Q1 - What Is The Difference Between Quality Assurance, Quality Control??

Quality Assurance (QA):

1. QA is planned and systematic way to evaluate quality of process used to produce a quality product.
2. QA deals with how to prevent bugs from occurring in a product being developed.
3. Quality Assurance is process oriented
4. QA comes under the category of Verification.

Quality Control (QC):

1. QC in software industry is evaluating software product, find the defects & suggest improvements.
2. QC implements the process established by QA.
3. The goal of Quality Control is to identify the defects in the software application after it is developed.
4. Quality Control is product oriented
5. QC comes under the category of Validation.

Q2 - What is Testware?

The Testware is:

* The subset of software which helps in performing the testing of application.
* Testware is required to plan, design, and execute tests. It contains documents, scripts, inputs, expected results, set-up and additional software or utilities used in testing.
* Testware is the term given to combination of all utilities and application software that required for testing a software package.

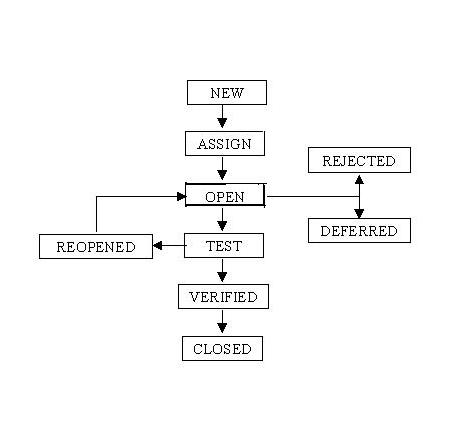
Q3 - What is the difference between build and release and version?

1. Build is a Executable file which is handed over to the tester to test the functionality of the developed part of the project.
2. Release is that which we finally hand it over to the client of the project after the development and testing phases are completed.
3. Version is the number of releases made according to the addition of the requirement of the client.

Q4 - What is bug leakage and bug release?

1. Bug Release:- When we released any version of an application or software with a group of some known bugs or issues. These issues or bugs are generally low severity and low priority bugs. These released issues or bugs are also described in the Release Notes.
2. Bug Leakage:- After the release of the application to the client, if the end user finds any type of defects while using that application, then it signifies the presence of defect leakage. In general, defect leakage highlights the defects that are found by the end users, after the release of the application and it is also referred to as bug leakage.

Q5 - Explain the steps for Bug Cycle



1. New: When a defect is logged and posted for the first time. It’s state is given as new.
2. Assigned: After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It’s state given as assigned.
3. Open:  At this state the developer has started analyzing and working on the defect fix.
4. Fixed:  When developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’ and the bug is passed to testing team.
5. Pending retest:  After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.
6. [Retest](http://istqbexamcertification.com/what-is-retesting/):  At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.
7. Verified:  The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “verified”.
8. Reopen:  If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.
9. Closed:  Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.
10. Duplicate: If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “duplicate“.
11. Rejected: If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “rejected”.
12. Deferred: The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are [priority](http://istqbexamcertification.com/what-is-the-difference-between-severity-and-priority/) of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
13. Not a bug:  The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and field of the application like change of color of some text then it is not a bug but just some change in the looks of the  application.

Q6 - What is Test plan?

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project.

1. Test Plan Identifier - Provide a unique identifier for the document. (Adhere to the Configuration Management System if you have one.)
2. Introduction - A brief introduction about the project and to the document. Here we Specify the goals/objectives.
3. Test items - A test item is a software item that is the application under test.
4. Features to be tested - List the features of the software/product to be tested.
5. Features not to be tested - Identify the features and the reasons for not including as part of testing.
6. **Approach** - Specify the testing levels [if it’s a Master Test Plan], the testing types, and the testing methods [Manual/Automated; White Box/Black Box/Gray Box]
7. Item pass/fail criteria - Specify the criteria that will be used to determine whether each test item (software/product) has passed or failed testing.
8. Suspension Criteria and Resumption Requirements - Specify criteria to be used to suspend the testing activity. Specify testing activities which must be redone when testing is resumed.
9. Test deliverables - The deliverables that are delivered as part of the testing process, such as test plans, test specifications and test summary reports.
10. **Environmental needs** - Defining the environmental requirements such as hardware, software, OS, network configurations, tools required.
11. **Responsibilities** - Lists the roles and responsibilities of the team members
12. Staffing and training needs - Captures the actual staffing requirements and any specific skills and training requirements.
13. **Schedule** - States the important project delivery dates and key milestones.
14. Risks and Mitigation - High-level project risks and assumptions and a mitigation plan for each identified risk.
15. Approvals - Captures all approvers of the document, their titles and the sign off date.

Q7 - what is Requirement Traceability Matrix?

A requirements traceability matrix is a **document that traces and maps user requirements** [requirement Ids from requirement specification document] with the test case ids. Purpose is to make sure that all the requirements are covered in test cases so that while testing no functionality can be missed. This document is prepared to make the clients satisfy that the coverage done is complete as end to end, this document consists of Requirement/Base line doc Ref No., Test case/Condition, and Defects/Bug id. Using this document the person can track the Requirement based on the Defect id.

The main purpose of Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while testing.

Requirement Traceability Matrix – Parameters include

* Requirement ID
* Risks
* Requirement Type and Description
* Trace to design specification
* Unit test cases
* Integration test cases
* System test cases
* User acceptance test cases
* Trace to test script

Types of Traceability Matrix

* Forward traceability: This matrix is used to check whether the project progresses in the desired direction and for the right product. It makes sure that each requirement is applied to the product and that each requirement is tested thoroughly. It maps requirements to test cases.
* Backward or reverse traceability: It is used to ensure whether the current product remains on the right track. The purpose behind this type of traceability is to verify that we are not expanding the scope of the project by adding code, design elements, test or other work that is not specified in the requirements. It maps test cases to requirements.
* Bi-directional traceability ( Forward + Backward): This traceability metrics 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.

Q8 -What Is Sanity Testing Explain It with Example?

**Sanity Testing is the subset of Regression Testing and it is performed when we do not have enough time for doing testing.** Here only extreme functionalities are checked without finer details. It is the surface level testing(meaning that a tester checks whether whole functionality of the software works in a proper way)where QA engineer verifies that all the menus, functions, commands available in the product and project are working fine. Smoke testing is normal health check up to a build of an application before taking it to testing in depth.

Sanity Testing Example

For Example in a project there are five modules like login page, home page, user detail page, new user creation, and task creation etc. So we have the bug in login page like on login page username field accepts the less than six alpha-numeric characters which are against the requirements as in requirements it is specified that username should not be below than six characters but as username accepts the less than six characters it is the bug.

So now the bug is reported by the testing team to the developer team to fix it. When the developing team fixes the bug and passed it to testing team than the testing team checks the other modules of the application means checks that fix bug does not affect the functionality of the other modules but keep one point always in mind that testing team only checks the extreme functionality of the modules, do not go deep to test the details because of the short time so this is the sanity testing.

Sanity testing is performed after the build has clear the Smoke test and has been accepted by QA team for further testing, sanity testing checks the major functionality with finer details.

Q9: Explain Smoke Testing Example in Easy and Simple Way?

Smoke Testing is done to ensure that whether build can be acceptable enough to test further. It is done to check the stability of the build. Smoke Testing is considered as the surface level testing which is always used to validate that build provided by development to QA team is ready to accept for further testing. In Smoke Testing we test the major point’s means major functionality of the application and it is also known by the name Build Acceptance Testing (BAT).

*Smoke Testing Example – Real One*

For example we are working in a small project named Employee Management System and in this project there are four modules like New Employee Module, Existing Employee Module, Admin Module, User Module etc. So firstly in this four modules development team performs the Smoke Testing by executing all the major functionality of modules like New Employee is able to login or not and after login new employee can seen the record of the existing employee or not, and employee that is created can also be edited, deleted or not.

So in this way Smoke Testing is done by development team before releasing means submitting the build to the Software Testing team.

