hand, are inexpensive to develop, are able to do some basic testing, but they will find few bugs. However, the bugs "dumb monkeys" do find will be hangs and crashes, i.e. the bugs you least want to have in your software product. "Monkey testing" can be valuable, but they should not be your only testing.

[View All Answers](#)

**Question - 198:**

What is Error-based Testing?

**Ans:**

Testing where information about programming style, error-prone language constructs, and other programming knowledge is applied to select test data capable of detecting faults, either a specified class of faults or all possible faults.



[View All Answers](#)

**Question - 199:**

What is Boundary value Testing?

**Ans:**

A testing technique using input values at, just below, and just above, the defined limits of an input domain; and with input values causing outputs to be at, just below, and just above, the defined limits of an output domain.

[View All Answers](#)

**Question - 200:**

Explain Testing?

**Ans:**

(1) The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.

(2) The process of analyzing a software item to detect the differences between existing and required conditions, i.e. bugs, and to evaluate the features of the software items.

[View All Answers](#)

**Question - 201:**

Explain Acceptance Testing?

**Ans:**

Testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. Contrast with testing, development; testing, operational.

or

Formal testing conducted to determine whether or not a system satisfies its acceptance criteria-enables an end user to determine whether or not to accept the system.

[View All Answers](#)

**Question - 202:**

What is Recovery/error testing?

**Ans:**

Recovery/error testing is testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

[View All Answers](#)

**Question - 203:**

Explain Installation testing?

**Ans:**

Installation testing is testing full, partial, upgrade, or install/uninstall processes. The installation test for a release is conducted with the objective of demonstrating production readiness.

This test includes the inventory of configuration items, performed by the application's System Administration, the evaluation of data readiness, and dynamic tests focused on basic system functionality. When necessary, a sanity test is performed, following installation testing.

[View All Answers](#)

**Question - 204:**

Explain End-to-end testing?

**Ans:**

Similar to system testing, the \*macro\* end of the test scale is testing a complete application in a situation that mimics real world use, such as interacting with a database, using network communication, or interacting with other hardware, application, or system.

[View All Answers](#)

**Question - 205:**

What is Gamma testing?

**Ans:**

Gamma testing is testing of software that does have all the required features, but did not go through all the in-house quality checks. Cynics tend to refer to software releases as "gamma testing".

[View All Answers](#)

**Question - 206:**

What is OO System Testing?

**Ans:**

OO system testing is really identical to its counterpart in the procedural environment.

[View All Answers](#)

**Question - 207:**



Functional testing?

**Ans:**

Functional testing is black-box type of testing geared to functional requirements of an application. Test engineers \*should\* perform functional testing.

Functional testing is the same as black box testing. Black box testing a type of testing that considers only externally visible behavior. Black box testing considers neither the code itself, nor the "inner workings" of the software.

Function testing is a testing process that is black-box in nature. It is aimed at examining the overall functionality of the product. It usually includes testing of all the interfaces and should therefore involve the clients in the process. Because every aspect of the software system is being tested, the specifications for this test should be very detailed describing who, where, when and how will conduct the tests and what exactly will be tested.

The portion of the testing that will involve the clients is usually conducted as an alpha test where the developers closely monitor how the clients use the system. They take notes on what needs to be improved.

[View All Answers](#)

**Question - 208:**

What is Volume Testing?

**Ans:**

Testing designed to challenge a system's ability to manage the maximum amount of data over a period of time. This type of testing also evaluates a system's ability to handle overload situations in an orderly fashion.

[View All Answers](#)

**Question - 209:**

Explain Regression Testing?

**Ans:**

1. Rerunning test cases which a program has previously executed correctly in order to detect errors spawned by changes or corrections made during software development and maintenance.

2. The objective of regression testing is to ensure the software remains intact. A baseline set of data and scripts is maintained and executed to verify changes introduced during the release have not "undone" any previous code. Expected results from the baseline are compared to results of the software under test. All discrepancies are highlighted and accounted for, before testing proceeds to the next level.

[View All Answers](#)

**Question - 210:**

What is Statement Testing?

**Ans:**

Testing to satisfy the criterion that each statement in a program be executed at least once during program testing.

[View All Answers](#)

**Question - 211:**

What is What are the parameters of performance testing?

