



SOFTWARE QUALITY ASSURANCE

Complete Interview Preparation Document for Manual & Automation Engineer



PREPARED BY

UMAR ARSHAD | Software Quality Assurance Engineer | Manual - Automation

I originally created this Document for all Fresh Graduates and Experience Professional candidates for Interview Preparation.

* But I have decided to open-source it and make it available for everyone who wants to learn SQA Fundamental, Database Concepts, SCRUM Fundamentals, OOP Basics, Automation concepts like selenium, JMeter, and API.

I added everything that you will need as an SQA Engineer Manual or Automation. The ideal use case of this document is that you print it and keep it next to you while you are learning and practicing on your computer.

Please approach me if you see any mistake or feel any wrong concept.

If anyone wants to get Training on Manual or Automation tools just ping me on my contact.

Enjoy! Cheers,

MUHAMMAD UMAR ARSHAD

Software Quality Assurance Engineer
Consultant Automation Engineer
Systems Ltd Pakistan | Visionet USA



CONTACT



umar.it.376@gmail.com



+92-307-2980604



umar.it.376@outlook



www.linkedin.com/in/muhammad-umar-arshad-005ba3139

Table of Contents

Chapter 1 Software Development Life Cycle (SDLC)	3
Chapter 2 STLC & Principle of Software Testing.....	7
Chapter 3 Testing Documents.....	10
Chapter 4 Methods & Types of Software Testing	18
Chapter 5 Techniques of Software Testing.....	30
Chapter 6 Some Common Concepts	33
Chapter 7 Road Map for Mobile Application Testing	36
Chapter 8 Agile Methodology and Concepts.....	38
Chapter 9 Database & SQL Concepts	44
Chapter 10 Common SQL Queries	60
Chapter 11 Application Program Interface (API).....	61
Chapter 12 Object Oriented Programming Basics (OOP).....	71
Chapter 13 Selenium WebDriver (Automation)	73
Chapter 14 JMeter Concepts (Automation)	86
Chapter 15 Aptitude Questions	93

Chapter 1 Software Development Life Cycle (SDLC)

1 What is SDLC? Imp

The Software Development Lifecycle is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software which meets customer expectations. The software development should be complete in the pre-defined time frame and cost.

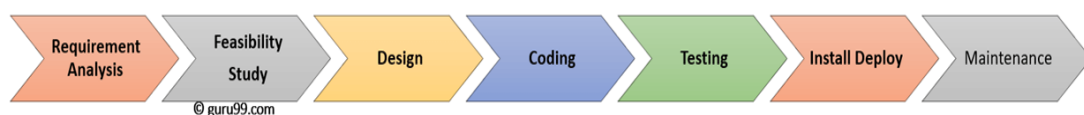
2 Why SDLC?

Here, are prime reasons why SDLC is important for developing a software system.

- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control
- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead

3 SDLC Phases? imp

The entire SDLC process divided into the following stages:



- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study:
- Phase 3: Design:
- Phase 4: Coding:
- Phase 5: Testing:
- Phase 6: Installation/Deployment:
- Phase 7: Maintenance:

3.1 Requirements Gathering/Analysis.

This is a process with much communication taking place between stakeholders, end users and the project team. Meetings with managers, stake holders and users are held in order to determine the requirements like; who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirement gathering phase. **The QA engineer playing the role to configure the requirements using requirements traceability matrix (RTM).**

3.2 Design:

In this phase the **software design** is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying **hardware and system requirements** and also helps in defining overall **system architecture**.

In this phase **the QA Engineers** comes up with **the Test strategy**, where they mention what to test, how to test.

3.3 Implementation / Coding:

Upon receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. **This is the longest phase of SDLC.** In this phase the **QA Engineers** comes up with the **Test Environment setup and test Case Documentation**.

3.4 Testing:

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase all types of like **unit testing, integration testing**, Smoke Testing, **functional testing, Sanity Testing, system testing, acceptance testing** is done as well as **non-functional testing** are also done.

3.5 Deployment: After successful testing the product is delivered / deployed to the customer for their use.

As soon as the product is given to the customers, they will first do the **beta testing/User Acceptance Testing**. If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the **bugs** are fixed then the final deployment will happen.

Popular SDLC models

Here, are some most important phases of SDLC life cycle:

4 Waterfall model

Waterfall model works well for smaller projects where requirements are very well understood.

The waterfall is a widely accepted SDLC model. In this approach, the whole process of the software development is divided into various phases. In this SDLC model, the outcome of one phase acts as the input for the next phase.

This SDLC model is documentation-intensive, with earlier phases documenting what need be performed in the subsequent phases.

5 Incremental Model

The incremental model is not a separate model. It is essentially a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met. In this method, every cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

6 V-Model

In this type of SDLC model testing and the development, the phase is planned in parallel. So, there are verification phases on the side and the validation phase on the other side. V-Model joins by Coding phase.

7 Agile Model - imp

Agile methodology is a practice which promotes continue interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks. **In 'Agile Model' after every sprint there is a demo-able feature to the customer.** Hence customer can see the features whether they are satisfying

their need or not. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications.

8 Spiral Model

The spiral model is a risk-driven process model. This SDLC model helps the team to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc.

This model adopts the best features of the prototyping model and the waterfall model. The spiral methodology is a combination of rapid prototyping and concurrency in design and development activities.

9 Prototyping Model

Prototyping model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved. It also creates base to produce the final system or software. It works best in scenarios where the project's requirements are not known in detail. It is an iterative, trial and error method which takes place between developer and client.

Chapter 2 STLC & Principle of Software Testing

1 What is Software Testing? imp

According to ANSI/IEEE 1059 standard – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.

2 What are the principles of software testing?

2.1 Testing shows presence of defects

Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

2.2 Exhaustive testing is impossible

If you were testing this Operating system, you would realize that defects are likely to be found in multi-tasking activity and need to be tested thoroughly which brings us to our next principal Defect Clustering.

2.3 Early testing

Early Testing - Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages.

2.4 Defect clustering

Defect Clustering which states that a small number of modules contain most of the defects detected. If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

2.5 Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide. Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects. To overcome this, the test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects.

2.6 Testing is context depending

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed software's are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type.

2.7 Absence of error - Fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user's needs & requirements.

3 What is STLC? imp

It is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product.

Following steps are involved in Software Testing Life Cycle (STLC).

1. Requirement Analysis (RTM)
2. Test Planning (Test Strategy, Test Plan, Test Bed Creation)
3. Test Case Development (Test Procedures, Test Scenarios, Test Cases)
4. Environment Setup
5. Test Execution
6. Defect Reporting

4 What are Quality Assurance and Quality Control? imp

Quality Assurance:

Quality Assurance involves in **process-oriented activities**. It ensures the prevention of defects in the process used to make Software Applications. So the defects don't arise when the Software Application is being developed. The process is:

- **Plan** - Organization should plan and establish the process related objectives and determine the processes that are required to deliver a high-Quality end product.
- **Do** - Development and testing of Processes and also "do" changes in the processes
- **Check** - Monitoring of processes, modify the processes, and check whether it meets the predetermined objectives

- **Act** - A Quality Assurance tester should implement actions that are necessary to achieve improvements in the processes.

Quality Control:

Quality Control involves in product-oriented activities. It executes the program or code to identify the defects in the Software Application.

5 What are the Quality assurance and Quality Control standards?

Quality assurance system standards, including ISO 9001, are defined as frameworks that provide regulations to organizations to ensure that their processes, inputs, products, and services are capable of meeting every customer requirement.

- Ensuring maximum satisfaction of clients by meeting their quality requirements
- Safety of products and services during usage
- Complying with international regulations and local legislative rules
- Being environmentally responsible
- Confidentiality of stakeholders including customers, employees, partners, and investors
- Assuring a safer workplace for employees
- Optimum allocation of resources and minimization of waste

6 What is Verification & Validation in software testing? imp

- **Validation:** Are we building the right system?
- **Verification:** Are we building the system, right?

In other words, validation is concerned with checking that the system will meet the customer's actual needs. validation is an extremely subjective process. Validation includes activities such as requirements modelling, prototyping and user evaluation.

While verification is concerned with whether the system is well-engineered, error-free, and so on. Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful. Verification includes all the activities associated with the producing high-quality software: testing, inspection, design analysis, specification analysis, and so on. It is a relatively objective process.

Chapter 3 Testing Documents

1. What is Test Plan Document?

Test plan document is a document which contains the plan for all the testing activities to be done to deliver a quality product. Test Plan document is derived from the Product Description, SRS, or Use Case documents for all future activities of the project. It is usually prepared by the Test Lead or Test Manager.

1. Test plan identifier
2. References
3. Introduction
4. Test items (functions)
5. Software risk issues
6. Features to be tested
7. Features not to be tested
8. Approach
9. Items pass/fail criteria
10. Suspension criteria and resolution requirements
11. Test deliverables
12. Remaining test tasks
13. Environmental needs
14. Staff and training needs
15. Responsibility
16. Schedule
17. Plan risks and contingencies
18. Approvals

2. What is Test Strategy? imp

Test Strategy is a high-level document (static document) and usually developed by the project manager. It is a document that captures the approach on how we go about testing the product achieving the goals. It is normally derived from the Business Requirement Specification (BRS). Documents like Test strategy doc is project-based document it can change according to project domain and requirements.

3. What is Test Suite? imp

Test Suite is a collection of test cases. The test cases which are intended to test an application.

4. What is Test Scenario? imp

Test Scenario gives the idea of what we have to test. Test Scenario is like a high-level test case. A test scenario is more agile and focuses on the end-to-end functionality of the software.

5. What is Test Case? imp

Test cases are the set of positive and negative executable steps of a test scenario, A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly.

Components of Test case:

- Test suite ID (Mandatory)
- Test case ID (Mandatory)
- Summary / Objective (Mandatory)
- Steps to be executed (Mandatory)
- Test Data (Mandatory)
- Pre-Conditions (Non-Mandatory)
- Post-Conditions (Non-Mandatory)
- Expected Result (Mandatory)
- Actual Result (Mandatory)
- Comments/Remarks (Non-Mandatory)
- Status – pass/fail (Mandatory)

6. What is Test Bed? imp

An environment configured for testing. Test bed consists of hardware, software, network configuration, an application under test, other related software.

7. What is Test Environment? imp

Test Environment is the combination of hardware and software on which Test Team performs testing. Example:

- Application Type: Web Application
- OS: Windows
- Web Server: IIS
- Web Page Design: Dot Net
- Client-Side Validation: JavaScript
- Server-Side Scripting: ASP Dot Net

Prepared by: **Mr. Umar Arshad** | Software Quality Assurance Engineer

- Database: MS SQL Server
- Browser: IE/Firefox/Chrome

8. What is Test Data?

Test data is the data that is used by the testers to run the test cases. Whilst running the test cases, testers need to enter some input data. To do so, testers prepare test data. It can be prepared manually and also by using tools.

9. What is Test Harness? imp

Test Harness in Software Testing is a collection of stubs, drivers and other supporting tools required to automate test execution. Test harness executes tests by using a test library and generates test reports. Test harness contains all the information needed to compile and run a test like test cases, target deployment port (TDP), source file under test, stubs, etc.

10. What is Test Closure?

Test Closure is the note prepared before test team formally completes the testing process. This note contains the total no. of test cases, total no. of test cases executed, total no. of defects found, total no. of defects fixed, total no. of bugs not fixed, total no of bugs rejected etc.,

11. What is Risk Factor and its Types?

In software testing Risks are the possible problems that might endanger the objectives of the project stakeholders. It is the possibility of a negative or undesirable outcome. A risk is something that has not happened yet and it may never happen; it is a potential problem.

The types of Risk in a Test Project can be broadly categorized as

1. Strategy Risk: This includes Budget, Communication and Management risks
2. Project Definition Risks: This includes Project target, Scope, and requirements risks.
3. Human Resources Risk: This includes Skill, Team members and organization risks.

12. What are the tasks of Test Closure activities in Software Testing?

Test Closure activities fall into four major groups.

Test Completion Check: To ensure all tests should be either run or deliberately skipped and all known defects should be either fixed, deferred for a future release or accepted as a permanent restriction.

Test Artifacts handover: Tests and test environments should be handed over to those responsible for maintenance testing. Known defects accepted or deferred should be documented and communicated to those who will use and support the use of the system.

Lessons learned: Analyzing lessons learned to determine changes needed for future releases and projects. In retrospective meetings, plans are established to ensure that good practices can be repeated and poor practices are not repeated.

Result: Archiving results, logs, reports, and other documents and work products in the CMS (configuration management system).

13. List out Test Deliverables?

1. Test Strategy
2. Test Plan
3. Effort Estimation Report
4. Test Scenarios
5. Test Cases/Scripts
6. Test Data
7. Requirement Traceability Matrix (RTM)
8. Defect Report/Bug Report
9. Test Execution Report
10. Graphs and Metrics
11. Test summary report
12. Test incident report
13. Test closure report
14. Release Note
15. Installation/configuration guide
16. User guide
17. Test status report
18. Weekly status report (Project manager to client)

14. What is RTM? imp

Requirements Traceability Matrix (RTM) is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled. Requirement Traceability Matrix AKA Traceability Matrix or Cross Reference Matrix.

Which Parameters to include in Requirement Traceability Matrix?

- Requirement ID
- Requirement Type and Description
- Test Cases with Status

Req No	Req Desc	Testcase ID	Status
123	Login to the application	TC01,TC02,TC03	TC01-Pass TC02-Pass
345	Ticket Creation	TC04,TC05,TC06, TC07,TC08,TC09 TC010	TC04-Pass TC05-Pass TC06-Pass TC06-Fail TC07-No Run
456	Search Ticket	TC011,TC012, TC013,TC014	TC011-Pass TC012-Fail TC013-Pass TC014-No Run

Types of Traceability Test Matrix

In Software Engineering, traceability matrix can be divided into three major components as mentioned below:

- **Forward traceability:** This matrix is used to check whether the project progresses in the forward desired direction and for the right product. It maps requirements to test cases.
- **Backward or reverse traceability:** It is used to ensure whether the current product remains on the right track. It maps test cases to requirements.
- **Bi-directional traceability (Forward + Backward):** This traceability matrix ensures that all requirements are covered by test cases. It analyzes the impact of a change in requirements affected by the Defect in a work product and vice versa.

How to create Requirement Traceability Matrix?

Prepared by: **Mr. Umar Arshad** | Software Quality Assurance Engineer

On the basis of the **Business Requirement Document (BRD)** and **Technical Requirement Document (TRD)**, testers start writing test cases.