Now when the build is hand over means releasing to the testing team than the software testing team has to check whether to accept or reject the build by checking the major functionality of that build like employee is able to login or not and after login they can seen the existing employee record or not and after that logout easily or not. So this is the Smoke Testing done by software tester

Q10 - What is Regression Testing Explain It with Example?

When a [bug](http://testingbasicinterviewquestions.blogspot.in/2012/05/how-to-report-bug-in-small-companies.html) is fixed by the development team than testing the other features of the applications which might be affected due to the bug fix is known as regression testing.

Regression testing is always done to verify that modified code does not break the existing functionality of the application and works within the requirements of the system.

For Example there are three Modules in the Project named Admin Module, Personal Information, and Employment Module and suppose bug occurs in the Admin Module like on Admin Module existing User is not able to login with valid login credentials so this is the bug. Now Testing team sends the above - mentioned Bug to the Development team to fix it and when development team fixes the Bug and hand over to Testing team than testing team checks that fixed bug does not affect the remaining functionality of the other modules (Admin, PI, Employment) and also the functionality of the same module (Admin) so this is known as the process of regression testing done bysoftware tester.

There are mostly two strategies to regression testing, 1) to run all tests and 2) always run a subset of tests based on a test case prioritization technique.

Regression testing will be conducted after any bug fixed or any functionality changed.

Q11. What is Boundary Value Analysis and Equivalence partitioning?

Boundary value analysis and equivalence partitioning both are test case design strategies in black box testing.

Equivalence partitioning is a Test Case Design Technique to divide the input data of software into different equivalence data classes. Test cases are designed for equivalence data class. In this technique, only one condition to be tested from each partition. Because we assume that, all the conditions in one partition behave in the same manner by the software. In a partition, if one condition works other will definitely work. Likewise we assume that, if one of the condition does not work then none of the conditions in that partition will work.

Example of Equivalence Class Partitioning?

* A text field permits only numeric characters
* Length must be 6-10 characters long

Partition according to the requirement should be like this:



While evaluating Equivalence partitioning, values in all partitions are equivalent that’s why 0-5 are equivalent, 6 – 10 are equivalent and 11- 14 are equivalent. At the time of testing, test 4 and 12 as invalid values and 7 as valid one.

It is easy to test input ranges 6–10 but harder to test input ranges 2-600. Testing will be easy in the case of lesser test cases but you should be very careful. Assuming, valid input is 7. That means, you belief that the developer coded the correct valid range (6-10).

Boundary value analysis is a test case design technique to test boundary value between partitions (both valid boundary partition and invalid boundary partition). A boundary value is an input or output value on the border of an equivalence partition, includes minimum and maximum values at inside and outside boundaries. Using Boundary Value Analysis technique tester creates test cases for required input field.

Suppose you have very important tool at office, accepts valid User Name and Password field to work on that tool, and accepts minimum 8 characters and maximum 12 characters. Valid range 8-12, Invalid range 7 or less than 7 and Invalid range 13 or more than 13.

Q12. What is Integration Testing? What are the approaches?

Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

Top down Testing: In this approach testing is conducted from main module to sub module. if the sub module is not developed a temporary program called STUB is used for simulate the sub module.

Bottom up testing: In this approach testing is conducted from sub module to main module, if the main module is not developed a temporary program called DRIVERS is used to simulate the main module.

Q13. Why We Use Stubs And Drivers?

Stubs are dummy modules that are always distinguish as "called programs", or you can say that is handled in [integration testing](http://testingbasicinterviewquestions.blogspot.in/2012/01/what-is-integration-testing-explain-it.html) (top down approach), it used when sub programs are under construction. Stubs are considered as the dummy modules that always simulate the low level modules.

Drivers are also considered as the form of dummy modules which are always distinguished as "calling programs”, that is handled in [bottom up integration testing](http://testingbasicinterviewquestions.blogspot.in/2012/05/explain-bottom-up-strategy-of.html), it is only used when main programs are under construction. Drivers areas considered as the dummy modules that always simulate the high level modules.

Example of Stubs and Drivers is given below:-

For Example we have 3 modules login, home, and user module. Login module is ready and need to test it, but we call functions from home and user (which is not ready). To test at a selective module we write a short dummy piece of a code which simulates home and user, which will return values for Login, this piece of dummy code is always called Stubs and it is used in a [top down integration](http://testingbasicinterviewquestions.blogspot.in/2012/04/explain-top-down-strategy-of.html).

Considering the same Example above: If we have Home and User modules get ready and Login module is not ready, and we need to test Home and User modules Which return values from Login module, So to extract the values from Login module We write a Short Piece of Dummy code for login which returns value for home and user, So these pieces of code is always called Drivers and it is used in Bottom Up Integration

Conclusion:-

So it is fine from the above example that Stubs act “called” functions in top down integration. Drivers are “calling” Functions in bottom up integration.

Q14. Introduce Yourself?

I Am………………….. I have completed my MCA in 2012 and at that time I have joined TK20 as a trainee. Now I am having……..years of experience and my current designation is…………………

Q15. What is your role in your Organization/Team?

1. Analyzing the requirements from clients.
2. Participating in preparing Test Plan, Test Scenario and Test cases.
3. Exploratory testing and Functional Testing
4. Defect tracking, Retesting and Regression
5. Communication with TL/Manager
6. Reporting

Q16. What is Agile and Scrum?

Agile development model is an **incremental software development model**. Software is developed in incremental, rapid cycle. It is an iterative development methodology where requirement evolve through collaboration between customers and self-organized teams.

SCRUM: Scrum is a **process or subset of Agile** which is combination of iterative and incremental model. Here the product owner provides a prioritized list from the product backlog and done a story grooming meeting with the scrum team to discuss the items and move them to sprint backlogs.

Scrum Team: is a combination of people from different departments work together to achieve the tasks aligned in a particular sprint.

Sprint: is a predefined interval or time frame in which the task has to be completed. It is usually of 2 to 3 weeks.

Product Owner: is the key stakeholder or the person represents the customer side. He/She have the final authority and should be reachable if any of the scrum member has any doubt that needs some clarification.

Scrum Master: is the facilitator of the scrum team and make sure that team is productive and progressive. In case of any impediment scrum master follows up and resolve the issue for team.

User Story: is nothing but the requirement or the feature which has to be implemented.

EPIC: are the user stories that are not defined or kept for future sprints. Suppose there are some features that required to be implemented in future whose details are not yet known.

Product Backlogs: is a kind of bucket or source where all the user stories are kept. Maintained by the Product

Owner. It is a wish list of the PO who prioritized the list as per the business needs.

Sprint Backlogs: User stories taken from the product backlogs based on the priority. To complete all user stories on

which the scrum team works on a particular sprint is called sprint backlogs.

Burn Down Chart: is a graph which shows the estimate vs actual efforts of the scrum team. It is a tracking method by which day to day task is tracked.

Sprint Planning Meeting: Planning meeting is the start of Scrum. Here PO selects a prioritized list of user stories from product backlogs and team brainstorms on it. Based on the discussion team decides the complexity of the story and give story points for the same.

Daily Scrum Meeting: It is a meeting done within the sprint on daily basics. Here states 3 points:

1. What did team did yesterday
2. What did team plan to do today
3. Any Impediments

Sprint Review Meeting: It is done at the end of the Sprint to demonstrate the implemented user stories to the PO. The PO may cross verify the user stories as per the acceptance criteria.

Sprint Retrospective: It is done after the review meeting and discuss the following points:

1. What went well during the sprint
2. What did not went well
3. Lesson learnt
4. Action Items.

Q17. PILOT TESTING AND BETA TESTING

PILOT TESTING: is done by the group of users who try to test the system prior deployment to provide feedback about the quality.

BETA TESTING: is done at the client side and all end users use the system and see whether the system is working as per given requirements.

Q18. Authorization and Authentication

Authorization: is the process to test whether the specific user allowed to access some controls of features of the application.

Authentication: is the process to test whether the specific user is allowed to use or login to the application.