**Ans:**

The term "performance testing" is often used synonymously with stress testing, load testing, reliability testing, and volume testing. Performance testing is part of system testing, but it's also a distinct level of testing. Performance testing verifies loads, volumes, and response times, as defined by requirements.

[View All Answers](#)

**Question - 212:**

Explain Path Testing?

**Ans:**

Testing to satisfy coverage criteria that each logical path through the program be tested. Often paths through the program are grouped into a finite set of classes. One path from each class is then tested.

[View All Answers](#)

**Question - 213:**

What is Audit?

**Ans:**

An inspection/assessment activity that verifies compliance with plans, policies, and procedures, and ensures that resources are conserved. Audit is a staff function; it serves as the "eyes and ears" of management.

[View All Answers](#)

**Question - 214:**

What is Operational Testing?

**Ans:**

Testing conducted to evaluate a system or component in its operational environment. Contrast with testing, development; testing, acceptance;

[View All Answers](#)

**Question - 215:**

Explain System Testing?



**Ans:**

1. The process of testing an integrated hardware and software system to verify that the system meets its specified requirements. Such testing may be conducted in both the development environment and the target environment.

or

The process of testing an integrated hardware and software system to verify that the system meets its specified requirements.

2. Final stage of the testing process should be System Testing. This type of test involves examination of the whole computer system. All the software components, all the hardware components and any interfaces.

The whole computer based system is checked not only for validity but also for met objectives.

It should include recovery testing, security testing, stress testing and performance testing.

Recovery testing uses test cases designed to examine how easily and completely the system can recover from a disaster (power shut down, blown circuit, disk crash, interface failure, insufficient memory, etc.). It is desirable to have a system capable of recovering quickly and with minimal human intervention. It should also have a log of activities happening before the crash (these should be part of daily operations) and a log of messages during the failure (if possible) and upon re-start.

Security testing involves testing the system in order to make sure that unauthorized personnel or other systems cannot gain access to the system and information or resources within it. Programs that check for access to the system via passwords are tested along with any organizational security procedures established.

Stress testing encompasses creating unusual loads on the system in attempts to brake it. System is monitored for performance loss and susceptibility to crashing during the load times. If it does crash as a result of high load, it provides for just one more recovery test.

Performance testing involves monitoring and recording the performance levels during regular and low and high stress loads. It tests the amount of resource usage under the just described conditions and serves as basis for making a forecast of additional resources needed (if any) in the future. It is important to note that performance objectives should have been developed during the planning stage and performance testing is to assure that these objectives are being met. However, these tests may be run in initial stages of production to compare the actual usage to the forecasted figures.

3. System testing is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

The purpose of system testing is to validate an application's accuracy and completeness in performing the functions as designed.

System testing simulates real life scenarios that occur in a "simulated real life" test environment and test all functions of the system that are required in real life.

System testing is deemed complete when actual results and expected results are either in line or differences are explainable or acceptable, based on client input.

Upon completion of integration testing, system testing is started. Before system testing, all unit and integration test results are reviewed by Software QA to ensure all problems have been resolved. For a higher level of testing it is important to understand unresolved problems that originate at unit and integration test levels.

[View All Answers](#)

**Question - 216:**

Explain Stress Testing?

**Ans:**

1. Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements.

2. Stress testing is testing that investigates the behavior of software (and hardware) under extraordinary operating conditions. For example, when a web server is stress tested, testing aims to find out how many users can be on-line, at the same time, without crashing the server. Stress testing tests the stability of a given system or entity. It tests something beyond its normal operational capacity, in order to observe any negative results. For example, a web server is stress tested, using scripts, bots, and various denial of service tools.

[View All Answers](#)

**Question - 217:**

What is The objective of regression testing?

**Ans:**

The objective of regression testing is to test that the fixes have not created any other problems elsewhere. The objective is to ensure the software has remained intact. A baseline set of data and scripts are maintained and executed to verify that changes introduced during the release have not "undone" any previous code. Expected results from the baseline are compared to results of the software under test. All discrepancies have to be highlighted and accounted for, before the testing can proceed to the next level.

[View All Answers](#)

**Question - 218:**

Explain Storage Testing?

**Ans:**

This is a determination of whether or not certain processing conditions use more storage [memory] than estimated.

[View All Answers](#)

**Question - 219:**

What is Qualification Testing?