Step 1: Our sample **Test Case** is

"Verify Login, when correct ID and Password is entered, it should log in successfully"

TestCase #	Test Case	Test Steps	Test Data	Expected Result
1	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

When correct password and id entered, it should login successfully

Step 2: Identify the **Technical Requirement** that this test case is verifying. For our test case, the technical requirement is T94 is being verified.

Login

T92 User-ID must not be blank

T93 Password must not be blank

T94 If userid and password are valid. Login

Here is our TRD
(Technical
Requirement
Document)

T94 If userid and password are valid. Login

T94 is our technical
requirement that verifies
successful login

Step 3: Note this Technical Requirement (T94) in the Test Case.

TestCase #	TR #	Test Steps	Test Data	Expected
1	T94	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Note the Technical
Requirement in the test case

Step 4: Identify the Business Requirement for which this TR (Technical Requirement-T94) is defined

BR#	Module Name	Applicable Roles	Description
B1	Login and Logout	Manager Customer	Customer: A customer can login using the login page Manager: A manager can login using the login page of customer. Post Login homepage will show different links based on role

Identify the Business Requirement for which T94 is defined

Step 5: Note the BR (Business Requirement) in Test Case

Test Case #	BR #	TR #	Test Case	Test Steps	Test Data	Expected Result
1	B1	T94	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Step 6: Do above for all Test Cases. Later Extract the First 3 Columns from your Test Suite. RTM in testing is Ready!

Business Requirement #	Technical Requirement #	Test Case ID
B1	T94	1
B2	T95	3
B3	T96	3
B4	T97	4

Requirement Traceability Matrix

Advantage of Requirement Traceability Matrix

- It confirms 100% test coverage
- It highlights any requirements missing or document inconsistencies
- It shows the overall defects or execution status with a focus on business requirements
- It helps in analyzing or estimating the impact on the QA team's work with respect to revisiting or re-working on the test cases.

15. What is Test Coverage? imp

Test Coverage states which requirements of the customers are to be verified when the testing phase starts. Test Coverage is a term that determines whether the test cases are

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

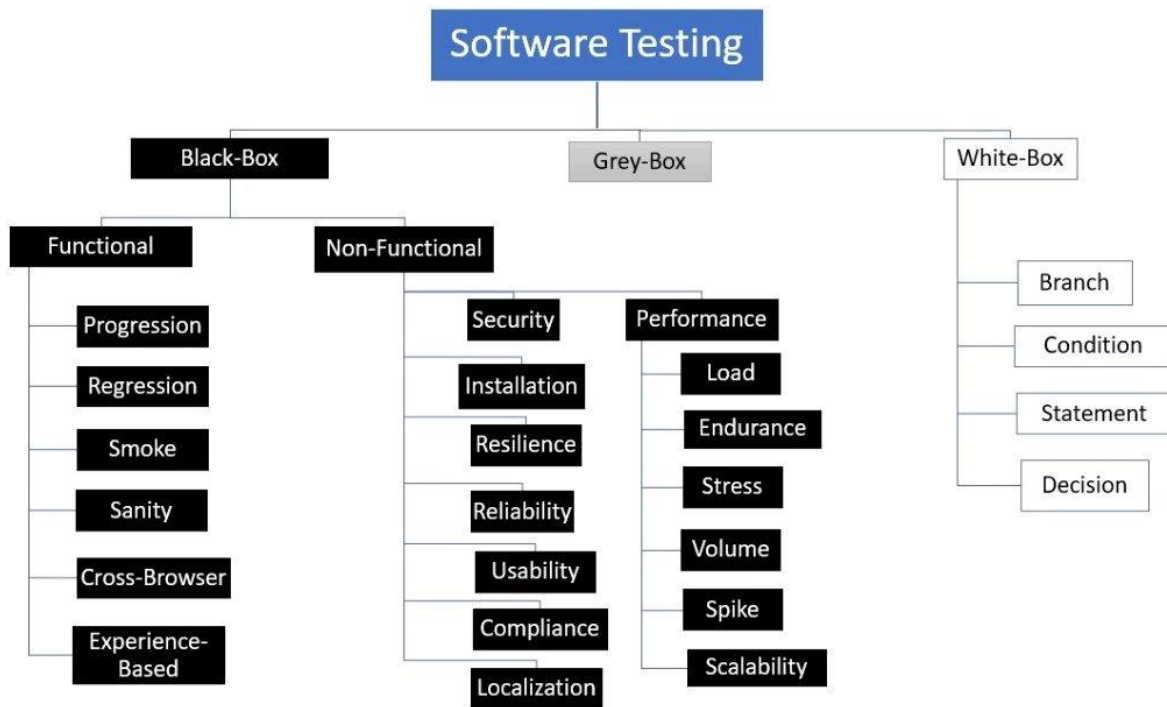
written and executed to ensure to test the software application completely, in such a way that minimal or NIL defects are reported.

How to achieve Test Coverage?

The maximum Test Coverage can be achieved by establishing good 'Requirement Traceability'.

- Mapping all internal defects to the test cases designed
- Mapping all the Customer Reported Defects (CRD) to individual test cases for the future regression test suite.

Chapter 4 Methods & Types of Software Testing



There are 3 methods of software testing.

1) White box

2) Black box

3) Grey Box

1. What is White Box Testing? imp

White Box Testing is also called as Glass Box, Clear Box, and Structural Testing. It is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually was done at the unit level.

2. What is Black Box Testing? imp

Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.

3. What is Grey Box Testing?

Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

There are 3 Types of Software testing (Alpha, Beta, Gamma)

4. What is Alpha Testing? imp

Alpha testing is done by the in-house developers (who developed the software) and testers. Sometimes alpha testing is done by the client or outsourcing team with the presence of developers or testers.

It has two phases:

- **In the first phase** of alpha testing, the software is tested by in-house developers. They use debugger software. The goal is to catch bugs quickly.
- **In the second phase** of alpha testing, the software is handed over to the software QA staff, for additional testing in an environment that is similar to the intended use.

5. What is Beta Testing? imp

Beta testing is done by a limited number of end users before delivery. Usually, it is done in the client place.

6. What is Gamma Testing?

Gamma testing is done when the software is ready for release with specified requirements. It is done at the client place. It is done directly by skipping all the in-house testing activities.

7. What is Functional Testing? imp

In simple words, what the system actually does is functional testing. To verify that each function of the software application behaves as specified in the requirement document. Testing all the functionalities by providing appropriate input to verify whether the actual output is matching the expected output or not. It falls within the scope of black box testing and the testers need not concern about the source code of the application.

7.1 What is Unit/Module Testing? imp

Unit Testing is also called as Module Testing or Component Testing. It is done to check whether the individual unit or module of the source code is working properly. It is done by the developers in the developer's environment.

7.2 What is Integration Testing? imp

Integration Testing is the process of testing the interface between the two software units. Integration testing is done by three ways. Big Bang Approach, Top-Down Approach, Bottom-Up Approach.

7.3 What is System Testing? imp

Testing the fully integrated application to evaluate the system's compliance with its specified requirements is called System Testing End to End testing. Verifying the completed system to ensure that the application works as intended or not.

System testing is carried out by specialist testers or independent testers.

System testing should investigate both functional and non-functional requirements of the testing.

7.4 What is Smoke Testing? imp

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as "Day 0" check. It is done at the "build level". It helps not to waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet.

7.5 What is Sanity Testing? imp

Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper. It is also called as a subset of Regression testing. It is done at the "release level". **We perform sanity testing when we don't have enough time for regression testing.**

7.6 What is Regression Testing? imp

Repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software components.

Usually, we do regression testing in the following cases:

1. New functionalities are added to the application
2. Change Requirement (In organizations, we call it as CR)
3. Defect Fixing
4. Performance Issue Fix
5. Environment change (E.g., Updating the DB from MySQL to Oracle)

7.7 What is Retesting Testing? imp

Retesting is done to make sure that the tests cases which failed in last execution are passed after the defects are fixed. Retesting is carried out based on the defect fixes. In Retesting, the cases which are failed earlier can be included to check if the functionality failure in an earlier build.

7.8 What is Exploratory Testing? Imp

Usually, this process will be carried out by domain experts. They perform testing just by exploring the functionalities of the application without having the knowledge of the requirements.

8. What is Monkey Testing?

Perform abnormal action on the application deliberately in order to verify the stability of the application.

9. What is Big Bang Approach?

Combining all the modules once and verifying the functionality after completion of individual module testing.

Top down and bottom up are carried out by using dummy modules known as Stubs and Drivers. These Stubs and Drivers are used to stand-in for missing components to simulate data communication between modules.

10. What is Top-Down Approach?

Testing takes place from top to bottom. High-level modules are tested first and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended. Stubs are used as a temporary module if a module is not ready for integration testing.

11. What is Bottom-Up Approach?

It is a reciprocal of the Top-Down Approach. Testing takes place from bottom to up. Lowest level modules are tested first and then high-level modules and finally integrating the high-level modules to a low level to ensure the system is working as intended. Drivers are used as a temporary module for integration testing.

12. What is User Acceptance Testing / UAT? imp

It is also known as pre-production testing. This is done by the end users along with the testers to validate the functionality of the application. After successful acceptance testing, formal testing is conducted to determine whether an application is developed as per the requirement. It allows the customer to accept or reject the application. Types of acceptance testing are Alpha, Beta & Gamma.

13. What is Positive and Negative Testing?

Positive Testing: It is to determine what system is supposed to do. It helps to check whether the application is justifying the requirements or not.

Negative Testing: It is to determine what system is not supposed to do. It helps to find the defects from the software.

14. What is Non-Functional Testing? imp

In simple words, how well the system performs is non-functionality testing. Non-functional testing refers to various aspects of the software such as performance, load, stress, scalability, security, compatibility etc., Main focus is to improve the user experience on how fast the system responds to a request.

15.1. What is Performance Testing? imp

This type of testing determines or validates the speed, scalability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, throughput, and resource-utilization levels that meet the performance objectives for the project or product.

15.2. What is Load Testing? imp

It is to verify that the system/application can handle the expected number of transactions and to verify the system/application behavior under both normal and peak load conditions.

15.3. What is Volume Testing? imp

It is to verify that the system/application can handle a large amount of data.

15.4. What is Stress Testing? imp

It is to verify the behavior of the system once the load increases more than its design expectations.

15.5. What is Scalability Testing?

Scalability testing is a type of non-functional testing. It is to determine how the application under test scales with increasing workload.

15.6. What is Concurrency Testing?

Concurrency testing means accessing the application at the same time by multiple users to ensure the stability of the system. This is mainly used to identify deadlock issues.

15.7. What is GUI Testing? imp

Graphical User Interface Testing is to test the interface between the application and the end user.

15.8. What is Recovery Testing?

Recovery testing is performed in order to determine how quickly the system can recover after the system crash or hardware failure. It comes under the type of non-functional testing.

15.9. What is Installation Testing?

It is to check whether the application is successfully installed and it is working as expected after installation.

15.10. What is Formal Testing? Imp

It is a process where the testers test the application by having pre-planned procedures and proper documentation.

15.11. What is Risk Based Testing? imp

Identify the modules or functionalities which are most likely cause failures and then testing those functionalities.

15.12. What is Compatibility Testing?

It is to deploy and check whether the application is working as expected in a different combination of environmental components.

15.13. What is Usability Testing?

To verify whether the application is user-friendly or not and was comfortably used by an end user or not. The main focus in this testing is to check whether the end user can understand and operate the application easily or not. An application should be self-exploratory and must not require training to operate it.

15.14. What is Security Testing? imp

Security testing is a process to determine whether the system protects data and maintains functionality as intended.

16. What is Soak Testing? imp

Running a system at high load for a prolonged period of time to identify the performance problems is called Soak or Endurance Testing.

17. What is Fuzz Testing? Imp

Fuzz testing is used to identify coding errors and security loopholes in an application. By inputting massive amount of random data to the system in an attempt to make it crash to identify if anything breaks in the application.

18. What is Adhoc Testing? imp

Ad-hoc testing is quite opposite to the formal testing. It is an informal testing type. In Adhoc testing, testers randomly test the application without following any documents and test design techniques. This testing is primarily performed if the knowledge of testers in the application under test is very high. Testers randomly test the application without any test cases or any business requirement document.

19. What is Interface Testing?

Interface testing is performed to evaluate whether two intended modules pass data and communicate correctly to one another.

19.1. What is Reliability Testing?

Perform testing on the application continuously for long period of time in order to verify the stability of the application.

20. What is Bucket Testing? imp

Bucket or Split testing is a method to compare two versions of an application against each other to determine which one performs better.

21. What is Defect Cascading in Software Testing?

Defect cascading in Software testing means triggering of other defects in an application. When a defect is not identified or goes unnoticed while testing, it invokes other defects. It leads to multiple defects in the later stages and results in an increase in a number of defects in the application.

For example, if there is a defect in an accounting system related to negative taxation then the negative taxation defect affects the ledger which in turn affects other reports such as Balance Sheet, Profit & Loss etc.,

22. What is Walk Through? imp

A walkthrough is an informal meeting conducted to learn, gain understanding, and find defects. The author leads the meeting and clarifies the queries raised by the peers in the meeting.

23. What is Inspection? imp

Inspection is a formal meeting lead by a trained moderator, certainly not by the author. The document under inspection is prepared and checked thoroughly by the reviewers before the meeting. In the inspection meeting, the defects found are logged and shared with the author for appropriate actions. Post inspection, a formal follow-up process is used to ensure a timely and corrective action.

24. Who are all involved in an inspection meeting?

Author, Moderator, Reviewer(s), Scribe/Recorder and Manager.

25. What is a Defect? imp

The variation between the actual results and expected results is known as a defect. If a developer finds an issue and corrects it by himself in the development phase, then it's called a defect.

26. What is a Bug? imp

If testers find any mismatch in the application/system in testing phase, then they call it as Bug.

27. What is an Error? imp

We can't compile or run a program due to a coding mistake in a program. If a developer unable to successfully compile or run a program, then they call it as an error.

28. What is a Failure? imp

Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue, then that particular issue is called as a failure.

29. What is Bug Severity? imp

Bug/Defect severity can be defined as the impact of the bug on customer's business. It can be Critical, Major or Minor. In simple words, how much effect will be there on the system because of a particular defect.

30. What is Bug Priority? imp

Defect priority can be defined as how soon the defect should be fixed. It gives the order in which a defect should be resolved. Developers decide which defect they should take

up next based on the priority. It can be High, Medium or Low. Most of the times the priority status is set based on the customer requirement.