Q19. Difference Between Waterfall and Agile

This is not correct answer. Please ignore this question

|  |  |
| --- | --- |
| Agile | Waterfall |
| Backward scalability: It is easy to change the requirements at any stage | We can not change the requirements and decision that we have made earlier |
| Flexibility to check error under any part of development makes the application more bug free | Can be tested only after the development module |
| Agile is a iterative team based approach of development | Each phase of SDLC takes place in sequence. It is a linear approach |

Q20. Test Case and test Scenario

|  |  |
| --- | --- |
| TEST CASE is “HOW TO BE TESTED” | TEST SCENARIO is “”WHAT TO BE TESTED” |
| Consists of Test case name, preconditions, steps, input, expected etc. | It is thinking from the point of view of customer and thinks what to test. Then prepare test cases for them. |
| Test cases gives us detailed information. | It is an one line statement. |

Q:21 List of HTTP status codes?

1xx Informational : This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and is terminated by an empty line.

100 Continue: The server has received the request headers and the client should proceed to send the request body

101 Switching Protocols: The requester has asked the server to switch protocols and the server has agreed to do so.

102 Processing: A WebDAV request may contain many sub-requests involving file operations, requiring a long time to complete the request.

2xx Success: This class of status codes indicates the action requested by the client was received, understood, accepted, and processed successfully.

Bug id

Bug description

Steps to reproduce

Saverity

Priority

Status open/close/reopen/

Test case:

Test case id

Test case description

Steps to reproduce

Expected result

Actual result

200 OK: Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request, the response will contain an entity describing or containing the result of the action.

201 Created: The request has been fulfilled, resulting in the creation of a new resource.

202 Accepted: The request has been accepted for processing, but the processing has not been completed. The request might or might not be eventually acted upon, and may be disallowed when processing occurs.

3xx Redirection: This class of status code indicates the client must take additional action to complete the request. Many of these status codes are used in [URL redirection](https://en.wikipedia.org/wiki/URL_redirection).

300 Multiple Choices: Indicates multiple options for the resource from which the client may choose (via [agent-driven content negotiation](https://en.wikipedia.org/wiki/Content_negotiation#Agent-driven)). For example, this code could be used to present multiple video format options, to list files with different [extensions](https://en.wikipedia.org/wiki/File_extensions), or to suggest [word sense disambiguation](https://en.wikipedia.org/wiki/Word_sense_disambiguation).

[301 Moved Permanently:](https://en.wikipedia.org/wiki/HTTP_301) This and all future requests should be directed to the given [URI](https://en.wikipedia.org/wiki/Uniform_resource_identifier).

[302 Found](https://en.wikipedia.org/wiki/HTTP_302):This is an example of industry practice contradicting the standard. The HTTP/1.0 specification (RFC 1945) required the client to perform a temporary redirect (the original describing phrase was "Moved Temporarily"),but popular browsers implemented 302 with the functionality of a 303 See Other. Therefore, HTTP/1.1 added status codes 303 and 307 to distinguish between the two behaviours. However, some Web applications and frameworks use the 302 status code as if it were the 303.

304 Not Modified: Indicates that the resource has not been modified since the version specified by the [request headers](https://en.wikipedia.org/wiki/List_of_HTTP_header_fields#Request_Headers) If-Modified-Since or If-None-Match. In such case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.

305 Use Proxy : The requested resource is available only through a proxy, the address for which is provided in the response. Many HTTP clients (such as [Mozilla](https://en.wikipedia.org/wiki/Mozilla) and [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer)) do not correctly handle responses with this status code, primarily for security reasons.

306 Switch Proxy:No longer used. Originally meant "Subsequent requests should use the specified proxy."

307 Temporary Redirect: In this case, the request should be repeated with another URI; however, future requests should still use the original URI. In contrast to how 302 was historically implemented, the request method is not allowed to be changed when reissuing the original request. For example, a POST request should be repeated using another POST request.

308 Permanent Redirect : The request and all future requests should be repeated using another URI. 307 and 308 parallel the behaviors of 302 and 301, but *do not allow the HTTP method to change*. So, for example, submitting a form to a permanently redirected resource may continue smoothly.

4xx Client Error:  The 4xx class of status code is intended for situations in which the client seems to have erred. Except when responding to a HEAD request, the server *should* include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition.

400 Bad Request :The server cannot or will not process the request due to an apparent client error (e.g., malformed request syntax, too large size, invalid request message framing, or deceptive request routing).

401 Unauthorized : Similar to *403 Forbidden*, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. See [Basic access authentication](https://en.wikipedia.org/wiki/Basic_access_authentication) and [Digest access authentication](https://en.wikipedia.org/wiki/Digest_access_authentication). 401 semantically means ["unauthenticated"](https://en.wikipedia.org/wiki/Authentication), i.e. the user does not have the necessary credentials.

402 Payment Required: Reserved for future use. The original intention was that this code might be used as part of some form of [digital cash](https://en.wikipedia.org/wiki/Digital_cash) or [micropayment](https://en.wikipedia.org/wiki/Micropayment) scheme, but that has not happened, and this code is not usually used. [Google Developers](https://en.wikipedia.org/wiki/Google_Developers) API uses this status if a particular developer has exceeded the daily limit on requests.21.co Bitcoin micropayment service uses this status in response for every client http request.

[403 Forbidden:](https://en.wikipedia.org/wiki/HTTP_403) The request was a valid request, but the server is refusing to respond to it. The user might be logged in but does not have the necessary permissions for the resource.

[404 Not Found:](https://en.wikipedia.org/wiki/HTTP_404) The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.

405 Method Not Allowed: A request method is not supported for the requested resource; for example, a GET request on a form which requires data to be presented via [POST](https://en.wikipedia.org/wiki/POST_%28HTTP%29), or a PUT request on a read-only resource.

406 Not Acceptable: The requested resource is capable of generating only content not acceptable according to the Accept headers sent in the request.

407 Proxy Authentication Required :The client must first authenticate itself with the [proxy](https://en.wikipedia.org/wiki/Proxy_server).

408 Request Timeout: The server timed out waiting for the request. According to HTTP specifications: "The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time."

5xx Server Error: The server failed to fulfill an apparently valid request.

500 Internal Server Error: A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.

501 Not Implemented: The server either does not recognize the request method, or it lacks the ability to fulfill the request. Usually this implies future availability (e.g., a new feature of a web-service API).

502 Bad Gateway: The server was acting as a [gateway](https://en.wikipedia.org/wiki/Gateway_%28telecommunications%29) or proxy and received an invalid response from the upstream server.

503 Service Unavailable: The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary state.

504 Gateway Timeout: The server was acting as a gateway or proxy and did not receive a timely response from the upstream server.

Q22. Can you explain usability testing?

Usability testing is a testing methodology where the end customer is asked to use the software to see if the product is easy to use, to see the customer's perception and task time. The best way to finalize the customer point of view for usability is by using prototype or mockup software during the initial stages. By giving the customer the prototype before the development start-up we confirm that we are not missing anything from the user point of view.

Software Testing Image

Q What is End to End testing?

End-to-end testing is a technique used to test whether the flow of an application right from start to finish is behaving as expected. The **purpose of performing end-to-end testing is to identify system dependencies and to ensure that the data integrity is maintained between various system components and systems**. End to End Testing is considered as the part of System Testing, and this testing is always conducted (carried out) to validate the flow of data between different modules.

Q What is System Testing?

System Testing is a level of the software testing where a complete and integrated software is tested. **The purpose of this test is to evaluate the system's compliance with the specified requirements**. Definition by ISTQB. system testing: The process of testing an integrated system to verify that it meets specified. System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose.

Q Difference between Verification and Validation?

|  |  |
| --- | --- |
| Verification | Validation |
| Are we building the system right? | Are we building the right system? |
| Verification is the process of evaluating products of a development phase to find out whether they **meet the specified requirements.** | Validation is the process of evaluating software at the end of the development process to determine whether **software meets the customer expectations and requirements.** |
| Following activities are involved in Verification: Reviews, Meetings and Inspections. | Following activities are involved in Validation: Testing like black box testing, white box testing, gray box testing etc. |
| Involves all the static testing techniques | Includes all the dynamic testing techniques |

Q what is exploratory testing?