**Ans:**

Formal testing, usually conducted by the developer for the consumer, to demonstrate that the software meets its specified requirements.

[View All Answers](#)

**Question - 220:**

Explain Performance Testing?

**Ans:**

1. Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements.

Although performance testing is described as a part of system testing, it can be regarded as a distinct level of testing. Performance testing verifies loads, volumes and response times, as defined by requirements.

2. Performance testing verifies loads, volumes, and response times, as defined by requirements. Although performance testing is a part of system testing, it can be regarded as a distinct level of testing.

[View All Answers](#)

**Question - 221:**

What is Parallel/audit testing?

**Ans:**

Parallel/audit testing is testing where the user reconciles the output of the new system to the output of the current system to verify the new system performs the operations correctly.

[View All Answers](#)

**Question - 222:**

What is Parallel Testing?

**Ans:**

Testing a new or an altered data processing system with the same source data that is used in another system. The other system is considered as the standard of comparison.

[View All Answers](#)

**Question - 223:**

What is Mutation testing?

**Ans:**

Mutation testing is testing where our goal is to make mutant software fail, and thus demonstrate the adequacy of our test case. How do we perform mutation testing?  
Step one: We create a set of mutant software. In other words, each mutant software differs from the original software by one mutation, i.e. one single syntax change made to one of its program statements, i.e. each mutant software contains one single fault.

Step two: We write and apply test cases to the original software and to the mutant software.

Step three: We evaluate the results, based on the following set of criteria: Our test case is inadequate, if both the original software and all mutant software generate the same output. Our test case is adequate, if our test case detects faults in our software, or, if, at least, one mutant software generates a different output than does the original software for our test case.

[View All Answers](#)

**Question - 224:**

Explain Functional testing?

**Ans:**

Functional testing is black-box type of testing geared to functional requirements of an application. Test engineers \*should\* perform functional testing.

Functional testing is the same as black box testing. Black box testing is a type of testing that considers only externally visible behavior. Black box testing considers neither the code itself, nor the "inner workings" of the software.

Function testing is a testing process that is black-box in nature. It is aimed at examining the overall functionality of the product. It usually includes testing of all the interfaces and should therefore involve the clients in the process. Because every aspect of the software system is being tested, the specifications for this test should be very detailed describing who, where, when and how will conduct the tests and what exactly will be tested.

The portion of the testing that will involve the clients is usually conducted as an alpha test where the developers closely monitor how the clients use the system. They take notes on what needs to be improved.

[View All Answers](#)

**Question - 225:**

Explain Incremental integration testing?

**Ans:**

Incremental integration testing is continuous testing of an application as new functionality is recommended. This may require that various aspects of an application's functionality are independent enough to work separately, before all parts of the program are completed, or that test drivers are developed as needed.

Incremental testing may be performed by programmers, software engineers, or test engineers.

[View All Answers](#)

**Question - 226:**

Explain Integration Testing?

**Ans:**

1. An orderly progression of testing in which software elements, hardware elements, or both are combined and tested, to evaluate their interactions, until the entire system has been integrated.

2. Integration testing is black box testing. The purpose of integration testing is to ensure distinct components of the application still work in accordance to customer requirements. Test cases are developed with the express purpose of exercising the interfaces between the components. This activity is carried out by the test team. Integration testing is considered complete, when actual results and expected results are either in line or differences are explainable/acceptable based on client input.

[View All Answers](#)

**Question - 227:**

Explain Compatibility testing?

**Ans:**

Compatibility testing is testing how well software performs in a particular hardware, software, operating system, or network environment.

[View All Answers](#)

**Question - 228:**

What is Assertion Testing?

**Ans:**

A dynamic analysis technique which inserts assertions about the relationship between program variables into the program code. The truth of the assertions is determined as the program executes.

[View All Answers](#)

**Question - 229:**

Explain Branch Testing?

**Ans:**

Testing technique to satisfy coverage criteria which require that for each decision point, each possible branch [outcome] be executed at least once. Contrast with testing, path; testing, statement.

[View All Answers](#)

**Question - 230:**

What is Branch Coverage Testing?

**Ans:**

A test method satisfying coverage criteria that requires each decision point at each possible branch to be executed at least once.

[View All Answers](#)

**Question - 231:**

Explain Unit testing?