Tell some examples of Bug Severity and Bug Priority? Imp

High Priority & High Severity: Submit button is not working on a login page and customers are unable to login to the application

Low Priority & High Severity: key feature failed but there's no impact on customer business, e.g., calculation fault in yearly report which end user won't use on daily basis.

High Priority & Low Severity: Spelling mistake of a company name on the homepage

Low Priority & Low Severity: FAQ page takes a long time to load

31. What is a Critical Bug?

A critical bug is a show stopper which means a large piece of functionality or major system component is completely broken and there is no workaround to move further. For example, Due to a bug in one module, we cannot test the other modules because that blocker bug has blocked other modules. Bugs which affect the customers' business are considered as critical.

Example:

1. "Sign In" button is not working on Gmail App and Gmail users are blocked to login to their accounts.
2. An error message pops up when a customer clicks on transfer money button in a Banking website.

32. What are entry criteria?

Entry criteria is a set of conditions that permits a task to perform, or in absence of any of these conditions, the task cannot be performed.

- The requirement document should be available.
- Complete understanding of the application flow is required.
- The Test Plan Document should be ready.

33. What is exit criteria?

Exit criteria is a set of expectations; this should be met before concluding the STLC phase.

- Test Cases should be written and reviewed.
- Test Data should be identified and ready.
- Test automation script should be ready if applicable.

34. What is the Test Management Reviews & Audit?

- **Management Review:**

Management Review is also known as Software Quality Assurance or (SQA). It focuses more on the software process rather than the software work products. Quality Assurance is a set of activities designed to ensure that the project manager follows the standard process which is already pre-defined. In other words, Quality Assurance makes sure the Test Manager is doing the right things in the right way.

- **Audit:**

An audit is the examination of the work products and related information to assesses whether the standard process was followed or not.

Chapter 5 Techniques of Software Testing

5.1. What is Boundary Value Analysis? imp

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition.

Example:

Input condition is valid between 1 to 10

Boundary values 0,1,2 and 9,10,11

5.2. What is Decision Table testing? imp

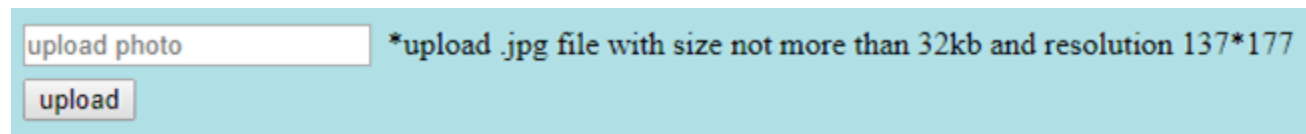
Decision Table is a Cause-Effect Table. This test technique is appropriate for functionalities which has logical relationships between inputs (if-else logic). In Decision table technique, we deal with combinations of inputs. To identify the test cases with decision table, we consider conditions and actions. We take conditions as inputs and actions as outputs.

How to make Decision Table for Upload Screen?

Now consider a dialogue box which will ask the user to upload photo with certain conditions like –

1. You can upload only '.jpg' format image
2. file size less than 32kb
3. resolution 137*177.

If any of the conditions fails, the system will throw corresponding error message stating the issue and if all conditions are met photo will be updated successfully.



Let's create the decision table for this case.

Prepared by: **Mr. Umar Arshad** | Software Quality Assurance Engineer

Conditions	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Format	.jpg	.jpg	.jpg	.jpg	Not .jpg	Not .jpg	Not .jpg	Not .jpg
Size	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb
resolution	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177
Output	Photo uploaded	Error message resolution mismatch	Error message size mismatch	Error message size and resolution mismatch	Error message for format mismatch	Error message format and resolution mismatch	Error message for format and size mismatch	

5.3. Equivalence Class Partitioning

Equivalence Partitioning or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In this technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.

- It divides the input data of software into different equivalence data classes.
- You can apply this technique, where there is a range in the input field.

Example 1: Equivalence and Boundary Value

- Let's consider the behavior of Order Pizza Text Box Below
- Pizza values 1 to 10 is considered valid. A success message is shown.

- While value 11 to 99 are considered invalid for order and an error message will appear, "Only 10 Pizza can be ordered"

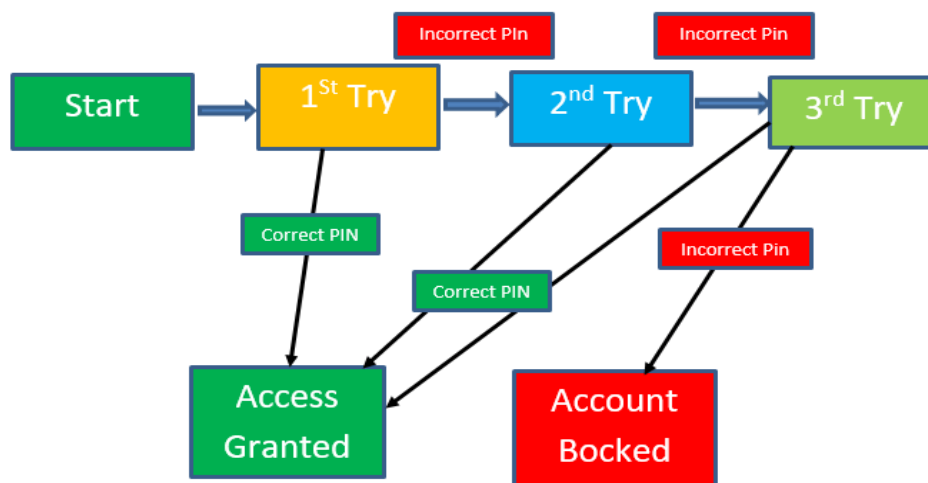
Order Pizza:

Here is the test condition

- Any Number greater than 10 entered in the Order Pizza field (let say 11) is considered invalid.
- Any Number less than 1 that is 0 or below, then it is considered invalid.
- Numbers 1 to 10 are considered valid
- Any 3 Digit Number say -100 is invalid.

5.4. What is State Transition?

Using state transition testing, we pick test cases from an application where we need to test different system transitions. We can apply this when an application gives a different output for the same input, depending on what has happened in the earlier state.



5.5. Error Guessing

Error Guessing is a software testing technique based on guessing the error which can prevail in the code. The technique is heavily based on the experience where the test analysts use their experience to guess the problematic part of the testing application. Hence, the test analysts must be skilled and experienced for better error guessing.

Chapter 6 Some Common Concepts

6.1. What is the difference between a Standalone application, Client-Server application and Web application?

Standalone application:

Standalone applications follow one-tier architecture. Presentation, Business, and Database layer are in one system for a single user.

Client-Server Application:

Client-server applications follow two-tier architecture. Presentation and Business layer are in a client system and Database layer on another server. It works majorly in Intranet.

Web Application:

Web server applications follow three-tier or n-tier architecture. The presentation layer is in a client system, a Business layer is in an application server and Database layer is in a Database server. It works both in Intranet and Internet.

6.2. What are main differences between UI and UX?

UX design refers to the term “user experience design”, while UI stands for “user interface design”.

There is an analogy I like to use to describe the different parts of a (digital) product:

If you imagine a product as the human body, the bones represent the code which give it structure.

The organs represent the UX design: measuring and optimizing against input for supporting life functions. And UI design represents the cosmetics of the body; its presentation, its senses and reactions.

6.3. Can you do System testing at any stage of SDLC?

We can do System Testing only when all the units are in place and working properly. It can only be done before User Acceptance Testing (UAT).

6.4. When to stop testing? (Or) How do you decide when you have tested enough? imp

There are many factors involved in the real-time projects to decide when to stop testing.

1. Testing deadlines or release deadlines
2. By reaching the decided pass percentage of test cases
3. The risk in the project is under acceptable limit
4. All the high priority bugs, blockers are fixed
5. When acceptance criteria are met.

6.5. What information should be included in a Defect or Bug report?

1. A brief summary of the defect
2. A full description of the defect including steps to reproduce
3. Screenshot attachments if required
4. Date the defect was found and raised
5. Who reported the defect?
6. Severity and/or Priority of the defect
7. Which component is the defect assigned?
8. Current status of Bug.

6.6. What are the test case for Glass of Water?

Check-list:

1. look & feel
2. height
3. weight
4. material used
5. radius at top & bottom
6. capacity
7. color
8. grip
9. breakable / unbreakable

6.7. What are the test case for AC and AC Remote?

Test scenarios for AC

10. Verify the type of AC, if it's Window AC or Split AC.
11. Verify the cooling capacity of the AC (ton). It should be as per the specifications.
12. Verify that the voltage requirement of the AC is as per the specifications.
13. Check if the AC voltage regulator works correctly or not.
14. Check the time taken by AC to start cooling, since the plug-in.
15. Check if it's working in every temperature setting.
16. Verify the minimum temperature that can be achieved by the AC.
17. Verify the maximum temperature that can be achieved by the AC.
18. Check if the different AC fan speeds, work correctly.
19. Verify that the information displayed in the display panel is correct.
20. Check if all the keys of the display panel are working.
21. Check if the display isn't too bright or too dark.
22. Check if the outer body of the AC is sturdy and rugged.
23. Check if the AC works with the remote.
24. Check if all the keys of the panel do the same work as mentioned.
25. Check if the grill is movable in all directions.
26. Verify if the AC switches off and on when reached the desired temperature, for saving electricity.
27. Check the amount of water leakage.

Test scenarios for AC remote

1. Verify the functionality of the on/off button.
2. Check if all the keys are in perfect condition.
3. Check if the remote belongs to the same AC.
4. Check if the name of the keys is mentioned.
5. Check the response time between the AC remote and the AC.
6. Check if the keys are doing the desired function or not.
7. Verify the durability of the keys and body of the remote.
8. Verify that the remote works each time old batteries are replaced.
9. Check if it's waterproof if mentioned.
10. Check if too much pressure isn't required to press the keys.
11. Check if the spacing among the keys is adequate.
12. Check the weight and dimensions of the remote.
13. Check if another function isn't triggered with any key other than specified.
14. Check if the body of the remote isn't flimsy.

Chapter 7 Road Map for Mobile Application Testing

What you need and what stages you need to test when you receive the Mobile Application, there are some major concerns I have listed and you should know.

7.1. Functional testing

Testing is done by certifying the requirements. like whether the application is working based on the requirements or not.

7.2. Android/iOS UI/Responsiveness testing

This is a user-centric testing of the application. In this test phase, items such as visibility of text in various screens of the app, interactive messages, alignment of elements, the look and feel of the app for different screens, size of fields etc. are tested under this.

Most important point of this testing:

- Device Selection (must prefer Real device always)
- Device emulators are cost effective and they come in handy during the initial development phase.
- But, to test the real-life scenarios, physical devices are the must. Both emulators and physical devices are to be used in a balanced manner for an optimized result.

7.3. Compatibility testing

The extension for Android apps is .APK. and for iOS apps is .ipa should be confirm. This testing is done mostly in the form of two matrices of OS Vs app and Device Model Vs App. Usually, a list of supported OS (and sometimes devices) is provided by the product owner or customer.

7.4. Interface Testing

This testing is done after all the modules of the app are completely developed, tested individually and all the bugs are fixed verified.

7.5. Network Testing

During this testing, request/response to/from the service is tested for various conditions.

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

This test is mainly done to verify the response time in which the activity is performed like refreshing data after sync or loading data after login etc.

7.6. Performance Testing

Performance of the application under some peculiar conditions are checked. Those conditions include:

- Low memory in the device.
- The battery is extremely at a low level.
- Poor/Bad network reception.

7.7. Installation/Uninstallation testing

This is to ensure smooth installation and uninstallation of the application without ending up in errors, partial installation etc.

7.8. Security Testing

Testing of the data flow for encryption and decryption mechanism is to be tested in this phase. Access to stored data is also tested in this phase.

7.9. Field testing

Field testing is done specifically for the mobile data network and not in-house but by going out and using the app as a normal user.

It is basically done to verify the behavior of the app when the phone has a 2G or 3G connection. Field testing verifies if the app is crashing under slow network connection or if it is taking too long to load the information.

7.10. Interrupt Testing

This is the Offline Scenario Verification. Conditions where the communication breaks in the middle are called as offline conditions.

Some of the conditions where interruptions of a network can be tested are as follows:

- Data cable removal during data transfer process.
- Network outage during the transaction posting phase.
- Network recovery after an outage.
- Battery removal or Power On/Off when it is in the transactional phase.

Chapter 8 Agile Methodology and Concepts

8.1. What is Agile Testing?

Agile Testing is a practice that a QA follows in a dynamic environment where testing requirements keep changing according to customer needs. It is done parallel to the development activity where the testing team receives frequent small codes from the development team for testing.

8.2. What is the difference between burn-up and burn-down charts?

Burn-up and burn-down charts are used to keep track of the progress of the project. Burn-up charts represent how much work has been completed in any project whereas Burn-down chart represents the remaining work in a project.

8.3. Define the roles in Scrum?

There are mainly three roles that a Scrum team have:

1. **Project Owner** has the responsibility of managing the product backlog. Works with end-users and customers and provides proper requirements to the team to build the proper product.
2. **Scrum Master** works with the scrum team to make sure each sprint gets completed on time. Scrum master ensures proper workflow for the team.
3. **Scrum Team**: Each member of the team should be self-organized, dedicated and responsible for the high quality of the work.

8.4. What is Product Backlog & Sprint Backlog?

Product backlog is maintained by the project owner which contains every feature and requirement of the product.

Sprint backlog can be treated as the subset of product backlog which contains features and requirements related to that particular sprint only.

8.5. Explain Velocity in Agile.

Velocity is a metric that is calculated by the addition of all efforts estimates associated with user stories completed in an iteration. It predicts how much work Agile can complete in a sprint and how much time will it require to complete a project.

8.6. Explain the difference between a traditional Waterfall model and Agile testing?

Agile testing is done parallel to the development activity whereas a traditional waterfall model testing is done at the end of the development. As done in parallel, agile testing is done on small features whereas, in a waterfall model, testing is performed on the whole application.

8.7. Explain Pair Programming and its benefits?

Pair programming is a technique in which two programmer works as a team in which one programmer writes code and other one reviews that code. They both can switch their roles.

Benefits:

- **Improved code quality:** As the second partner reviews the code simultaneously, it reduces the chances of mistake.
- **Knowledge transfer is easy:** One experienced partner can teach another partner about the techniques and codes.

8.8. What is Re-factoring?

Modification of the code without changing its functionality to improve the performance is called Re-factoring.