The exploratory testing means testing of software without any specific plans and schedules. This is formal testing process where we don’t have any test cases or test planning documents to test the application. Testers have to understand the application first by exploring the application and based on this understand they should come up with the test scenarios. After that start actual testing of application.

Q Describe Use Case Testing.

Use case is written by the user point of view. A use case is a description of the process which is performed by the end user for a particular task.

Q Define STLC

Requirement/Design review

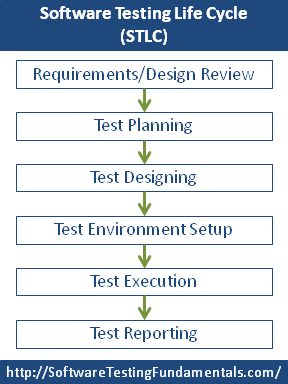
Test planning

Test Designing

Test Environment setup

Test Execution

Test reporting



|  |  |  |
| --- | --- | --- |
| Phase | Activity | Deliverables |
| Requirements/ Design Review | You review the software requirements/ design (Well, if they exist.) | * ‘Review Defect’ Reports |
| Test Planning | Once you have gathered a general idea of what needs to be tested, you ‘plan’ for the tests. | * [Test Plan](http://softwaretestingfundamentals.com/test-plan/) * Test Estimation * Test Schedule |
| Test Designing | You design/ detail your tests on the basis of detailed requirements/design of the software (sometimes, on the basis of your imagination). | * [Test Cases](http://softwaretestingfundamentals.com/test-case/) / [Test Scripts](http://softwaretestingfundamentals.com/test-script/) /Test Data * Requirements Traceability Matrix |
| Test Environment Setup | You setup the test environment (server/ client/ network, etc) with the goal of replicating the end-users’ environment. | * Test Environment |
| Test Execution | You execute your Test Cases/ Scripts in the Test Environment to see whether they pass. | * Test Results (Incremental) * [Defect Reports](http://softwaretestingfundamentals.com/defect-report/) |
| Test Reporting | You prepare various reports for various stakeholders. | * Test Results (Final) * Test/ Defect Metrics * Test Closure Report * Who Worked Late & on Weekends (WWLW) Report [Depending on how fussy your Management is] |

Q What is Static and Dynamic Testing?

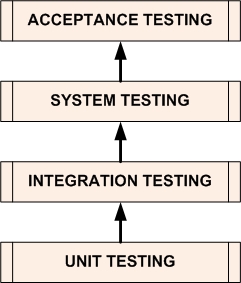
STATIC: manually checks the code, requirement documents, and design documents to find errors. Hence, the name "static". Main objective of this testing is to improve the quality of software products by finding errors in early stages of the development cycle.

DYNAMIC: Dynamic testing executes the software and validates the output with the expected outcome. Dynamic testing is performed at all levels of testing and it can be either black or white box testing. It checks for functional behavior of software system , memory/cpu usage and overall performance of the system. Hence the name "Dynamic".

# Software Testing Levels

# https://www.tutorialspoint.com/software\_testing/software\_testing\_levels.htm

**SOFTWARE TESTING LEVELS** are the different stages in the software development life cycle where testing is conducted. There are four levels of software testing: Unit >> Integration >> System >> Acceptance.



|  |  |
| --- | --- |
| **Level** | **Summary** |
| [Unit Testing](http://softwaretestingfundamentals.com/unit-testing/) | A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed. |
| [Integration Testing](http://softwaretestingfundamentals.com/integration-testing/) | A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. |
| [System Testing](http://softwaretestingfundamentals.com/system-testing/) | A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the **system’s compliance with the specified requirements.** |
| [Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/) | A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the **system’s compliance with the business requirements and assess whether it is acceptable for delivery**. |

Note: Some tend to include Regression Testing as a separate level of software testing but that is a misconception. Regression Testing is, in fact, just a type of testing that can be performed at any of the four main levels.

# Software Testing Types

**SOFTWARE TESTING TYPES** listed here are a few out of the hundreds of software testing types. The different types of testing you can perform on a software is limited only by the degree of your imagination. Here, we provide you summary of some of the major ones.

## LIST OF SOFTWARE TESTING TYPES

|  |  |
| --- | --- |
| **Type** | **Summary** |
| [Smoke Testing](http://softwaretestingfundamentals.com/smoke-testing/) | Smoke Testing, also known as “Build Verification Testing”, is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. |
| [Functional Testing](http://softwaretestingfundamentals.com/functional-testing/) | Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications. |
| [Usability Testing](http://softwaretestingfundamentals.com/usability-testing/) | Usability Testing is a type of software testing done from an end-user’s perspective to determine if the system is easily usable. |
| [Security Testing](http://softwaretestingfundamentals.com/security-testing/) | Security Testing is a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders. |
| [Performance Testing](http://softwaretestingfundamentals.com/performance-testing/) | Performance Testing is a type of software testing that intends to determine how a **system performs in terms of responsiveness and stability under a certain load**. |
| [Regression Testing](http://softwaretestingfundamentals.com/regression-testing/) | Regression testing is a type of software testing that intends to ensure that changes (enhancements or defect fixes) to the software have not adversely affected it. |
| [Compliance Testing](http://softwaretestingfundamentals.com/compliance-testing/) | Compliance Testing [also known as conformance testing, regulation testing, standards testing] is a type of testing to determine the compliance of a system with internal or external standards. |

## DISTINCTION

Please note that Software Testing **Types** are different from **Levels** or **Methods**. In contrast to the Software Testing Types listed above, [Software Testing Levels](http://softwaretestingfundamentals.com/software-testing-levels/) are the tests done at various stages of software development and [Software Testing Methods](http://softwaretestingfundamentals.com/software-testing-methods/) are the ways the tests are conducted. For example, you can do Functional Testing (A Type) during System Testing (A Level) using Black Box Testing (A Method).

Q What is Globalization Testing?

Ans. Process of verifying software whether it can be **run independent of its geographical and cultural environment**. Checking if the application is having features of setting and changing language, date, format and currency if it is designed for global users.

Q What is Localization Testing?

Ans. Verifying of globalized application **for a particular locality of users, cultural and geographical conditions**.

Q What is Acceptance Testing?  
Acceptance testing is performed after system testing is done and all or most of the major defects have been fixed. The goal of acceptance testing is to establish confidence in the delivered software/system that it meets the end user/customers requirements and is fit for use Acceptance testing is done by user/customer and some of the project stakeholders. Acceptance testing is done in production kind of environment.

Alpha Testing

Alpha testing is conducted at developer's site, potential users, members or developers organization are invited to use the system and report defects.

Beta Testing

Beta testing is also known as field testing, it is done by potential or existing users/customers at an external site without developers involvement, this test is done to determine that the software satisfies the end users/customers needs.

SELENIUM QUESTIONS

What are the annotations used in TestNG ?

Ans: @Test, @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeClass, @AfterClass, @BeforeMethod, @AfterMethod.

What are different types of locators?

Ans- There are 8 types of locators and all are the static methods of the By class.

By.id(), By.name(), By.tagName(), By.className(), By.linkText(), By.partialLinkText(), By.xpath, By.cssSelector().

How will you find an element using Selenium?

In Selenium every object or control in a web page is referred as an elements, there are different ways to find an element in a web page they are

* ID
* Name
* Tag
* Attribute
* CSS
* Linktext
* PartialLinkText
* Xpath etc

What is the difference between Assert and Verify?

Assert- it is used to verify the result. If the test case fails then it will stop the execution of the test case there itself and move the control to other test case.

Verify- it is also used to verify the result. If the test case fails then it will not stop the execution of that test case.

Explain what assertion in Selenium is and what are the types of assertion?

Assertion is used as a verification point. It verifies that the state of the application conforms to what is expected.  The types of assertion are “assert”, “Verify” and “wait For”.

What is XPath

XPath is a technique for uniquely identifying an element on a web page. It behaves like an address to a HTML element such as check boxes, text, images, links, and divs, etc. In Selenium, we treat XPath as one of the most trusted element locators. XPath is much more than an address as it not only points to the end point, it also contains the whole map to lead to a destination.