**Ans:**

Unit testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable.

[View All Answers](#)

**Question - 232:**

What is Open box testing?

**Ans:**

Open box testing is same as white box testing. It's a testing approach that examines the application's program structure, and derives test cases from the application's program logic.

[View All Answers](#)

**Question - 233:**

What is Clear box testing?

**Ans:**

Clear box testing is the same as white box testing. It is a testing approach that examines the application's program structure, and derives test cases from the application's program logic.

Another term for white-box testing. Structural testing is sometimes referred to as clear-box testing, since "white boxes" are considered opaque and do not really permit visibility into the code. This is also known as glass-box or open-box testing.

[View All Answers](#)

**Question - 234:**

Explain Black-box Testing?

**Ans:**

1. Functional testing based on requirements with no knowledge of the internal program structure or data. Also known as closed-box testing.
2. Black box testing indicates whether or not a program meets required specifications by spotting faults of omission -- places where the specification is not fulfilled.
3. Black-box testing relies on the specification of the system or the component that is being tested to derive test cases. The system is a black-box whose behavior can only be determined by studying its inputs and the related outputs
4. Black box testing is functional testing, not based on any knowledge of internal software design or code. Black box testing are based on requirements and functionality.

[View All Answers](#)

**Question - 235:**

Explain Software Testing Strategies?

**Ans:**

A strategy for software testing integrates software test case design techniques into a well - planned series of steps that result in the successful construction of software.

Common Characteristics of Software Testing Strategies

- Testing begins at module level and works outward towards the integration of the entire system.
- Different testing techniques are appropriate at different points in time.
- Testing is conducted by the developer of the software and for large projects by an independent test group.
- Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

[View All Answers](#)

**Question - 236:**



What is The Mission of Testing?

**Ans:**

In well-run projects, the mission of the test team is not merely to perform testing, but to help minimise the risk of product failure. Testers look for manifest problems in the product, potential problems, and the absence of problems. They explore, assess, track, and report product quality, so that others in the project can make informed decisions about product development. It's important to recognise that testers are not out to "break the code." We are not out to embarrass or complain, just to inform. We are human meters of product quality.

[View All Answers](#)

**Question - 237:**

Explain Glass box testing?

**Ans:**

Glass box testing is the same as white box testing. It's a testing approach that examines the application's program structure, and derives test cases from the application's program logic.

[View All Answers](#)

**Question - 238:**

Explain Dynamic Testing?

**Ans:**

Verification or validation performed which executes the system's code.

[View All Answers](#)

**Question - 239:**

Explain Beta Testing?

**Ans:**

(1) Acceptance testing performed by the customer in a live application of the software, at one or more end user sites, in an environment not controlled by the developer.

(2) For medical device software such use may require an Investigational Device Exemption [IDE] or Institutional Review Board [IRB] approval.

Testing conducted at one or more end user sites by the end user of a delivered software product or system.

Beta testing is testing an application when development and testing are essentially completed and final bugs and problems need to be found before the final release.

Beta testing is typically performed by end-users or others, not programmers, software engineers, or test engineers.

Following alpha testing, "beta versions" of the software are released to a group of people, and limited public tests are performed, so that further testing can ensure the product has few bugs. Other times, beta versions are made available to the general public, in order to receive as much feedback as possible. The goal is to benefit the maximum number of future users.

[View All Answers](#)

**Question - 240:**

Explain Exhaustive Testing?

**Ans:**

Executing the program with all possible combinations of values for program variables. Feasible only for small, simple programs.

[View All Answers](#)

**Question - 241:**

What is Interface Testing?

**Ans:**

Testing conducted to evaluate whether systems or components pass data and control correctly to one another. Contrast with testing, unit; testing, system.

[View All Answers](#)

**Question - 242:**

What is OO Integration Testing?

**Ans:**

This strategy involves testing the classes as they are integrated into the system. The traditional approach would test each operation separately as they are implemented into a class. In OO system this approach is not viable because of the "direct and indirect interactions of the components that make up the class"

. Integration testing in OO can be performed in two basic ways :

- Thread-based - Takes all the classes needed to react to a given input. Each class is unit tested and then thread constructed from these classes tested as a set.