8.9. Explain the Iterative and Incremental Development in Agile?

Iterative Development: Software is developed and delivered to the customer and based on the feedback again developed in cycles or releases and sprints.

Example: Release 1 software is developed in 5 sprints and delivered to the customer. Now, the customer wants some changes, then the development team plan for 2nd release which can be completed in some sprints and so on.

Incremental Development: Software is developed in parts or increments. In each increment, a portion of the complete requirement is delivered.

8.10. How do you deal when requirements change frequently?

This question is to test the analytical capability of the candidate.

The answer can be: Work with PO to understand the exact requirement to update test cases. Also, understand the risk of changing the requirement. Apart from this, one should be able to write a generic test plan and test cases. Don't go for the automation until requirements are finalized.

8.11. What is a test stub?

Test stub is a small code that mimics(copy) a specific component in the system and can replace it. Its output is the same as the component it replaces.

8.12. What qualities should a good Agile tester have?

- He should be able to understand the requirements quickly.
- He should know Agile concepts and principals.
- As requirements keep changing, he should understand the risk involved in it.
- The agile tester should be able to prioritize the work based on the requirements.
- Communication is a must for an Agile tester as it requires a lot of communication with developers and business associates.

8.13. What is the difference between Epic, User stories & Tasks?

User Stories: It defines the actual business requirement. Generally created by the business owner.

Task: To accomplish the business requirements development team create tasks.

Epic: A group of related user stories is called an Epic.

8.14. What is a Task board in Agile?

Task board is a dashboard that shows the progress of the project.

It contains:

- **User Story:** It has the actual business requirement, which consist of User Acceptance Criteria(UAC).
- **To Do:** Tasks that can be worked on.
- **In Progress:** Tasks in progress.
- **To Verify:** Tasks pending for verification or testing
- **Done:** Completed tasks.

8.15. What is Test Driven Development (TDD)?

It is a Test-first development technique in which we add a test first before we write the complete production code. Next, we run the test and based on the result refactor the code to fulfill the test requirement.

8.16. How QA can add value to an agile team?

QA can provide value addition by think outside the box about the various scenarios to test a story. They can provide quick feedback to the developers about whether new functionality is working fine or not.

8.17. What is Scrum ban?

It is a software development model that is a combination of Scrum and Kanban. Scrum ban is considered for maintaining projects in which there are frequent changes or unexpected user stories. It can reduce the minimum completion time for user stories.

8.18. What is the Application Binary Interface?

Application Binary Interface or ABI is defined as an interface for complied application programs or we can say it describes the low-level interface between an application and the operating system.

8.19. What is the Zero sprint in Agile?

It can be defined as a pre-preparation step to the first sprint. Activities like setting development environment, preparing backlog, etc need to be done before starting the first sprint and can be treated as Sprint zero.

8.20. What is Spike?

There may be some technical issues or design problem in the project which needs to be resolved first. To provide the solution to this problem “Spikes” are created.
Spikes are of two types- Functional and Technical.

8.21. Name some Agile quality strategies.

Some Agile quality strategies are-

1. Re-factoring
2. Small feedback cycles
3. Dynamic code analysis
4. Iteration

8.22. What is the importance of daily stand-up meetings?

Daily stand-up meeting is essential for any team in which team discuss,

1. How much work has been completed?
2. What are the plans to resolve technical issues?

3. What steps need to be done to complete the projects etc.?

8.23. What is a tracer bullet?

It can be defined as a spike with the current architecture or the current set of best practices. The purpose of a tracer bullet is to examine how an end-to-end process will work and examine feasibility.

8.24. How the velocity of the sprint is measured?

If capacity is measured as a percentage of a 40 hours' weeks, then, completed story points * team capacity

If capacity is measured in man-hours, then Completed story points/team capacity.

8.25. How many types of SCRUM Meeting?

There are 5 types of Scrum meetings which occur at a particular time during a Sprint cycle and each particular type serves a distinct purpose.

1. Sprint Planning Meeting
2. Daily Scrum Meeting
3. Sprint Review Meeting
4. Sprint Retrospective Meeting
5. Backlog Refinement Meeting

Sprint Planning Meeting

At the beginning of every Sprint, Sprint planning meeting is held. Usually, the entire team is expected to be present during this meeting, including the product owner and the Scrum master. The goal of this meeting is to develop realistic Sprint backlog and define the highest priority tasks which need to be done during the length of each Sprint.

Daily Scrum Meeting

Daily Scrum meeting, or daily standups – as many people call them, are short 15 minutes' meetings which occur on daily basis. They are typically held at the same time and same place every day and are strictly time boxed to no longer than 15 minutes. This ensures the discussion to stay light, relevant, and quick.

- What did you accomplish yesterday?
- What are you working on today?

- Are there any impediments in your way?

Sprint Review Meeting

At the end of each Sprint, a Sprint Review meeting is held. The core objective of this meeting is to demonstrate the functionality of the product and what has been achieved during a particular Sprint. Generally, product owner, Scrum Master, and other stakeholders are present to review the product.

Sprint Retrospective Meeting

A retrospective is a meeting held after a product ships to discuss what happened during the product development and release process, with the goal of improving things in the future based on those learnings and conversations.

I Hope, these questions will help you in preparing for the Agile testing and methodology interview.

Thanks!

Chapter 9 Database & SQL Concepts

9.1. What is Database

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Database management system stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

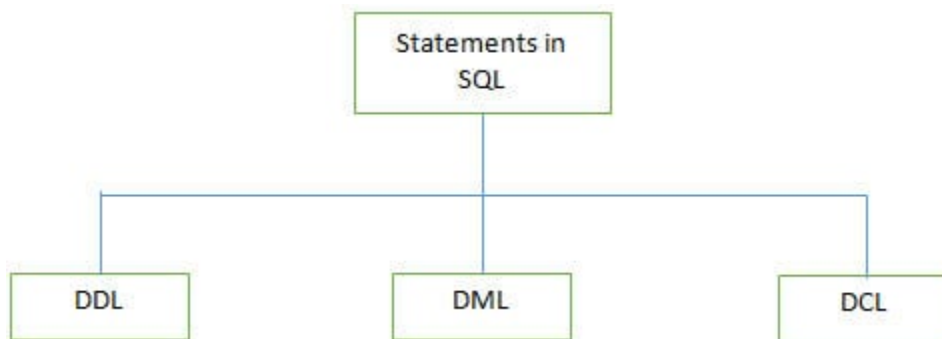
9.2. What is SQL?

Structured Query Language SQL is a database tool that is used to create and access the database to support software applications.

9.3. What are tables in SQL?

The table is a collection of record and its information at a single view.

9.4. What are the different types of statements supported by SQL?



There are 3 types of SQL statements:

a. **DDL (Data Definition Language):**

It is used to define the database structure such as tables. It includes three statements such as CREATE, ALTER, DROP and TRUNCATE.

Some of the DDL Commands are listed below:

CREATE: It is used for creating the table.

```
CREATE TABLE table_name  
  
column_name1 data_type(size),  
column_name2 data_type(size),  
column_name3 data_type(size),
```

ALTER: The ALTER table is used for modifying the existing table object in the database.

```
ALTER TABLE table_name  
  
ADD column_name datatype
```

OR

```
ALTER TABLE table_name  
  
DROP COLUMN column_name
```

b. DML (Data Manipulation Language):

These statements are used to manipulate the data in records. Commonly used DML statements are INSERT, UPDATE, and DELETE.

The SELECT statement is used as a partial DML statement, used to select all or relevant records in the table.

c. DCL (Data Control Language):

These statements are used to set privileges such as GRANT and REVOKE database access permission to the specific user.

9.5. How do we use the DISTINCT statement? What is its use?

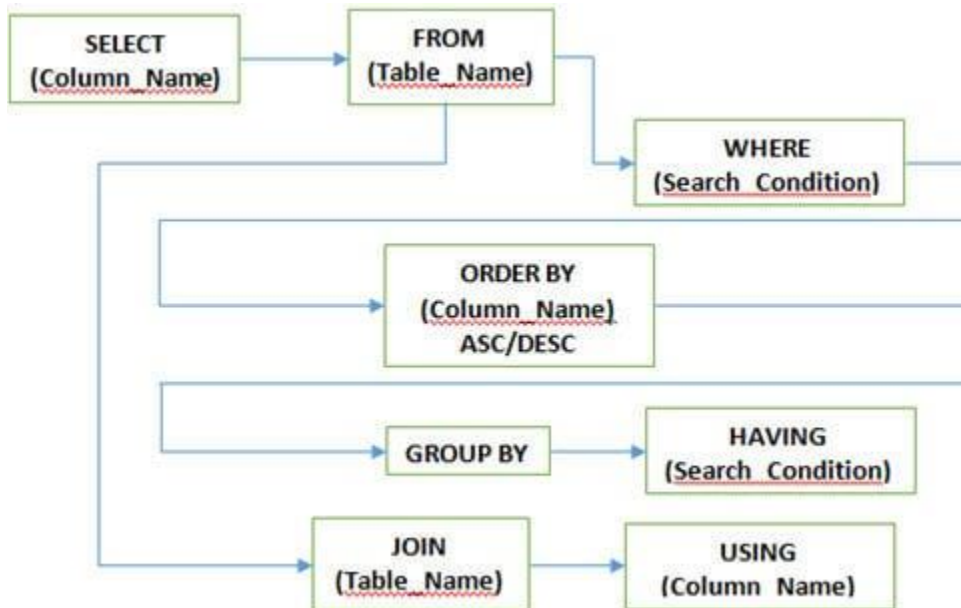
The DISTINCT statement is used with the SELECT statement. If the record contains duplicate values, then the DISTINCT statement is used to select different values among duplicate records.

Syntax:

```
SELECT DISTINCT column_name(s)  
  
FROM table_name;
```

9.6. What are the different Clauses used in SQL?

Answer:



WHERE Clause: This clause is used to define the condition, extract and display only those records which fulfill the given condition.

Syntax:

```
SELECT column_name(s)

FROM table_name

WHERE condition;
```

GROUP BY Clause: It is used with SELECT statement to group the result of the executed query using the value specified in it. It matches the value with the column name in tables and groups the end result accordingly.

Syntax:

```
SELECT column_name(s)

FROM table_name

GROUP BY column_name;
```

HAVING clause: This clause is used in association with the GROUP BY clause. It is applied to each group of results or the entire result as a single group. It is much similar as WHERE clause but the only difference is you cannot use it without GROUP BY clause

Syntax:

```
SELECT column_name(s)

FROM table_name

GROUP BY column_name
```

```
HAVING condition;
```

ORDER BY clause: This clause is used to define the order of the query output either in ascending (ASC) or in descending (DESC). Ascending (ASC) is set as the default one but descending (DESC) is set explicitly.

Syntax:

```
SELECT column_name(s)

FROM table_name

WHERE condition

ORDER BY column_name ASC|DESC;
```

USING clause: USING clause comes in use while working with SQL JOIN. It is used to check equality based on columns when tables are joined. It can be used instead of the ON clause in JOIN.

Syntax:

```
SELECT column_name(s)

FROM table_name

JOIN table_name

USING (column_name);
```

9.7. Why do we use SQL constraints? Which constraints we can use while creating a database in SQL?

Constraints are used to set the rules for all records in the table. If any constraints get violated then it can abort the action that caused it.

Constraints are defined while creating the database itself with the CREATE TABLE statement or even after the table is created once with the ALTER TABLE statement.

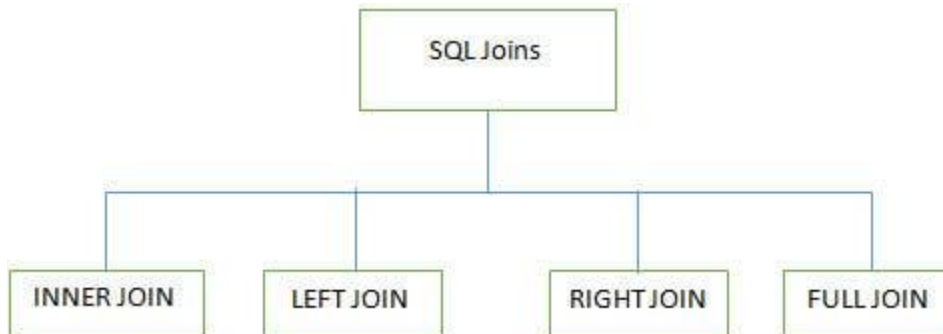
There are 5 major constraints are used in SQL, such as

- **NOT NULL:** That indicates that the column must have some value and cannot be left NULL.
- **UNIQUE:** This constraint is used to ensure that each row and column has a unique value and no value is being repeated in any other row or column.
- **PRIMARY KEY:** This constraint is used in association with NOT NULL and UNIQUE constraints such as on one or the combination of more than one column to identify the particular record with a unique identity.
- **FOREIGN KEY:** It is used to ensure the referential integrity of data in the table. It matches the value in one table with another using the PRIMARY KEY.
- **CHECK:** It ensures whether the value in columns fulfills the specified condition.

9.8. What is the difference between Null and Empty?

The main difference between null and empty is that the null is used to refer to nothing while empty is used to refer to a unique string with zero length.

9.9. What are different JOINS used in SQL?



4 major types of Joins are used while working on multiple tables in SQL databases:

INNER JOIN: It is also known as SIMPLE JOIN which returns all rows from BOTH tables when it has at least one matching column.

Syntax:

```
SELECT column_name(s)

FROM table_name1 

INNER JOIN table_name2

ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
INNER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

There will be 4 records selected. **Results are:**

Emp_Id	Joining_Date
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

Employee and **Orders** tables have a matching customer_id value.

LEFT JOIN (LEFT OUTER JOIN): This join returns all rows from the LEFT table and its matched rows from a RIGHT table.

Syntax:

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01
NULL	NULL	NULL	2016/03/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
LEFT OUTER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

There will be 4 records selected. **You will see the following results:**

Emp_Id	Joining_Date
NULL	NULL
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

RIGHT JOIN (RIGHT OUTER JOIN): This join returns all rows from the RIGHT table and its matched rows from the LEFT table.

Syntax:

```
SELECT column_name(s)
FROM table_name1
RIGHT JOIN table_name2
ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01
NULL	NULL	NULL	2016/03/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date FROM Employee
RIGHT JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

Output:

Emp_id	Joining_Date
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

FULL JOIN (FULL OUTER JOIN): This join returns all results when there is a match either in the RIGHT table or in the LEFT table.