What is Firebug?

Firebug is the most famous add-on for the Firefox browser. It gives you control over the document object model of the web page so that you can modify, manage, and monitor the CSS, HTML, and JavaScript in real-time. You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page.

Problem with Page Objects:

Page objects is a **well known design pattern**, widely accepted by the automation engineers, to create separate class file for each page of the application to group all the elements as properties and their behaviors / business functionalities as methods of the class. But it might not be a great idea always, especially when the page has more / **different sets of elements** / **complex element like a grid** / **calendar widget** / a HTML table etc.

Difference between POM and Keyword Driven Framework.

As I understand, in **Page Object**, we must create a java class for each page. In **Keyword Driven Framework**, we need to create a generic module, which based on input key perform action based on predefined rule(s).

My question is, when we implement keyword driven framework, is page object not an option anymore? Can a keyword driven framework and Page Object implementation co-exist?

Yes they can coexists. Whatever way you want to test I see Page Object as the foundation. It lower the maintenance cost a lot.

**Keyword driven Framework is created as a higher abstraction layer so that non-technical easier understands the test case design.** Such as a function named login explains that it will login. This login function then uses the Page Objects to create element calls.

**So Page Object is only a lower level of abstraction to make it easier to create test cases.**

<http://www.ranorex.com/blog/keyword-driven-test-automation-framework>

This link explains what keyword driven is.

This is a sample keyword driven framework implemented in Selenium.

<http://www.testautomationguru.com/keyword-driven-framework-for-localization-testing-using-selenium-webdriver/>

It, kind of, does not make sense to use POM in the pure low level keyword driven framework. Because each and every element of the page and the corresponding action would be described in a spreadsheet.

If you are going for high level keyword driven or hybrid framework, then you can use POM.

Hybrid framework: <http://www.testautomationguru.com/hybrid-test-automation-framework/>

The above example is in QTP. But you will get the idea.

Page Object Design: <http://www.testautomationguru.com/arquillian-graphene-page-fragments/>

Keyword driven will work based on abstraction of keywords in your classes. For simple example click, input, button, dropdown, radio button, check box etc are keywords and have to maintain individual functions in your class. For maintain these most of them uses excel, csv.

Ex: BaseClass contains all the operations of key words like public void input(String locatorType, String locatorValue) //here locator type would be of xpath, id, name, class, tag and respective value

Issues:- Today I want to go with signup and signin, Ok no probs. I prepared steps and ran successfully then tomorrow management want to execute only login functionality. How can we manage, can you rewrite the steps? Can i use any testing framework, may be ambiguity comes from excel input and testng! So is it preferable? NO because it is easy to see but hard to manage!

Now Page Object Model:- It's not a framework, its a design pattern or methodology. Here if you use any testing framework(testng) its easy to maintain your methods/operations. We can use data driven that make sense.

So based on your project and need we have to maintain and prepare framework and stick with that framework.

I have Keyword Driven, Hybrid and Page Object Modal.

**Page Object Modal: It works well for the completed application. But if you are working on a agile then it is not suitable..** If the locator keeps on changing, every time you will get failure.. Consider that there is a cancel button on your sign in page and you are least bother about the change in cancel. if the locator changes for that then it will make you fail all the test case because every test case should come across login page only/.

Hybrid: Keyword+ POM:

It is a better choice. Because you keep every locator in a separate file and always you will look the locator whatever you need for that test case only.

WHY IS THE FIREBUG USEFUL IN SELENIUM AUTOMATION?

1- Display source – It gives you the ability to review the HTML of the web page after the JavaScript engine completes it processing.

2- Highlight changes – It allows to detect and highlight (in Yellow) any HTML changes as they appear on the web page. This feature would fetch your attention instantly to make sure nothing gets missed.

3- On the fly Inspection – FireBug has its “Inspect” option for quickly viewing the prospective locators as you exercise it on a web element.

4- Copy HTML – You can easily copy the HTML code of the page or part of a web page using the “innerHTML” property or the XPath expression for the element.

What is FirePath?

It is an **extension** to FireBug that adds a development tool to edit, inspect and generate XPath expressions and CSS3 Selectors.

Why it is Useful to Selenium Automation Tester

1) You can type self-written XPath and check if it is correct by highlighting the results directly on the Webpage.

2) Generate an XPath expression or a CSS selector for an element by right clicking on it and selecting “Inspect in FirePath” in the context menu.

3) Like Firebug it also gives you the Xpath of the selected Element.

Explain the difference between single and double slash in X-path?

Single Slash(/)

1. Single Slash(/) starts selection from the document node.
2. It allows us to create Absolute path expressions.

Double Slash(//)

1. It start  selection matching anywhere in the document
2. It enables to create relative path expression.

List out the technical challenges with Selenium?

1. Selenium supports only web based applications
2. For any reporting related capabilities have to depend on third party tools
3. Since Selenium is a freeware tool, there is no direct support if one is in trouble with the support of applications
4. There **is no object repository concept in** Selenium, so maintainability of the objects is very high

Why testers should opt for Selenium and not QTP?

1. Selenium is an open source whereas QTP is a commercial tool
2. Selenium supports Firefox, IE, Opera, Safari  on operating systems like Windows, Mac, linux etc. however QTP is limited to Internet Explorer on Windows.
3. Selenium supports many programming languages like Java, Ruby, Perl, Python whereas QTP supports only VB script

What is the difference between setSpeed() and sleep() methods?

Both will delay the speed of execution.

Thread.sleep () :  It will stop the current (java) thread for the specified period of time.  It is done only once. It takes a single argument in integer format

Ex: thread.sleep(2000)- It will wait for 2 seconds

* It waits only once at the command given at sleep

SetSpeed () :  For specific amount of time it will stop the execution for every selenium command. It takes a single argument in integer format

Ex: selenium.setSpeed(“2000”)- It will wait for 2 seconds

* Runs each command  after setSpeed delay by the number of milliseconds mentioned in set Speed

This command is useful for demonstration purpose or if you are using a slow web application

What is same origin policy? How you can avoid same origin policy?

The “Same Origin Policy” is introduced for security reason, and it ensures that **content of your site will never be accessible by a script from another site**.  As per the policy, any code loaded within the browser can only operate within that website’s domain.

**To avoid “Same Origin Policy” proxy injection method is used**, in proxy injection mode the Selenium Server acts as a **client configured HTTP proxy**, which sits between the **browser and application** under test and then **masks the AUT under a fictional URL**

What are the different types of waits available in WebDriver?

There are two [types of waits available in WebDriver](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):

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, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

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.**

WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);

**How can you find if an element in 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 button, 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();*

**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();*

**How to select value in a dropdown?**

Value in the drop down can be selected using WebDriver’s Select class.

Syntax:

selectByValue:

*Select selectByValue1 = new Select (driver.findElement (By.id(“SelectID\_One”)));*

*selectByValue1.selectByValue(“greenvalue”);*

selectByVisibleText:

*Select selectByVisibleText1 = new Select (driver.findElement(By.id(“SelectID\_Two”)));*

*selectByVisibleText1.selectByVisibleText(“Lime”);*

selectByIndex:

*Select selectByIndex1 = new Select(driver.findElement(By.id(“SelectID\_Three”)));*

*selectByIndex1.selectByIndex(2);*

**What are the different types of navigation commands?**

Following are the [navigation commands](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):

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”);*

How tohandle frame in WebDriver?

An inline frame acronym as iframe is used to insert another document with in 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();*

**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.findElements(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”));*

|  |  |
| --- | --- |
|  |  |

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

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. The command neither requires any parameter nor does is return any value.

quit(): Unlike close() method, quit() method closes down all the windows that the program has opened. Same as

close() method, the command neither requires any parameter nor does is return any value.