- Uses-based - Tests classes in groups. Once the group is tested, the next group that uses the first group (dependent classes) is tested. Then the group that uses the second group and so on. Use of stubs or drivers may be necessary. Cluster testing is similar to testing builds in the traditional model. Basically collaborating classes are tested in clusters.

[View All Answers](#)

**Question - 243:**

What is Incremental testing?

**Ans:**

Incremental testing is partial testing of an incomplete product. The goal of incremental testing is to provide an early feedback to software developers.

[View All Answers](#)

**Question - 244:**

What is Alpha Testing?

**Ans:**

1. Acceptance testing performed by the customer in a controlled environment at the developer's site. The software is used by the customer in a setting approximating the target environment with the developer observing and recording errors and usage problems. Testing of a software product or system conducted at the developer's site by the end user.
2. Alpha testing is testing of an application when development is nearing completion. Minor design changes can still be made as a result of alpha testing. Alpha testing is typically performed by a group that is independent of the design team, but still within the company, e.g. in-house software test engineers, or software QA engineers.
3. Alpha testing is final testing before the software is released to the general public. First, (and this is called the first phase of alpha testing), the software is tested by in-house developers. They use either debugger software, or hardware-assisted debuggers. The goal is to catch bugs quickly. Then, (and this is called second stage of alpha testing), the software is handed over to us, the software QA staff, for additional testing in an environment that is similar to the intended use.

[View All Answers](#)

**Question - 245:**

What is Manual Testing?

**Ans:**

That part of software testing that requires operator input, analysis, or evaluation.

or

A manual test is a test for which there is no automation. Instead, test steps are outlined in a document for the tester to complete. The tester can then report test results and submit bugs as appropriate.

[View All Answers](#)

**Question - 246:**

What is Grey box testing?

**Ans:**

Grey box testing is a software testing technique that uses a combination of black box testing and white box testing. Gray box testing is not black box testing, because the tester does know some of the internal workings of the software under test. In grey box testing, the tester applies a limited number of test cases to the internal workings of the software under test. In the remaining part of the grey box testing, one takes a black box approach in applying inputs to the software under test and observing the outputs.

[View All Answers](#)

**Question - 247:**

What is Test schedule?

**Ans:**

The test schedule is a schedule that identifies all tasks required for a successful testing effort, a schedule of all test activities and resource requirements.

[View All Answers](#)

**Question - 248:**

What is Test methodology?

**Ans:**

Test methodology is up to the end client, and can be used, reused, and molded to your end client's needs. Rob Davis believes that using the right test methodology is important in the development and ongoing maintenance of his clients' applications.

[View All Answers](#)

**Question - 249:**

Explain Quality Control?

**Ans:**

The operational techniques and procedures used to achieve quality requirements.

[View All Answers](#)

**Question - 250:**

What is Quality Assurance (QA)?

**Ans:**

- (1) The planned systematic activities necessary to ensure that a component, module, or system conforms to established technical requirements.
- (2) All actions that are taken to ensure that a development organization delivers products that meet performance requirements and adhere to standards and procedures.
- (3) The policy, procedures, and systematic actions established in an enterprise for the purpose of providing and maintaining some degree of confidence in data integrity and accuracy throughout the life cycle of the data, which includes input, update, manipulation, and output.
- (4) (QA) The actions, planned and performed, to provide confidence that all systems and components that influence the quality of the product are working as expected individually and collectively.

[View All Answers](#)

**Question - 251:**

What is Software Quality Assurance?

**Ans:**



Software QA involves the entire software development PROCESS - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to 'prevention'.

[View All Answers](#)

Interview Questions Answers.ORG

## Testing Most Popular & Related Interview Guides

- 1 : [Manual Testing Interview Questions and Answers.](#)
- 2 : [Rational TestSuite Interview Questions and Answers.](#)
- 3 : [QTP Interview Questions and Answers.](#)
- 4 : [Database Testing Interview Questions and Answers.](#)
- 5 : [JUnit Interview Questions and Answers.](#)
- 6 : [Test Cases Interview Questions and Answers.](#)
- 7 : [Software Testing Interview Questions and Answers.](#)
- 8 : [Mobile Testing Interview Questions and Answers.](#)
- 9 : [QA Testing Interview Questions and Answers.](#)
- 10 : [WinRunner Interview Questions and Answers.](#)

Follow us on FaceBook

[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/  
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)