Syntax:

```
SELECT column_name(s)
FROM table_name1
FULL OUTER JOIN table_name2
ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01
NULL	NULL	NULL	2016/03/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date
```

```
FROM Employee
```

```
FULL OUTER JOIN Joining
```

```
ON Employee.Emp_id = Joining.Emp_id
```

```
ORDER BY Employee.Emp_id;
```

There will be 8 records selected. **These are the results that you should see.**

Emp_Id	Joining_Date
NULL	NULL
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01
NULL	2016/03/01
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

9.10. What are transactions and their controls?

A transaction can be defined as the sequence task that is performed on databases in a logical manner to gain certain results. Operations like Creating, updating, deleting records performed in the database come from transactions.

In simple words, we can say that a transaction means a group of SQL queries executed on database records.

There are 4 transaction controls such as

- **COMMIT:** It is used to save all changes made through the transaction.
- **ROLLBACK:** It is used to roll back the transaction. All changes made by the transaction are reverted back and the database remains as before.
- **SET TRANSACTION:** Set the name of the transaction.
- **SAVEPOINT:** It is used to set the point where the transaction is to be rolled back.

9.11. What are the properties of the transaction?

Properties of the transaction are known as ACID properties. These are:

- **Atomicity:** Ensures the completeness of all transactions performed. Checks whether every transaction is completed successfully or not. If not, then the transaction is aborted at the failure point and the previous transaction is rolled back to its initial state as changes are undone.
- **Consistency:** Ensures that all changes made through successful transactions are reflected properly on the database.
- **Isolation:** Ensures that all transactions are performed independently and changes made by one transaction are not reflected on others.
- **Durability:** Ensures that the changes made in the database with committed transactions persist as it is even after a system failure.

9.12. How many Aggregate functions are available in SQL?

SQL Aggregate functions determine and calculate values from multiple columns in a table and return a single value.

There are 7 aggregate functions in SQL:

- **AVG():** Returns the average value from specified columns.
- **COUNT():** Returns number of table rows.
- **MAX():** Returns the largest value among the records.
- **MIN():** Returns smallest value among the records.
- **SUM():** Returns the sum of specified column values.
- **FIRST():** Returns the first value.
- **LAST():** Returns last value.

9.13. What are Scalar functions in SQL?

Scalar functions are used to return a single value based on the input values.

Scalar Functions are as follows:

- **UCASE():** Converts the specified field in the upper case.
- **LCASE():** Converts the specified field in lower case.
- **MID():** Extracts and returns character from the text field.
- **FORMAT():** Specifies the display format.
- **LEN():** Specifies the length of the text field.
- **ROUND():** Rounds up the decimal field value to a number.

9.14. What are triggers?

Triggers in SQL is kind of stored procedures used to create a response to a specific action performed on the table such as INSERT, UPDATE or DELETE. You can invoke triggers explicitly on the table in the database.

Action and **Event** are two main components of SQL triggers. When certain actions are performed, the event occurs in response to that action.

Syntax:

```
CREATE TRIGGER name {BEFORE|AFTER} (event [OR..])
```

```
ON table_name [FOR [EACH] {ROW|STATEMENT}]
```

```
EXECUTE PROCEDURE functionname {arguments}
```

9.15. What is View in SQL?

A View can be defined as a virtual table that contains rows and columns with fields from one or more tables.

Syntax:

```
CREATE VIEW view_name AS
```

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE condition
```

9.16. How we can update the view?

SQL CREATE and REPLACE can be used for updating the view.

Execute the below query to update the created view.

Syntax:

```
CREATE OR REPLACE VIEW view_name AS
```

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE condition
```

9.17. Explain the working of SQL Privileges?

SQL GRANT and REVOKE commands are used to implement privileges in SQL multiple user environments. The administrator of the database can grant or revoke privileges to or from users of database objects by using commands like SELECT, INSERT, UPDATE, DELETE, ALL, etc.

GRANT Command: This command is used to provide database access to users other than the administrator.

Syntax:

```
GRANT privilege_name
```

```
ON object_name
```

```
TO {user_name|PUBLIC|role_name}
```

```
[WITH GRANT OPTION];
```

In the above syntax, the GRANT option indicates that the user can grant access to another user too.

REVOKE command: This command is used to provide database deny or remove access to database objects.

Syntax:

```
REVOKE privilege_name
```

```
ON object_name
```

```
FROM {user_name|PUBLIC|role_name};
```

9.18. How many types of Privileges are available in SQL?

There are two types of privileges used in SQL, such as

- **System privilege:** System privilege deals with the object of a particular type and provides users the right to perform one or more actions on it. These actions include performing administrative tasks, ALTER ANY INDEX, ALTER ANY CACHE GROUP CREATE/ALTER/DELETE TABLE, CREATE/ALTER/DELETE VIEW etc.
- **Object privilege:** This allows to perform actions on an object or object of another user(s) viz. table, view, indexes etc. Some of the object privileges are EXECUTE, INSERT, UPDATE, DELETE, SELECT, FLUSH, LOAD, INDEX, REFERENCES etc.

9.19. What is SQL Injection?

SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database in a way that once it is executed, the database is exposed to an attacker for the attack. This technique is

usually used for attacking data-driven applications to have access to sensitive data and perform administrative tasks on databases.

For Example,

```
SELECT column_name(s) FROM table_name WHERE condition;
```

9.20. What is SQL Sandbox in SQL Server?

SQL Sandbox is a safe place in the SQL server environment where untrusted scripts are executed. There are 3 types of SQL sandbox:

- **Safe Access Sandbox:** Here a user can perform SQL operations such as creating stored procedures, triggers etc. but cannot have access to the memory as well as cannot create files.
- **External Access Sandbox:** Users can access files without having the right to manipulate the memory allocation.
- **Unsafe Access Sandbox:** This contains untrusted codes where a user can have access to memory.

9.21. What is the difference between SQL and PL/SQL?

SQL is a Structured Query Language to create and access databases whereas PL/SQL comes with procedural concepts of programming languages.

9.22. What is the difference between SQL and MySQL?

SQL is a Structured Query Language that is used for manipulating and accessing the relational database. On the other hand, MySQL itself is a relational database that uses SQL as the standard database language.

9.23. What is the use of the NVL function?

NVL function is used to convert the null value to its actual value.

9.24. What do you mean by Subquery?

Query within another query is called as Subquery. A subquery is called inner query which returns output that is to be used by another query.

9.25. How many row comparison operators are used while working with a subquery?

There are 3-row comparison operators that are used in subqueries such as IN, ANY and ALL.

- **Second Normal Form (2NF):** Follows 1NF and creates and places data subsets in an individual table and defines the relationship between tables using the primary key.
- **Third Normal Form (3NF):** Follows 2NF and removes those columns which are not related through the primary key.
- **Fourth Normal Form (4NF):** Follows 3NF and does not define multi-valued dependencies. 4NF is also known as BCNF.

9.31. What is a Relationship? How many types of Relationships are there?

The relationship can be defined as the connection between more than one table in the database.

There are 4 types of relationships:

- One to One Relationship
- Many to One Relationship
- Many to Many Relationship
- One to Many Relationship

9.32. What is Collation?

Collation is a set of rules that check how the data is sorted by comparing it. Such as character data is stored using correct character sequence along with case sensitivity, type, and accent.

9.33. What do we need to check in Database Testing?

In Database testing, the following thing is required to be tested:

- Database connectivity
- Constraint check
- Required application field and its size
- Data Retrieval and processing with DML operations
- Stored Procedures
- Functional flow

9.34. What is Database White Box Testing?

Database White Box testing involves:

- Database Consistency and ACID properties
- Database triggers and logical views
- Decision Coverage, Condition Coverage, and Statement Coverage
- Database Tables, Data Model, and Database Schema
- Referential integrity rules

9.35. What is Database Black Box Testing?

Database Black Box testing involves:

- Data Mapping
- Data stored and retrieved
- Use of Black Box testing techniques such as Equivalence Partitioning and Boundary Value Analysis (BVA)

Chapter 10 Common SQL Queries

10.1. How to write query for find the highest salary from employees table.

```
SELECT name, MAX(salary) as salary FROM employee
```

10.2. How to write query for find the second highest salary from employees table.

1st Method

```
SELECT name, MAX(salary) AS salary  
FROM employee  
WHERE salary < (SELECT MAX(salary)  
FROM employee);
```

2nd Method

For Second max salary

```
Select name, salary From Employee ORDER BY Salary DESC LIMIT  
1,1;
```

For third max salary

```
Select salary From Employee ORDER BY Salary DESC LIMIT 2,1;
```

10.3. How to write a query to show the details of a student from Students table whose name start with K?

```
SELECT * FROM Student WHERE Student_Name like 'K%';
```

Here 'like' operator is used to perform pattern matching.

10.4. How to write the query to find the total no of columns in Table?

```
SELECT TABLE_NAME , count(COLUMN_NAME)  
FROM information_schema.columns
```

Chapter 11 Application Program Interface (API)

Many of the new modern web applications are built using web-services, micro-services, and APIs. As testers, we should be knowledgeable and experienced in testing APIs and Web Services.

Here are some fundamental API Testing Interview Questions mainly aimed at software testers.

11.1. What is API Testing?

API is a collection of routines, tools, protocols that together are required for building the software application. Any system software or application software which consists of multiple APIs can perform Application Programming Interface (API) testing. This form of testing includes interaction between various or says multiple APIs as well as the interaction between API and application program. The procedure mainly includes making API calls using software and observing system response after receiving the output.

11.2. Enlist some common tests that are performed on APIs.

There can be multiple reasons for performing API testing.

Let us see some common test examples, where this form of testing is used to verify:

- Any data structure updated by API which requires proper validation.
- Input condition-based return values.
- Call to another API or if any other event is triggered or some interruption is raised.
- The return values can also be null or wrong results.
- Modification of some resources like an update of the database, process killing, etc.

11.3. What is the API test environment?

Setting up a test environment of API is a complex method where the configuration of the server and database is done as per the requirement of the software application. Graphical User Interface (GUI) is not available in this form of testing. After installation, API is verified for its proper functioning. In this process initial environment that invokes API is being set up with a defined set of parameters so that test results can be examined.

11.4. Explain the API testing approach.

Mentioned below are the factors which determine the approach:

- Write appropriate test cases for the APIs and use testing techniques like boundary value analysis, equivalence class, etc. for verifying the functionality.
- Verify the calls of the combination of two or more value-added parameters.
- Define the scope and basic functionality of the API program.
- Define the accurate input parameters.
- Test case execution and comparison of the results with expected results.
- Determining API behavior under conditions like the connection with files, etc.

11.5. Explain in brief the different types of output observed of an API.

API is considered as the essential connecting part of this digital world. It basically resides in the business logic layer where it performs functions like processing commands, application coordination, initiates logical decisions, etc. The main consideration is returning correct results under any type of conditions. Mainly, the output or results observed of an API are divided into three sections as follows:

- Returning the result status values as 'Pass' or 'Fail'.
- Result as data or any specific information.
- An event where the call to any API function will initiate the call to another API function.

11.6. Difference between SOAP and REST API?

- SOAP stands for Simple Object Access Protocol whereas REST stands for Representational State Transfer.
- SOAP is state full and REST is state less.
- SOAP is a protocol whereas REST is an architectural pattern.
- SOAP uses service interfaces to expose its functionality to client applications while REST uses Uniform Service locators to access to the components on the hardware device.
- SOAP needs more bandwidth for its usage whereas REST doesn't need much bandwidth.
- Comparing SOAP vs REST API, SOAP only works with XML formats whereas REST work with plain text, XML, HTML and JSON.
- SOAP cannot make use of REST whereas REST can make use of SOAP.

11.7. Enlist some best practices that are followed to make API testing successful.

Performing tests repeatedly define some best practices for making testing successful.

Enlisted below are some best practices for API testing:

- Test cases should be grouped under category with expected results that happen consistently and other typical results.
- Test cases should include selected parameters as well as API call declarations.
- API load tests are performed to determine system application stress.

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

- Maintain the limits of the variables used in the tests as well as avoid 'Test Chaining'.
- To make ease for the testers, API call is being prioritized and call sequencing is planned.
- Every input combination and dependencies are considered for complete test coverage.
- Automation of the test cases, documentation is done as and when required.

11.8. What are the tools used for API testing?

Best API Testing tools:

- SOAPUI
- Runscope
- LOADUI
- Automated API testing
- Curl
- Postman

There are a few more others than the above-listed tools that are used for API testing.

11.9. What are the tools used for API test automation?

Automation testing is a must when we talk about agile development in API testing. However, the language in which the code is written is also an important factor as it decides the tool language.

Some important API test automation tools are:

- **SOAPUI:** It is an open-source API testing tool which is considered as the best testing tool because of its feature like creating complex validation scripts and test cases, efficient test coverage, etc.
- **HP QTP/UFT:** This is now known as HP UFT i.e., Unified Functional Testing. This tool is basically used for systems without user interface like web services, etc.
- **PARASOFT:** This testing tool runs on various platforms and is used to test API which does not have a Graphical User interface (GUI).
- HTTP master
- NUnit and JUnit testing tools are used where the code is written in .Net and Java respectively.

11.10. What is the API framework?

API framework is described by the config file which consists of the list of all APIs that are required to be activated and are activated for any particular program run. This is essential as every test run does not require all APIs.

The purpose of the 'Config' file is to describe and enlist every configurable component within a test run.

11.11. Explain API documentation.

As it is a well aware fact that, for any foundation, there has to be good documentation. API documentation likewise, serves as a quick reference for accessing the library or working within a program.

When we go through any such documents, it must consist of a proper plan, content source, proper layout or sketch for delivery, information related to each function, etc.

API documentation tools are:

- JavaDoc
- Doxygen

Enlisted below are the categories in which every function is being documented which mainly revolve around the parameters:

- Function description
- Sequence, syntax, and elements required for each parameter.
- Syntax and type of error message that can occur.
- Links related to functions.

11.12. Name some most used templates for API documentation.

Some free templates which make API documentation much easier and simple are:

- Slate
- FlatDoc
- Swagger
- API blueprint
- RestDoc
- Miredot
- Web service API Specification.

11.13. Enlist some of the API examples which are very well known and popular.

There are several such examples. **Enlisted below are some most popular ones:**

- **Google Maps API:** These are designed mainly for mobile and desktop use with the help of a flash interface and JavaScript.
- **Amazon Advertising API:** Amazon is known for their products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly.
- **Twitter:** The API for twitter is usually in two categories, one for accessing data and the other for interacting with the twitter search.
- **YouTube:** This API used for YouTube includes various functionalities including videos, live streaming, player, etc.