**How can we handle web based pop up?**

WebDriver offers the users with a very efficient way to [handle these pop ups **using Alert interface**](http://www.softwaretestinghelp.com/handle-alerts-popups-selenium-webdriver-selenium-tutorial-16/). 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();*

**How can we handle windows based pop up?**

Selenium is an automation testing tool which supports only web application testing, that means, it doesn’t support testing of windows based applications. However Selenium alone can’t help the situation but along with some third party intervention, this problem can be overcome. There are several third party tools available for handling window based pop ups along with the selenium **like AutoIT, Robot class** etc.

How to assert 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”));*

**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 method.
* @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 method.
* @AfterClass: Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test method.
* @Ignore: Method annotated as @Ignore lets the system know that this method shall not be executed.

**What is TestNG and how is it better than Junit?**

[TestNG](http://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/) is an advance framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, JUnit gained an 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 Web Driver'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

**What is a framework?**

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.

What are the advantages of Automation framework?

Advantage of [Test Automation framework](http://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/)

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

**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 number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts 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. Behavior Driven Development Framework: Behavior Driven Development framework allows automation of functional validations in easily readable and understandable format to Business Analysts, Developers, Testers, etc.

**What is Object Repository? How can we create 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 hard coding them within the scripts.**

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

That’s all for now.

**Q Mention 5 different exceptions you had in Selenium web driver?**

The 5 different exceptions you had in Selenium web drivers are

* WebDriverException
* NoAlertPresentException
* NoSuchWindowException
* NoSuchElementException
* TimeoutException

**Explain how you can switch back from a frame?**

To switch back from a frame use method defaultContent()

Syntax-driver.switchTo().defaultContent();

**Run Failed Test Cases Using TestNG in Selenium WebDriver:**

Case 1: Execute failed test cases using TestNG in Selenium – By using “testng-failed.xml”

Steps To follow:

1. After the first run of an automated test run. Right click on Project – Click on Refresh
2. A folder will be generated named “test-output” folder. Inside “test-output” folder, you could find “testng-failed.xml”
3. Run “testng-failed.xml” to execute the failed test cases again.

Case 2: Execute failed test cases using TestNG in Selenium – By Implementing TestNG IRetryAnalyzer.

Create a class to implement IRetryAnalyzer. Here I am creating a class (say, RetryFailedTestCases) and implementing IRetryAnalyzer.

RetryFailedTestCases implements IRetryAnalyzer:

package softwareTestingMaterial;

import org.testng.IRetryAnalyzer;

import org.testng.ITestResult;

public class RetryFailedTestCases implements IRetryAnalyzer {

    private int retryCnt = 0;

    //You could mentioned maxRetryCnt (Maximiun Retry Count) as per your requirement. Here I took 2, If any failed testcases then it runs two times

    private int maxRetryCnt = 2;

    //This method will be called everytime a test fails. It will return TRUE if a test fails and need to be retried, else it returns FALSE

    public boolean retry(ITestResult result) {

        if (retryCnt < maxRetryCnt) {

            System.out.println("Retrying " + result.getName() + " again and the count is " + (retryCnt+1));

            retryCnt++;

            return true;

        }

        return false;

    }

}

*Case3: (Best One)*

Creates a new class

Create an Object of TestNG class

There is a SetTestSuites()  method, pass a List object to this method.

Create a List Object and add the path of the testng-failed.xml

And there is a method Run() of TestNG class, execute the same.

Note: We can pass more than one XML’s path to the Add method.

package runner;

import java.util.ArrayList;

import java.util.List;

import org.testng.TestNG;

public class testrunner {

public static void main(String[] args)

{

//creating TestNG class object

TestNG testrunner = new TestNG();

//creating a list object

List <String> suites = new ArrayList<String>();

//add the path of the testng-failed.xml

suites.add("C:\\Users\\pankaj.kalra\\eclipse-workspace\\TestNG\_POM\\test-output\\testng-failed.xml");

//call the TestNG class’s object and pass the list object as a parameter.

testrunner.setTestSuites(suites);

//run the method

testrunner.run();

}

}

**Action Class:**

Handling MouseOver menu

* + - 1. Create Action class object
      2. Use movetoelement() and pass the webelement in the parameter
      3. Build().perform()

Actions obj = new Actions (driver);

WebElement element = driver.findelement (By.xpath “”);

Obj.movetoelement (element).build().perform()

Handling Right Click

Actions obj = new Actions (driver);

WebElement element = driver.findelement (By.xpath “”);

//here Action is an interface and perform() is its function

Action rightclick = obj.contextClick(element).build();

rightclick.perform();

Drag and Drop.

Actions obj = new Actions (driver);

WebElement source = driver.findelement (By.xpath “”);

WebElement target = driver.findelement (By.xpath “”);

Obj.dragAndDrop(source, target).build().perform();

========================================================

Exceptions in Selenium

*An Exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program’s instructions or in simple words, any issue which makes your test case stop in between the execution.*

**The process of creating the exception object and handing it over to run-time environment is called “throwing the exception”.**

* NoSuchElementException
* ElementNotVisibleException
* NoSuchFrameException
* NoAlertPresentException(This Exception occurs when the driver is switching to an invalid Alert, which is not available. E.g. driver.switchTo().alert().accept();)
* NoSuchWindowException
* WebDriverException (This Exception occurs when the driver is performing the action after immediately closing the browser.)
* SessionNotFoundException (This Exception occurs when the driver is performing the action after immediately quitting the browser.)
* ElementNotSelectableException
* TimeoutException

**Difference between Exception and Error:**

Exception: Exception occurs in the programmers code which can be handled and resolvable.

---->>>>>>> Example: AritmeticException, DivideByZeroException, NullPointerException, ClassNotFoundException etc

Error: Errors are not resolvable by programmer. Error occurs due to lack of system resources

---->>>>>>> Example: Stack over flow, hardware error, JVM error etc.

Throw: Sometimes we want to generate exception explicitly in our code, for example in Selenium Automation Framework most of the time we print self-written logs, once we catch an exception and then we need to throw that exception back to the system so that the test case can be terminated. Throw keyword is used to throw exception to the runtime to handle it.

Throws: When we are throwing any exception in a method and not handling it, then we need to use throws keyword in method signature to let caller program know the exceptions that might be thrown by the method.

Finally: The finally keyword is used to create a block of code that follows a try block. A finally block of code always executes, whether or not an exception has occurred.

=======================================================================

File Upload using AutoIT

<http://learn-automation.com/upload-file-in-selenium-webdriver-using-autoit/>

How To Skip TestNG Test:

One way of skipping a test method is by using *throw new SkipException()* exception.Once SkipException() thrown, remaining part of that test method will not be executed and control will goes directly to next test method execution.

|  |
| --- |
| package softwareTestingMaterial;    import org.testng.annotations.Test;  import org.testng.SkipException;    public class SkipTestCase {  @Test  public void aSkipTest(){  String a ="Skip Test";  if(a.equals("Skip Test")){  throw new SkipException("Skipping - This is not ready for testing);  }else{  System.out.println("I am in else condition");  }  System.out.println("I am out of the if else condition");  }    @Test  public void nonSkipTest(){  System.out.println("No need to skip this test");  }  }  Output:  [TestNG] Running:  No need to skip this test  ===============================================  Default suite  Total tests run: 2, Failures: 0, Skips: 0  =============================================== |

*JAVA*

Java static keyword - The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than instance of the class.

Java static property is shared to all objects.

1. JAVA Static Variable:  The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc. The static variable gets memory only once in class area at the time of class loading. It makes your program memory efficient (i.e. it saves memory).

Suppose there are 500 students in my college, now all instance data members will get memory each time when object is created. All students have its unique roll no and name so instance data member is good. Here, college refers to the common property of all objects. If we make it static, this field will get memory only once.

1. Java static method -

* A static method belongs to the class rather than object of a class.
* A static method can be invoked without the need for creating an instance of a class.
* Static method can access static data member and can change the value of it.
* The static method cannot use non-static data member or call non-static method directly.

We can call static methods directly while we cannot call non static methods directly. You need to create and instantiate an object of class for calling non static methods.

Non static stuff (methods, variables) can not be accessible Inside static methods Means we can access only static stuff Inside static methods. Opposite to It, Non static method do not have any such restrictions. We can access static and non static both kind of stuffs Inside non static methods

* We can not override a static method but we can overload the static method.