11.14. What are the testing methods that come under API testing?

API testing generally involves the following testing methods:

- Unit testing and Functional testing
- Load testing for testing the performance under load.

- Discovery testing for listing, creating and deleting the number of calls that have been documented in API.
- Usability testing and Reliability testing for obtaining consistent results.
- Security testing and Penetration testing for validating all types of authentication.
- Automation testing for creating and executing scripts that require API calls execution regularly.
- End to end Integration testing and Web UI testing.
- API documentation testing for determining its efficiency and effectiveness.

11.15. Differentiate API testing and Unit Testing.

The difference between API testing and Unit testing can be understood from the below table:

UNIT testing	API Testing
Unit testing is usually performed by developers where every functionality is tested separately.	API testing is performed by the testers for end to end testing of the functionality.
As they have the limited scope of testing, thus basic functionalities are only considered for testing.	As they have the broader scope of testing, all issues that are functional are considered for testing.
It is a form of white box testing.	It is a form of black box testing.
Usually, unit testing is done before the code is included in the build.	API testing is performed after the build is ready for testing.
The Source code is involved in this form of testing.	Source code is not involved in this form of testing.

11.16. What challenges are included under API testing?

Challenges are the part of every form of testing and the same goes with API testing too.

Mentioned below are some common challenges that are faced in API testing:

- The first and foremost challenge is selecting an appropriate parameter and then its combination.
- Parameter categorization
- Proper sequencing of call is required as this may lead to inadequate coverage in testing.
- Output verification and validation
- Another important challenge is providing input values, which is very difficult as GUI is not available in this case.

11.17. What are the types of issues observed while performing API testing?

When testing is performed, then there have to be issues associated with them. Issues observed while performing this form of testing are not new or much different but they are common in this category.

Find below the list of such issues/defects:

- Inconsistent or absence of error handling mechanism
- Repetition or redundancy of the functionalities
- Missing required functionality in some cases
- Passing incorrect argument to the input values
- Improper messaging
- Stress and performance issues
- Reliability issues with respect to connection with other APIs
- Multithreading and improper handling issues.

11.18. Why API testing is determined as the most suitable form for Automation testing?

Yes, it's true that API testing is now preferred over GUI testing and is considered as most suitable.

Below are the few reasons behind this statement.

- Verify all the functional paths of the system under test very effectively.
- Provides the most stable interface.
- Easier to maintain and provides fast feedback.

11.19. How is UI level testing different from API testing?

The main consideration of the UI (User Interface) level testing is to test the graphical interface part of the application include features like font, layout, etc.

Whereas, the main consideration of the API testing is establishing communication between different software systems and it mainly resides in business logic layer. It never concentrates on the look of the application.

11.20. What is Test API?

Test Api is known as the library of test building blocks which are essential for developers and testers for creating testing tools as well as automated test suites.

11.21. What do you know about API errors and warnings?

When something goes wrong i.e. the outcome is not as expected then the error occurs and warnings are described as a message in the proper format. There can be one or multiple warnings within the same module.

Different types of warnings that can occur are:

- Parameter validation warning
- Missing module warning

Different types of errors that can occur are:

- Documentation errors

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

- Missing module errors
- Parameter validation errors
- Some standard error messages.

11.22. What are HTTP Request and HTTP Response?

POST: The POST is most-often utilized to **create** new resources.

GET: The HTTP GET method is used to **read** (or retrieve) a representation of a resource.

PUT: PUT is most-often utilized for **update** capabilities.

PATCH: PATCH is used for **modify** capabilities.

DELETE: DELETE is pretty easy to understand. It is used to **delete** a resource identified by a URI.

An HTTP **request** method is made up of four components:

- **Request Method** – Get, Post, Put, Delete (these are the common ones)
- **Request URI** – the URL of the resource
- **Request Header** – Accept-Language, Accept-Encoding, User-Agent, Host
- **Request Body** – this is the data to be sent to the resource

An HTTP **response** method is made up of three components:

- **Response Status Code** – 200, 301, 404, 500 (these are the most common ones)
- **Response Header Fields** – Date, Server, Last-Modified, Content-Type
- **Response Body** – this is the data that comes back to the client from the server

11.23. What are the different HTTP Status codes?

2x Success

Code	Text	Purpose
200	OK	For successful GET and PUT requests.
201	Created	For a successful POST request.

Code	Text	Purpose
202	Accepted	For a request that resulted in a scheduled task being created to perform the actual request.
204	No Content	For a successful request that produced no response (such as DELETE requests).

3x Redirection

Code	Text	Purpose
301	Moved Permanently/Redirection	When the API routes have changed (unlikely), or if the incoming request is not secure (http), the request will be redirected to the secure (https) version.
304	Not Modified	This response will be sent if the request included an If-Modified-Since header, but the resource has not been modified since the specified date.

4xx Client Error

Code	Text	Purpose
400	Bad Request	Issued when a malformed request was sent.
401	Unauthorized	This response is sent when your client failed to provide credentials or its credentials were invalid.
403	Forbidden	Returned when permissions do not allow the operation.
404	Not Found	When a particular resource doesn't exist or couldn't be found.

Code	Text	Purpose
405	Method Not Allowed	The resource was found, but doesn't support the request method.
406	Not Acceptable	When the client specifies a response content type in the <code>Accept</code> header that is not supported.
409	Conflict	A change requested by the client is being rejected, due to a condition imposed by the server. The exact reasons for this response will vary from one resource to the next
413	Request Entity Too Large	When the client requests too many objects. For example, the <code>limit</code> parameter exceeded the maximum.
415	Unsupported Media Type	Returned due to issues with the <code>Content-Type</code> header.
422	Missing or Invalid Data	The request cannot be processed either because it omitted required fields or because it contained invalid data. See the response for more details.
429	Too Many Requests	When an OAuth client exceeds the rate limit for API requests to a store.

5xx Server Error

Code	Text	Purpose
500	Internal Server Error	When an error has occurred within the API.
501	Not Implemented	When a request method is sent that is not supported by the API (e.g., <code>TRACE</code> , <code>PATCH</code>).
503	Service Unavailable	When the store is "Down for Maintenance," being upgraded to a new version, or is suspended due to administrative action or a billing issue.

Code	Text	Purpose
507	Insufficient Storage	When the store has reached a limitation for the resource.

Hope this article will be very helpful for your API Testing interview preparation

Chapter 12 Object Oriented Programming Basics (OOP)

12.1. What is OOP?

The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

12.2. What is the Access Modifier?

Defines the access type of the method i.e. from where it can be accessed in your application.

- **Public:** accessible in all classes in your application.
- **Protected:** accessible within the package in which it is defined and in its subclass(es)(including subclasses declared outside the package)
- **Private:** accessible only within the class in which it is defined.
- **Default** (declared/defined without using any modifier): accessible within same class and package within which its class is defined.

12.3. What is the class?

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties/variables or methods that are common to all objects of one type.

12.4. What is Object?

Object is a basic unit of Object-Oriented Programming and represents real-life entities. A typical Java program creates many objects, which as you know, interact by invoking methods.

12.5. What is Method?

A method is a collection of statements that perform some specific task and return the result to the caller. A method can perform some specific task without returning anything. Methods allow us to reuse the code without retyping the code.

12.6. What is abstraction?

Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essential units are not displayed to the user.

Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details.

Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car.

12.7. What is Encapsulation?

It is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Another way to think about encapsulation is, it is a protective shield that prevents the data from being accessed by the code outside this shield.

- Technically in encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared.
- As in encapsulation, the data in a class is hidden from other classes, so it is also known as data-hiding.

12.8. What is Inheritance?

Inheritance is an important pillar of OOP (Object Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features (fields and methods) of another class.

12.9. What is Polymorphism?

It refers to the ability of OOPs programming languages to differentiate between entities with the same name efficiently. This is done by Java with the help of the signature and declaration of these entities.

Polymorphism in Java are mainly of 2 types

- Overloading
- Overriding

Method Overloading is a **Compile time polymorphism**. In method overloading, more than one method shares the same method name with a different signature in the class. In method overloading, the return type can or cannot be the same, but we have to change the parameter because, in java, we cannot achieve the method overloading by changing only the return type of the method.

Method Overriding is a **Run time polymorphism**. In method overriding, the derived class provides the specific implementation of the method that is already provided by the base class or parent class. In method overriding, the return type must be the same or co-variant (return type may vary in the same direction as the derived class).

Chapter 13 Selenium WebDriver (Automation)

13.1. What is Automation Testing?

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves the use of a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn't require any manual intervention.

13.2. What are the benefits of Automation Testing?

Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human-generated errors
6. Saves time and money

13.3. Why should Selenium be selected as a test tool?

Selenium

1. is a free and open source
2. have a large user base and helping communities
3. have cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)
4. have great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing

13.4. What is Selenium? What are the different Selenium components?

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web-based applications and a wide range of browsers and platforms. Due to its existence in the open-source community, it has become one of the most accepted tools amongst the testing professionals.

Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason, it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes the following sets of tools:

- **Selenium Integrated Development Environment (IDE)** – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.

- **Selenium Remote Control (RC)** – Selenium RC is a server that allows a user to create test scripts in the desired programming language. It also allows executing test scripts within the large spectrum of browsers.
- **Selenium WebDriver** – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
- **Selenium Grid** – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

13.5. What are the testing types that can be supported by Selenium?

Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

13.6. What are the limitations of Selenium?

Following are the limitations of Selenium:

- Selenium supports testing of only web-based applications
- Mobile applications cannot be tested using Selenium
- Captcha and Barcode readers cannot be tested using Selenium
- Reports can only be generated using third-party tools like TestNG or JUnit.
- As Selenium is a free tool, thus there is no ready vendor support through the user can find numerous helping communities.
- The user is expected to possess prior programming language knowledge.

13.7. When should I use Selenium IDE?

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

13.8. What is Selenese?

Selenese is the language which is used to write test scripts in Selenium IDE.

13.9. What are the different types of locators in Selenium?

The locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium:

- ID
- ClassName
- Name
- TagName
- LinkText
- PartialLinkText

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

- Xpath
- CSS Selector
- DOM

13.10. What is the difference between assert and verify commands?

Assert: Assert command checks whether the given condition is true or false. Let's say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.

Verify: Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn't halt i.e. any failure during verification would not stop the execution and all the test steps would be executed.

13.11. What is an XPath?

XPath is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags. Both being markup languages and since they fall under the same umbrella, XPath can be used to locate HTML elements.

The fundamental behind locating elements using XPath is the traversing between various elements across the entire page and thus enabling a user to find an element with the reference of another element.

13.12. What is the difference between Static Xpath and Dynamic Xpath?

An absolute xpath in HTML Document Object Modal (DOM) starts with html e.g.

```
1. html/body/div[5]/div[2]/div/div[2]/div[2]/h2[1]
```

and a relative xpath finds the closed id to the DOM element and generates xpath starting from that element e.g.

```
1. //*[@id='answers']/h2[1]/a[1]
```

Single Slash “/” – Single slash is used to create Xpath with absolute path.

Double Slash “//” – Double slash is used to create Xpath with relative path.

13.13. When should I use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

13.14. How do I launch the browser using WebDriver?

The following syntax can be used to launch Browser:

```
WebDriver driver = new FirefoxDriver();  
WebDriver driver = new ChromeDriver();  
WebDriver driver = new InternetExplorerDriver();
```

13.15. What are the different types of Drivers available in WebDriver?

The different drivers available in WebDriver are:

- FirefoxDriver
- InternetExplorerDriver
- ChromeDriver
- SafariDriver
- OperaDriver
- AndroidDriver
- IPHONEDriver
- HtmlUnitDriver

13.16. What are the different types of waits available in WebDriver?

There are two types of waits available in WebDriver:

1. Implicit Wait
2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, the subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

There are some common conditions that are frequently of use when automating web browsers.

- title_is
- title_contains
- presence_of_element_located
- visibility_of_element_located
- visibility_of
- presence_of_all_elements_located
- text_to_be_present_in_element
- text_to_be_present_in_element_value

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

- `frame_to_be_available_and_switch_to_it`
- `invisibility_of_element_located`
- `element_to_be_clickable`
- `staleness_of`
- `element_to_be_selected`
- `element_located_to_be_selected`
- `element_selection_state_to_be`
- `element_located_selection_state_to_be`
- `alert_is_present`

13.17. How to type in a textbox using Selenium?

The user can use `sendKeys("String to be entered")` to enter the string in the textbox.

Syntax:

```
WebElement username = drv.findElement(By.id("Email"));
// entering username
username.sendKeys("sth");
```

13.18. How can you find if an element is displayed on the screen?

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. `isDisplayed()`
2. `isSelected()`
3. `isEnabled()`

Syntax:

isDisplayed():

```
boolean buttonPresence = driver.findElement(By.id("gbqfba")).isDisplayed();
```

isSelected():

```
boolean buttonSelected = driver.findElement(By.id("gbqfba")).isSelected();
```

isEnabled():

```
boolean searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled();
```

13.19. How can we get a text of a web element?

Get command is used to retrieve the inner text of the specified web element. The command doesn't require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

Syntax:

```
String Text = driver.findElement(By.id("Text")).getText();
```

13.20. How to select value in a dropdown?

The value in the dropdown can be selected using WebDriver's Select class.

Syntax:**selectByValue:**

```
Select selectByValue = new Select(driver.findElement(By.id("SelectID_One")));  
selectByValue.selectByValue("greenvalue");
```

selectByVisibleText:

```
Select selectByVisibleText = new Select  
(driver.findElement(By.id("SelectID_Two")));  
selectByVisibleText.selectByVisibleText("Lime");
```

selectByIndex:

```
Select selectByIndex = new Select(driver.findElement(By.id("SelectID_Three")));  
selectByIndex.selectByIndex(2);
```

13.21. What are the different types of navigation commands?

Following are the navigation commands:

navigate().back() – The above command requires no parameters and takes back the user to the previous webpage in the web browser's history.

Sample code:

```
driver.navigate().back();
```

navigate().forward() – This command lets the user to navigate to the next web page with reference to the browser's history.

Sample code:

```
driver.navigate().forward();
```

navigate().refresh() – This command lets the user to refresh the current web page there by reloading all the web elements.

Sample code:

```
driver.navigate().refresh();
```

navigate().to() – This command lets the user to launch a new web browser window and navigate to the specified URL.

Sample code:

```
driver.navigate().to("https://google.com");
```

13.22. How to click on a hyper link using linkText?

```
driver.findElement(By.linkText("Google")).click();
```

The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.