OOP Principle:

1. Encapsulation: Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

Encapsulation is the technique of making the fields (variables) in a class private and providing access to the fields via public methods. If a field is declared private, it cannot be accessed by anyone outside the class, thereby hiding the fields within the class.

Example: When we try to login to our Gmail account, there mapping of data and authentication algorithm is running on backend. They are attached in such a way that if the authentication is successfully completed, then only the login will take place.

2. Abstraction: abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does it. In Java, abstraction is achieved using Abstract classes and interfaces.

An abstract class is a class that is declared with abstract keyword. If class contains any abstract method then you must have to declare your class as abstract class in java software development language.  Also abstract classes can be subclassed, but they cannot be instantiated. That means you cannot create object of abstract class. It may or may not have abstract methods.

Abstract method cannot have a body. Actual implementation of abstract method will be done by it's child class. If any class extends abstract class then that subclass must have to implement all the abstract methods declared by it's super class (abstract class).

abstract class *Vehicle*  
{  
   public abstract void *engine*();    
}  
public class *Car* extends *Vehicle* {  
    public void *engine*()  
    {  
        System.out.println("Car engine");     
        //car engine implementation  
    }  
    public static void main(String[] args)  
    {  
        Vehicle v = new Car();  
        v.*engine*();  
    }  
}

POLYMORPHISM: Polymorphism in java is a concept by which we can perform a *single action by different ways*. Polymorphism is the capability of a method to do different things based on the object that it is acting upon. In other words, polymorphism allows you define one interface and have multiple implementations.

Method Overloading: In Java, it is possible to define two or more methods of same name in a class, provided that there argument list or parameters are different. This concept is known as Method Overloading.

When Java encounters a call to an overloaded method, it simply executes the version of the method whose parameters match the arguments used in the call.

It allows the user to achieve compile time polymorphism.

Rules for Method Overloading

1. Overloading can take place in the same class or in its sub-class.
2. Constructor in Java can be overloaded
3. Overloaded methods must have a different argument list.
4. The parameters may differ in their type or number, or in both.
5. They may have the same or different return types.

Class Overload  
{  
    void demo (int a)  
    {  
       System.out.println ("a: " + a);  
    }  
    void demo (int a, int b)  
    {  
       System.out.println ("a and b: " + a + "," + b);  
    }  
    double demo(double a) {  
       System.out.println("double a: " + a);  
       return a\*a;  
    }  
}  
class MethodOverloading  
{  
    public static void main (String args [])  
    {  
        Overload Obj = new Overload();  
        double result;  
        Obj .demo(10);  
        Obj .demo(10, 20);  
        result = Obj .demo(5.5);  
        System.out.println("O/P : " + result);  
    }  
}

Here the method demo() is overloaded 3 times: first having 1 int parameter, second one has 2 int parameters and third one is having double arg. The methods are invoked or called with the same type and number of parameters used.

Output:

a: 10  
a and b: 10,20  
double a: 5.5  
O/P : 30.25

METHOD OVERRIDING Child class has the same method as of base class. In such cases child class overrides the parent class method without even touching the source code of the base class. This feature is known as method overriding.

Rules for Method Overriding:

1. applies only to inherited methods
2. Overriding method can have different return type ([refer this](http://stackoverflow.com/questions/14694852/can-overridden-methods-differ-in-return-type))
3. Abstract methods must be overridden
4. Static and final methods cannot be overridden
5. Constructors cannot be overridden
6. It is also known as Runtime polymorphism.

public class BaseClass  
{  
    public void methodToOverride() //Base class method  
    {  
         System.out.println ("I'm the method of BaseClass");  
    }  
}  
public class DerivedClass extends BaseClass  
{  
    public void methodToOverride() //Derived Class method  
    {  
         System.out.println ("I'm the method of DerivedClass");  
    }  
}  
  
public class TestMethod  
{  
     public static void main (String args []) {  
        // BaseClass reference and object  
        BaseClass obj1 = new BaseClass();   
        // BaseClass reference but DerivedClass object  
        BaseClass obj2 = new DerivedClass();   
        // Calls the method from BaseClass class  
        obj1.methodToOverride();   
        //Calls the method from DerivedClass class  
        obj2.methodToOverride();   
     }  
}

Output:

I'm the method of BaseClass  
I'm the method of DerivedClass

**Difference between Abstract Class and Interfaces.**

|  |  |
| --- | --- |
| Abstract class | Interface |
| 1) Abstract class can have abstract and non-abstract methods. | Interface can have only abstract methods. |
| 2) Abstract class doesn't support multiple inheritance. | Interface supports multiple inheritance. |
| 3) Abstract class can have final, non-final, static and non-static variables. | Interface has only static and final variables. |
| 4) Abstract class can have static methods, main method and constructor. | Interface can't have static methods, main method or constructor. |
| 5) Abstract class can provide the implementation of interface. | Interface can't provide the implementation of abstract class. |
| 6) The abstract keyword is used to declare abstract class. | The interface keyword is used to declare interface. |
| 7) Example:  public abstract class Shape{  public abstract void draw();  } | Example:  public interface Drawable{  void draw();  } |

INTERFACE

Interface Is looks like class but It Is not class. When you implements that Interface In any class then all those Interface rules must be applied on that class. In sort, If you Implement an Interface on class then you must have to override all the methods of Interface In your class.

Interface can be Implemented with any class using *implements* keyword. There are set of rules to be followed for creating an Interface. Let me tell you all these rules first and then give you an example of an Interface.

* Interface can not hold constructor.
* Interface can not hold instance fields/variables.
* Interface can not hold static methods.
* You can not instantiate/create object of an Interface.
* Variables Inside Interface must be static and mandatory to initialize the variable.
* Any class can Implement Interface but can not extend Interface.
* Can write body less methods Inside Interface.
* By default all the methods and variables of Interface are public so no need to provide access modifiers.

Interview Questions:

1. Tell me about yourself, I am more interested to know why QA?
2. Difference b/w IOS and Android testing? According to you which one is tougher ?
3. Suppose you have one fresher, one mid level or another senior level, how would you define their KPIs.
4. Top two qualities as scrum master?
5. How would you help your QA to perform manual testing in a fastest ways?
6. We are looking for Mobile Automation tester? Tell me your exposure on that?

***Difference between Adhoc and Exploratory Testing***

|  |  |
| --- | --- |
| Adhoc Testing | Exploratory Testing |
| 1. Adhoc Testing means learn the application than test it. | 1. Exploratory Testing means test the application while learning. |
| 2. In Adhoc Testing QA is always asked to test an application **with** detailed set of documents. | 2. In Exploratory Testing QA is always asked to test an application **without** any specific set of documents. |
| 3. In this Testing we always gather information regarding the software/application from complete possible sources and document and then test the application/software. | 3. In this Testing we gather the information, and also document and test the application simultaneously. |
| 4. **In Adhoc Testing tester should have good knowledge about the application in order to test the software.** | 4. In Exploratory Testing tester should **increases their knowledge by exploring the application/software.** |
| 5. In this testing testers have significant testing of the software before test it. | 5. In this testing testers may be learning the software before testing it. |
| 6. It is not considered as a type of any. | 6. It is considered as a type of Adhoc Testing |
| 7. It is not an approach to testing. | 7. It is an approach, not a technique. |

1. [**Ad Hoc Testing**](http://qatestlab.com/services/No-Documentation/ad-hoc-testing/) implies learning of the software before its testing. During Exploratory Testing, you learn and test the software simultaneously.
2. Before Ad Hoc Testing, we collect data on the particular software from different resources, and then we start the testing. During Exploratory Testing, we **collect data and perform the testing at the same time.**
3. For Ad Hoc Testing it is necessary to know the software well. For Exploratory Testing you have to gain your knowledge in course of work with the software.
4. Before Ad Hoc Testing, you need to perform considerable software testing. Before Exploratory Testing of the software, you may learn this software.
5. Exploratory Testing is a type of Ad Hoc Testing. Ad Hoc Testing is an independent testing type.