The above-mentioned link can also be accessed by using the following command.

```
driver.findElement(By.partialLinkText("Goo")).click();
```

The above command finds the element based on the substring of the link provided in the parenthesis and thus `partialLinkText()` finds the web element with the specified substring and then clicks on it.

13.23. How to handle frame in WebDriver?

An inline frame acronym as `iframe` is used to insert another document within the current HTML document or simply a web page into a web page by enabling nesting.

Select iframe by id

```
driver.switchTo().frame("ID of the frame");
```

Locating iframe using tagName

```
driver.switchTo().frame(driver.findElements(By.tagName("iframe")).get(0));
```

Locating iframe using index

frame(index)

```
driver.switchTo().frame(0);
```

frame(Name of Frame)

```
driver.switchTo().frame("name of the frame");
```

frame(WebElement element)

Select Parent Window

```
driver.switchTo().defaultContent();
```

13.24. When do we use `findElement()` and `findElements()`?

findElement(): `findElement()` is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

Syntax:

```
WebElement element = driver.findElement(By.xpath("//div[@id='example']/ul/li"));
```

findElements(): `findElements()` is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of `WebElements`.

Syntax:

```
List <WebElement> elementList  
= driver.findElements(By.xpath("//div[@id='example']/ul/li"));
```

13.25. What is the difference between `driver.close()` and `driver.quit` command?

They do a similar thing but not exactly the same.

close(): `WebDriver`'s `close()` method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the `WebDriver`.

quit(): `driver.quit()` is used to exit the browser, end the session, tabs, pop-ups etc.

13.26. Can Selenium handle windows-based pop up?

Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

13.27. How can we handle web-based pop-up?

WebDriver offers the users a very efficient way to handle these pop-ups using Alert interface. There are the four methods that we would be using along with the Alert interface.

- void dismiss() – The dismiss() method clicks on the “Cancel” button as soon as the pop-up window appears.
- void accept() – The accept() method clicks on the “Ok” button as soon as the pop-up window appears.
- String getText() – The getText() method returns the text displayed on the alert box.
- void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

Syntax:

```
// accepting javascript alert  
Alert alert = driver.switchTo().alert();  
alert.accept();
```

13.28. How to assert the title of the web page?

```
//verify the title of the web page  
assertTrue("The title of the window is incorrect.",driver.getTitle().equals("Title of the page"));
```

13.29. How to mouse hover on a web element using WebDriver?

WebDriver offers a wide range of interaction utilities that the user can exploit to automate mouse and keyboard events. Action Interface is one such utility which simulates the single user interactions.

Thus, In the following scenario, we have used Action Interface to mouse hover on a drop down which then opens a list of options.

Sample Code:

```
// Instantiating Action Interface  
  
Actions actions=new Actions(driver);  
  
// howering on the dropdown  
  
actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform();  
  
// Clicking on one of the items in the list options
```

```
WebElement subLinkOption=driver.findElement(By.id("id of the sub link"));  
subLinkOption.click();
```

13.30. How to retrieve CSS properties of an element?

The values of the css properties can be retrieved using a get() method:

Syntax:

```
driver.findElement(By.id("id")).getCssValue("name of css attribute");  
driver.findElement(By.id("id")).getCssValue("font-size");
```

13.31. What is Junit?

Junit is a unit testing framework introduced by Apache. Junit is based on Java.

13.32. What are Junit annotations?

Following are the JUnit Annotations:

- **@Test:** Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
- **@Before:** Method annotated as @Before lets the system know that this method shall be executed every time before each of the test methods.
- **@After:** Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
- **@BeforeClass:** Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test methods.
- **@AfterClass:** Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test methods.
- **@Ignore:** Method annotated as @Ignore lets the system know that this method shall not be executed.

13.33. What is TestNG and how is it better than Junit?

TestNG is an advanced framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, JUnit gained enormous popularity across the Java applications, Java developers and Java testers with remarkably increasing the code quality. Despite being easy to use and straightforward, JUnit has its own limitations which give rise to the need of bringing TestNG into the picture. TestNG is an open source framework which is distributed under the Apache Software License and is readily available for download.

TestNG with WebDriver provides an efficient and effective test result format that can, in turn, be shared with the stakeholders to have a glimpse on the product's/application's health thereby eliminating the drawback of WebDriver's incapability to generate test reports. TestNG has an inbuilt exception handling mechanism which lets the program to run without terminating unexpectedly.

There are various advantages that make TestNG superior to JUnit. Some of them are:

- Added advance and easy annotations
- Execution patterns can set
- Concurrent execution of test scripts
- Test case dependencies can be set

13.34. What is a framework?

The framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

13.35. What are the advantages of the Automation framework?

The advantage of Test Automation framework

- Reusability of code
- Maximum coverage
- Recovery scenario
- Low-cost maintenance
- Minimal manual intervention
- Easy Reporting

13.36. What are the different types of frameworks?

Below are the different types of frameworks:

1. **Module Based Testing Framework:**
The framework divides the entire “Application Under Test” into the number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts have taken together builds a larger test script representing more than one module.
2. **Library Architecture Testing Framework:**
The basic fundamental behind the framework is to determine the common steps and group them into functions under a library and call those functions in the test scripts whenever required.
3. **Data Driven Testing Framework:**
Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in “Key-Value” pairs. Thus, the key can be used to access and populate the data within the test scripts.
4. **Keyword Driven Testing Framework:**
The Keyword Driven testing framework is an extension to Data-driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.
5. **Hybrid Testing Framework:**
Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.

6. Test Driven Development (TDD) Imp

TDD stands for Test Driven Development. In this software development technique, we create the test cases first and then write the code underlying those test cases.

7. Behavior Driven Development (BDD) imp

Behavior Driven Development framework allows automation of functional validations in an easily readable and understandable format to Business Analysts, Developers, Testers, etc.

Example: Write the behavior of the application for entering the username and password.

Scenario: Login check

Given I am on the login page

When I enter "username" username

And I enter "Password" password

And I click on the "Login" button

Then I am able to login successfully.

8. Page Object Model (POM) imp

Page Object Model (POM) is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance.

Advantages of POM

1. Page Object Design Pattern says operations and flows in the UI should be separated from verification. This concept makes our code cleaner and easy to understand.
2. The Second benefit is the object repository is independent of test cases, so we can use the same object repository for a different purpose with different tools.
3. Code becomes less and optimized because of the reusable page methods in the POM classes.
4. Methods get more realistic names which can be easily mapped with the operation happening in UI. i.e. if after clicking on the button we land on the home page, the method name will be like 'gotoHomePage()'.

13.37. How can I read test data from excels?

Test data can efficiently be read from excel using JXL or POI API.

13.38. What is the difference between Selenium and QTP?

Feature	Selenium	Quick Test Professional (QTP)
Browser Compatibility	Selenium supports almost all the popular browsers like Firefox, Chrome, Safari, Internet Explorer, Opera etc.	QTP supports Internet Explorer, Firefox and Chrome. QTP only supports Windows Operating System
Distribution	Selenium is distributed as an open-source tool and is freely available	QTP is distributed as a licensed tool and is commercialized
Application under Test	Selenium supports testing of only web-based applications	QTP supports testing of both the web-based application and windows based application
Object Repository	Object Repository needs to be created as a separate entity	QTP automatically creates and maintains Object Repository
Language Support	Selenium supports multiple programming languages like Java, C#, Ruby, Python, Perl etc	QTP supports only VB Script
Vendor Support	As Selenium is a free tool, user would not get the vendor's support in troubleshooting issues	Users can easily get the vendor's support in case of any issue

13.39. Can WebDriver test Mobile applications?

WebDriver cannot test Mobile applications. WebDriver is a web-based testing tool, therefore applications on the mobile browsers can be tested.

13.40. Can captcha be automated?

No, captcha and barcode reader cannot be automated.

13.41. What is Object Repository? How can we create an Object Repository in Selenium?

Object Repository is a term used to refer to the collection of web elements belonging to Application Under Test (AUT) along with their locator values. Thus, whenever the element is required within the script, the locator value can be populated from the Object Repository. Object Repository is used to store locators in a centralized location instead of hardcoding them within the scripts.

In Selenium, objects can be stored in an excel sheet which can be populated inside the script whenever required.

13.42. What is an Exception?

Exceptions are events due to which java program ends abruptly without giving expected output. Java provides a framework where a user can handle exceptions.

13.43. What is exception handling?

When you start working with Selenium webdriver, you will come across different exceptions based on the code you write. When an exception occurs, the normal flow of program halts and an exception object is created. Exception Handling is a mechanism to handle runtime errors such as `ClassNotFoundException`, `IOException`, `SQLException`, `RemoteException`, etc.

Though there are many Exception classes under `WebDriverException`, we commonly see the below ones.

- *NoSuchElementException*
- *NoSuchWindowException*
- *NoSuchFrameException*
- *NoAlertPresentException*
- *InvalidSelectorException*
- *ElementNotVisibleException*
- *ElementNotSelectableException*
- *TimeoutException*
- *NoSuchSessionException*
- *StaleElementReferenceException*

13.44. How to Switch Tabs in window using selenium?

```
p = driver.current_window_handle
parent = driver.window_handles[0]
child = driver.window_handles[1]
driver.switch_to.window(child)
```

13.45. What is SELF in python?

__init__ method

"**__init__**" is a reserved method in **python** classes. It is called as a constructor in object oriented terminology. This method is called when an object is created from a class and **it** allows the class to **initialize** the attributes of the class.

Hope in this article you will find answers to most frequently asked Selenium and WebDriver Interview questions.

Chapter 14 JMeter Concepts (Automation)

14.1. Explain the architecture of JMeter.

JMeter is a Java-based open-source application that is basically designed for the purpose of Load Testing. It supports all major protocols that are supported in Load Runner. Unlike any browser, JMeter works on levels of protocols and does not execute JavaScript present in HTML web pages.

14.2. Does JMeter simulate actual browser behavior?

No, JMeter does not support the actual browser behavior. It does not render the HTML webpages as the normal browser does. The response can be viewed in HTML format but the actual timings are not present in the generated samples.

14.3. What is Distributed testing?

Distributed Testing means using multiple machines for load testing in which one of the machines can be made master and others can be kept as a slave. It is very important to note that all the machines should be on the same network and should have the same version of Java and JMeter.

14.4. What is the use of Regular Expression in JMeter?

Regular Expression is used for extracting some values dynamically from the responses. These values can be used in the subsequent request or can be saved for reporting purposes. Regular Expression is used in both Pre-Processors as well as Post Processors.

14.5. What are the types of processors in JMeter?

Basically, there are two types of processors in JMeter namely Pre-Processor and Post Processor.

Pre-Processors execute before the main sampler and can change the scope of the sampler whereas **Post Processors** execute after the main sampler and are applicable to all samplers in the same scope of Test Plan. They can be used to extract some fields from the server response and store them in variables.

14.6. What are the different ways of Data Parameterization in JMeter?

Data Parametrization makes the scripts reusable where the values is not required to be hardcoded for the same request with different parameters.

Below is the data parametrization that is supported in JMeter:

- CSV Data Set Config
- User-Defined Variables.

14.7. What are the maximum recommended threads on a single system?

It depends on the hardware configuration of your system which includes a processor, JVM, allocated memory -Xmx, etc.

Other factors that impact thread count are the number of components in your test plan i.e., the number of config elements or processors and it also depends on whether you are using GUI/Non-GUI Mode.

14.8. Explain the difference between Gaussian and Poisson Timers.

Both Gaussian and Poisson Timers work on a mathematical formula with some constant delay and additional offset. Difference between the two lies in the fact that how the lambda value is calculated in the case of Poisson timer and how deviation is calculated in the case of Gaussian Timer.

14.9. What are the major differences between JMeter and Load Runner?

JMeter is considered as the major competitor of Load Runner in the industry. **Enlisted are some of the major differences:**

Load Runner	JMeter
Licensed Software	Open Source tool.
Developed by Mercury	Developed by Apache.
UI is very impressive	It lacks in UI
It has more technical capabilities.	Less technically sound as compared to Load Runner.
Supports SAP, Siebel and Peoplesoft.	Doesn't support SAP and Siebel

14.10. What is the use of co-relation in JMeter?

Co-relation is a process of extracting the values from the server response and storing it in a variable to be used in any other request which is to follow.

For Example, for testing any login functionality if you have to use the session ID/cookie ID, you can extract the values from the response of GET Request of the login page and then dynamically use the same while making POST request for a login.

14.11. What are the different types of listeners?

Listeners are used for storing the execution results of load testing in different forms be it in a table, graph, tree or in any other presentable format so that it can be presented to the client. There are different type of inbuild listeners in JMeter and many others can be imported into it by using plugins as per the requirement.

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

Some of the inbuild listeners are:

- View results in Table
- View results in Tree
- Graph results
- Aggregate graph
- Aggregate report
- Assertion results
- Response time graph

14.12. Explain the flow of the Test Script Recorder.

HTTP(s) Test Script Recorder is used to record all the Http(s) requests going to the server from your application. Some configurations require to be done in JMeter in order to make it work.

Steps followed to record https traffic are:

- Add HTTP(s) Test script recorder to WorkBench.
- Enter the port number to start your proxy server.
- Choose the target either as "Workbench" or add a Recording Controller in your test plan and select the same target for storing all the recordings under it.
- Start the proxy server.
- Configure your browser with manual proxy settings pointing to the same port number used in the test script recorder.

14.13. Can JMeter record actions from mobile? If yes, how?

Yes, JMeter can record HTTP or Https request going to the server from your mobile application also. It is required that mobile and JMeter are on the same network.

Below is the configuration required:

- Configure your proxy server in JMeter to run at a specified port.
- Set up the proxy on your mobile wifi settings and enter the same port number that is used in the recorder.
- Install the Root CA certificate on your mobile.
- Hit server requests from your mobile and observe it getting captured by the specified controller.

14.14. How to do master-slave configuration in JMeter?

Master-slave configuration is a part of distributed testing in which more than one machine is used to perform load testing of the server under test.

It is very important that all machines are on the same network and all have the same version of JMeter. In distributed testing, one machine considered as the master and the others are kept as slaves by doing some configurations.

The process is specified below:

- On the master machine, edit the JMeter. Properties file and add the IP addresses of slave machines against the remote host field in the file.
- Save the file and open the JMeter again.

- Now, from the RUN menu in JMeter, select Remote Start and choose the IP of the machine to be invoked.
- Choose RUN menu and select Remote Start all to start all the slave machines for your testing.

14.15. What are the JMeter supported protocols?

JMeter **supports various standard protocols like:**

- HTTP/HTTPS
- SOAP
- LDAP
- FTP
- SMTP
- TCP

14.16. Explain the syntax of JMeter variables and functions.

Just as in any other programming language, variables and functions are used in JMeter also in order to make the scripts reusable.

Syntax of Variable – \${var}

There are many inbuilt functions that are available in JMeter to perform various actions. Function string can be generated from the Function Dialogue Box itself.

For Example, if you want to get the machine IP stored in a machineIP variable, you can use the string \${__machineIP(machineIP)}.

14.17. Why is it recommended to run JMeter in GUI mode?

JMeter tests can be run both GUI as well as Non-GUI Mode. It is highly recommended to run the load test in non-GUI mode because the AWT event thread can kill the tests in case of high load scenarios.

The various non-GUI mode supported with JMeter are:

- Command-line
- ANT plugin
- MAVEN plugin
- Jenkins

14.18. Is it possible to run selenium scripts in JMeter? If yes, how?

Yes, it is possible to run selenium scripts in JMeter to get some ideas on their performance.

There are two ways of doing it. Either you can use JUnit libraries to build selenium scripts and save as Jars and copy the same in the JMeter directory. And then add JUnit sampler to your test plan and import the Jar file.

Otherwise, the Webdriver sampler plugin can be added in the JMeter ext folder. Restart the JMeter. Write your selenium code in the Webdriver sampler and then execute it to see the performance.

Prepared by: **Mr. Umar Arshad | Software Quality Assurance Engineer**

14.19. How do you manage sessions and cookies in JMeter?

Sessions and cookies can be managed in JMeter by using config elements such as HTTP Cache Manager which provides an option to clear the cookies in every iteration and also allows to add user-defined cookies.

HTTP Cache manager helps you in clearing cache after each iteration as per your requirement in the load tests and also limits the number of elements that can be stored in the cache. Both of these config elements can be attached to the HTTP sampler.

14.20. What are the important steps for testing JDBC request?

JDBC requests are used to establish a connection with the databases and then measure the response time of the queries.

Important steps for testing JDBC requests are:

- **Setting up Config Element**, JDBC Connection configuration in which Database URL and JDBC Driver Class needs to be added as per the database which is being used. Also, add the variable name for this connection configuration so as to use it in the sampler.
- Add JDBC Request. Add the same variable name added above and write your queries to the test.

14.21. What is BeanShell scripting?

BeanShell is a lightweight Java scripting that is used in JMeter to perform some complex task. BeanShell sampler can perform various functions with the use of coding. You can print the thread number, get the current sampler executed, fetch the cookies, etc.

14.22. Can JMeter measure the performance of a complete application? For Example, you have multiple screens in your mobile app. Can JMeter measure the time taken to flip the screens?

No, JMeter does not measure the transition time between the screens. It can only measure the server actions not the UI interactions.

14.23. What is a Root CA certificate?

HTTPS connection requires a certificate to authenticate the connections which get established when the browser hits the webserver. JMeter generates it temporarily to intercept the SSL traffic in order to record the actions. For recording actions via mobile, you need to have this certificate on your mobile to record the actions.

14.24. Which factors decide the maximum threads that one should generate per system?

It depends on the hardware of the system.

For Example, on a 2-3 GHz CPU, 400-600 threads can be generated. It also depends on the components in your test plan. More the processors and XML parsing elements, the more the CPU load and hence fewer threads. For high load, it is recommended to use multiple machines for load testing.

14.25. What is Tidy Parsing?

Tidy Parsing is a type of parsing that is used in Xpath extractor. If the response is in pure XML then tidy parsing is not required whereas, in the case of XHTML, it is mandatory to check the tidy parsing option in order to fetch the correct results.

14.26. What are the important plugins that are supported in JMeter?

JMeter supports different types of plugins which are helpful in generating high-quality results.

Below are the major plugins that are supported:

- Thread group plugin – Stepping thread group plugin.
- Samplers' plugins like Webdriver.
- Listener's plugins.

14.27. What are the types of the controller in JMeter?

Controllers are used in JMeter to control the flow of execution of requests.

Below are the controllers that are used in JMeter:

- Recording controller
- IF controller
- While controller
- Transaction controller
- Loop controller
- Simple controller
- Module controller

14.28. What are the samplers?

How does a Jmeter knows what type of request has been sent to server???

It is through Samplers. Samplers are a must to add to a Test Plan as only it can let Jmeter know what type of request need to go to which server and with any predefined parameters or not. Requests could be HTTP, HTTP(s), FTP, TCP, SMTP, SOAP, JDBC etc.

14.29. What are the Assertions.

Till now, we have covered how JMeter hits the server and how the responses are displayed via listeners. To ensure that the response received is correct and as per expectation, we need to add assertions. Assertions are simply validations which we need to put on responses to compare the results.

Below are the types of assertions commonly used:

- Response Assertion

- Duration Assertion
- Size Assertion
- XML Assertion
- HTML Assertion

I'm sure that this article on JMeter interview questions would have helped to enrich your knowledge on JMeter concepts to a great extent.

A thorough understanding of all the questions covered here would help you to crack any interview confidently.

We wish you all the success!!

Chapter 15 Aptitude Questions

15.1. If you have a 5-litre jug and a 3-litre jug, how would you measure exactly 4 liters?

Answer

1. Fill the 3L jug fully and empty it in the 5L jug
2. Now again fill 3L fully and pour its water in the 5L jug (remember it already has 3 liters of water in it) until it full.
3. Now you're left with only 1 liter of water in the 3L jug (*hope you are getting it)
4. Now empty 5L jug and pour that remaining 1-liter water that we got in step 3 into the 5L jug.
- a. Now 5L jug contains 1 liter of water.
5. Now fill the 3L jug fully and empty it into the 5L jug so you will now have 4 liters of water in 5L jug.

That's how you get 4litres of water.

15.2. I give you 4 tablets which contain 2 for fever and 2 for cold, all the same size, shape, taste, weight and color, no cover. You have to take 1 cold and 1 fever tablet right now. How will you choose correctly?

Answer

So basically, you have 4 tablets. 2 for cold and other 2 for fever. Now start dividing the tablets into half. Keep one half in your hand and other half on table. Once you are done dividing all the tablets into half, take either of them. Problem solved.

15.3. There is a room with a door (closed) and three light bulbs. Outside the room, there are three switches, connected to the bulbs. You may manipulate the switches as you wish, but once you open the door you can't change them. Identify each switch with its bulb. All bulbs are in working condition.

Answer

Let the bulbs be X, Y, and Z

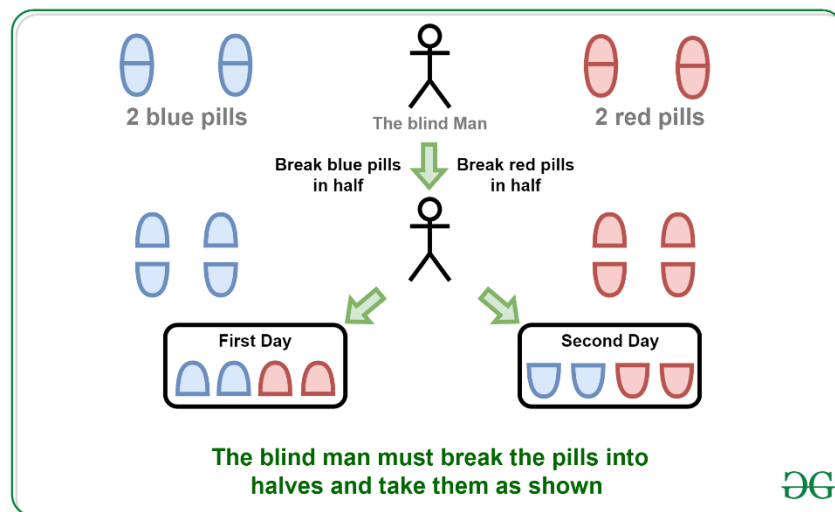
Turn on switch X for 5 to 10 minutes. Turn it off and turn on switch Y. Open the door and touch the light bulb.

1. if the light is on, it is Y
2. if the light is off and hot, it is X
3. if the light is off and cold, it is Z

15.4. A blind man is alone on a deserted island. He has two blue pills and two red pills. He must take exactly one red pill and one blue pill or he will die.

Answer

Firstly, break each of the pills in half, and as you do this pop one half in your mouth and lay the other half aside for tomorrow. When he's done this with all four pills he will have consumed one red pill and one blue pill. And have the same leftover.



15.5. A newspaper made of 16 large sheets of paper folded in half. The newspaper has 64 pages altogether. The first sheet contains pages 1, 2, 63, 64.

If we pick up a sheet containing page number 45. What are the other pages that this sheet contains?

Answer

On the back of 45, it is 46. The pages are such that for each page p , $65-p$ will be also on the same page.

Then,

$$65-45 = 20$$

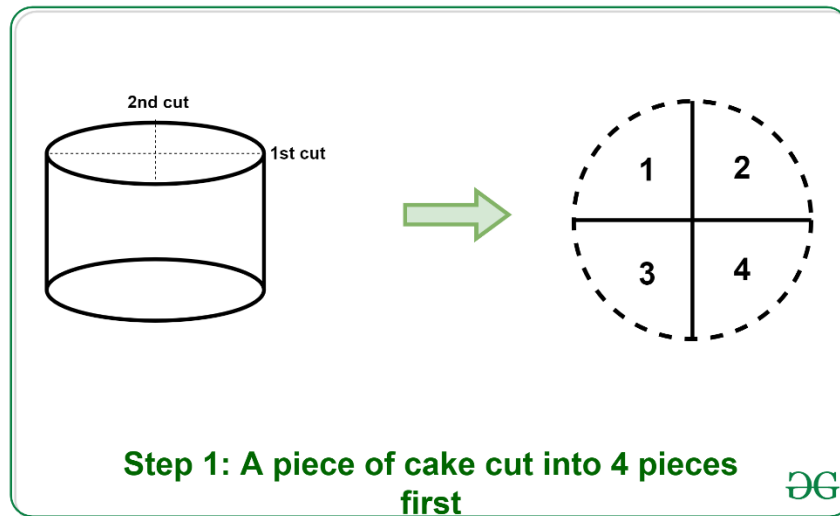
$$65-46 = 19$$

So, the four pages in this sheet are 19, 20, 45, 46.

15.6. You have a birthday cake and have to cut it into 8 equal pieces by making 3 cuts only. How do you do it?

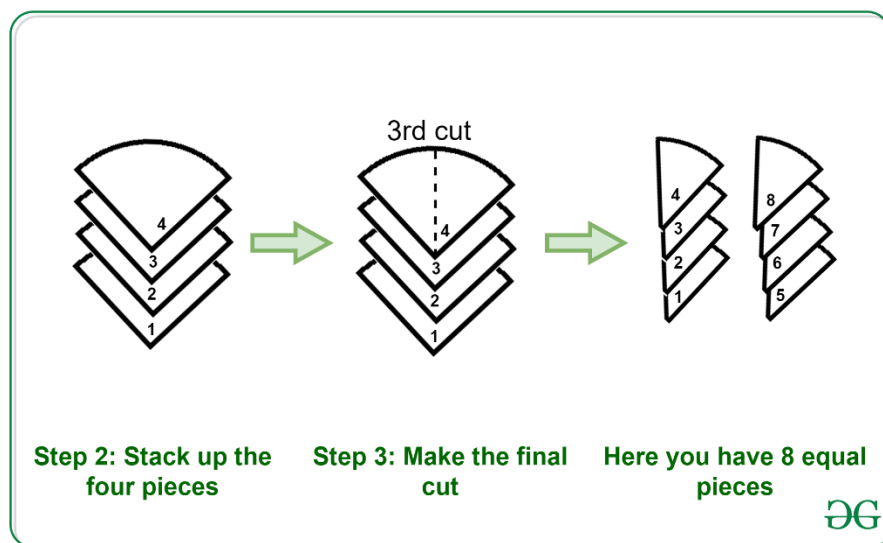
Answer

Step 1: Cut the cake into quarters (4 pieces) using 2 of the cuts – one horizontally down the center of the cake and the other vertically down the center of the cake. This will leave you with 4 pieces (or slices) of cake.



Step 2: Then take all 4 pieces and arrange them in a stack that is 4 pieces high.

Step 3: Finally, you can just cut that stack of 4 pieces in half – using your third and final cut – and then you will end up with 8 pieces of cake!



15.7. Which letter replaces the “?”:

L : O
R : I
E : V
M : ?

Solution : N, because if we assign the values as A=1, B=2.....Z=26, then each pair adds upto 27.

example : L O

$$12+15 = 27.$$

And for M:N

$$13+14=27$$

15.8. A Lady (L) bought an item of Rs 100 from the Shopkeeper (C). She paid him through a 500 Rs Note. Realizing that he did not have change, the shopkeeper C got change for that note from another shopkeeper (S) and paid Rs 400 to the Lady.

After a few days, S realized that the note is fake, And this railed at C and took 500 Rs back from him.

So in this whole process how much money did C loose in the end?

Answer: 500

15.9. What is half of two plus two?

A) Half of $2+2 = (\text{Half of } 2) + 2 = (1) + 2 = 3$.

B) Half of $2+2 = \text{Half of } (2+2) = (.5)(4) = 2$.

This is based on the user perspective for the perfect solution based on the situation where in it is used.

But, technically to answer this, the correct answer is 3.

15.10. Hand shake puzzle formula to calculate no of handshakes.

The formula for the number of handshakes possible at a party with n people is

$$\# \text{ handshakes} = n*(n - 1)/2.$$

This is because each of the n people can shake hands with n - 1 people (they would not shake their own hand), and the handshake between two people is not counted twice.

This formula can be used for any number of people. For example, with a party of 5 people, find the number of handshakes possible.

$$\# \text{ handshakes} = 5 \times (5 - 1) / 2.$$

$$\# \text{ handshakes} = 5 \times (4) / 2.$$

$$\# \text{ handshakes} = 20 / 2.$$

$$\# \text{ handshakes} = 10$$

So, there are 10 handshakes that can be made between 5 people.

15.11. You have 9 balls, equally big, equally heavy - except for one, which is a little heavier.

How would you identify the heavier ball if you could use a pair of balance scales only twice?

ANS:

Divide the 9 balls into 3 groups of 3. Compare the weight of two of those groups.

The heavier group should then be obvious, it will either tip the scales, or, if the scales stay balanced, then it is the group you didn't include.

Now, choose 2 balls from this group and compare their weights, and using the same logic as before, the heavier ball will be obvious.

Thanks