**Ad Hoc Testing is not considered to be testing approach. Exploratory Testing is an approach.**

**Exploratory Testing** is a testing approach that allows you to apply your ability and skill as a tester in a powerful way. Testers have to understand the application first by exploring the application, finding out about the software. Exploratory testing is a simultaneous process of test design and test execution all done at the same time. The focus of exploratory testing is more on testing as a "thinking" activity. See it [Pros and Cons](http://www.guru99.com/exploratory-testing.html). It is a bit formal process than Ad-hoc testing. [Sometimes it is referred as improved version of Ad-hoc](http://whatis.techtarget.com/definition/ad-hoc-testing).

**Ad-hoc** testing is an informal testing type with **an aim to break the system**. This testing is usually an unplanned activity . It does not follow any test design techniques to create test cases. This testing is primarily performed if the knowledge of testers in the system under test is very high. This testing requires no documentation/ planning /process to be followed. Usually [Ad-Hoc](http://www.guru99.com/adhoc-testing.html) testing is performed after the formal test execution. [It has following types](http://www.tutorialspoint.com/software_testing_dictionary/adhoc_testing.htm):-

1. Buddy testing
2. Pair Testing
3. **Monkey Testing**

Ad-hoc testing is itself known as **Monkey** or **Random** testing.

Exploratory testing involves the simultaneous learning, test design and test execution.

If you're working out the random inputs while you're in session and then learning something about the system behavior from the outputs, then you're conducting monkey tests inside an exploratory testing session.

Ad-hoc testing refers to whether the test was planned or not.

Both monkey tests and exploratory test sessions can be planned or unplanned, thus either can be ad-hoc

## Difference between authentication and authorization.

|  |  |
| --- | --- |
| **Authentication** | **Authorization** |
| It is the process of verifying the identity of a user. | It is the process of checking whether the user has the access rights to the system. |
| It always proceeds to authorization. | It is the process of allowing an authenticated user access to resources. |
| It has two separate levels because all the requests coming through the IIS before it is handled. | It allows two ways to authorize the access to a given resources. |
| They have additional schemes like windows authentication, forms authentication and passport authentication. | The two ways are URL authorization and File authorization |

**Sessions** - are created & managed by the **web server (e.g,. Apache)** - and these sessions are used to identify and manage valid User Sessions (HTTP is stateless by nature!).

Once you login to any web application, and the given credentials (user id and password) are validated by the application, the associated Web Server starts a unique session exclusively for you. And this is how **server** **keeps recognizing** your one after another HTTP requests you send to the web server.

Finally, either the Sessions are timed-out after a pre-set duration (e.g. 30 minutes) or they are programmed to expire - after a certain period of activities or eventually an active session is terminated once you explicitly logout.

**Cookies** are something - that are sent by the **web servers** to browsers (aka clients) and usually these cookies resides on the client machines (computers) - for the specified period of time (until it expires). They comprise of small set data / files

The cookies are used for various purposes. One of most common usage is to **monitor the pattern of user’s (client’s) activity on a website** (who has sent the cookie)

Like sessions, cookies can also be programmed to expire after a certain duration or under certain circumstances.

All **the server side programming languages e.g. PHP** provides the infrastructure to manage the life cycle of cookies and sessions.

# Cookies and Sessions

## Stateless applications

* **Web application servers are generally "stateless":**
  + Each HTTP request is independent; server can't tell if 2 requests came from the same browser or user.
  + Web server applications maintain no information in memory from request to request (only information on disk survives from one request to another).
* Statelessness not always convenient for application developers: need to tie together a series of requests from the same user.

## Browser cookies

* Cookie basics:
  + The first time a browser connects with a particular server, there are no cookies.
  + When the server responds it includes a Set-Cookie: header that defines a cookie.
  + Each cookie is just a name-value pair.
  + In the future whenever the browser connects with the same server, it includes a Cookie: header containing the name and value, which the server can use to connect related requests.
* What's in a cookie?
  + Name and data.
    - Data size limited by browsers (typically < 4 KB).
    - A server can define multiple cookies with different names, but browsers limit the number of cookies per server (around 50).
  + Domain for this cookie: server, port (optional), URL prefix (optional). The cookie is only included in requests matching its domain.
  + Expiration date: browser can delete old cookies.

## Sessions

* **Cookies are used by the server to implement *sessions*:**
  + A pool of data related to an active connection (one browser instance).
* Typically the **cookie for an application contains an identifier for a session.**
* Web frameworks like Rails do most of the work of managing sessions and cookies:
  + Rails provides session, a hash-like object in which you can store anything you like
    - Data will be available in all future requests from the same browser.
  + Rails automatically checks for a session cookie at the start of each request:
    - Cookie exists? use it to find session data
    - No cookie? Create new session, new cookie
  + End of each request: save session data where it can be found by future requests.
* Managing session state:
  + Approach #1: just keep state in main memory
  + Approach #2: store session state in files on disk
  + Approach #3: store session state in a database
  + Most frameworks allow you to control session storage:
    - Provide an object that saves and restores session data.
* **Server must eventually delete stale session data.**
* **Sessions have numerous security issues, which we will discuss later.**

***Introduction to Mobile Application Testing:***

Gone are the days when the telephone used to be an appliance that sat in a corner and had to ring to get our attention or a computer was a machine only a few people used – they are now an extension of our being- a window to the world and virtual servants that do as they are told. Computers were a rage and changed how we humans thought, behaved, learnt and existed.

Mobile technology and smart devices are the trend now and will change the future of the world as we know it. We all can vouch for it, can’t we? Now, it will be amateurish if I list what we use these mobile devices for. You all know it – Maybe better than we do. J

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/mobile-application-testing-1.jpg)

Let’s get straight to what this tutorial is going to be about.

**This tutorial will be both an introduction and your guide to Mobile Testing. So, read through!**

**What You Will Learn:**[[show](http://www.softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/)]

### Types of Mobile Testing

There are broadly 2 kinds of testing that take place on mobile devices:

**#1. Hardware testing:**

The device including the internal processors, internal hardware, screen sizes, resolution, space or memory, camera, radio, Bluetooth, WIFI etc. This is sometimes referred to as, simple “Mobile Testing”.

**#2. Software or Application testing:**

The applications that work on mobile devices and their functionality are tested. It is called the “Mobile Application Testing” to differentiate it from the earlier method. Even in the mobile applications, there are few basic differences that are important to understanding:

* 1. **Native apps:** A native application is created for use on a platform like mobile and tablets.
  2. **Mobile web apps** are server-side apps to access website/s on mobile using different browsers like Chrome, Firefox by connecting to a mobile network or wireless network like WIFI.
  3. **Hybrid apps** are combinations of native app and web app. They run on devices or offline and are written using web technologies like HTML5 and CSS.

**There are few basic differences that set these apart:**

* Native apps have single platform affinity while mobile web apps have the cross-platform affinity.
* Native apps are written in platforms like SDKs while Mobile web apps are written with web technologies like HTML, CSS, asp.net, Java, PHP.
* For a native app, installation is required but for mobile web apps, no installation is required.
* A native app can be updated from play store or app store while mobile web apps are centralized updates.
* Many native apps don’t require Internet connection but for mobile web apps, it’s a must.
* Native app works faster when compared to mobile web apps.
* Native apps are installed from app stores like [Google play store](https://play.google.com/store?hl=en" \o "Google app store) or [app store](http://www.apple.com/osx/apps/app-store.html" \o "App store) where mobile web are websites and are only accessible through the Internet.

***The rest of the article is going to be about Mobile Application Testing.***

### Significance of Mobile Application Testing

Testing applications on mobile devices is more challenging than testing web apps on desktop due to

* **Different range of mobile devices** with different screen sizes and hardware configurations like a hard keypad, virtual keypad (touch screen) and trackball etc.
* **Wide varieties of mobile devices** like HTC, Samsung, Apple and Nokia.
* **Different mobile operating systems** like Android, Symbian, Windows, Blackberry and IOS.
* **Different versions of operation system** like iOS 5.x, iOS 6.x, BB5.x, BB6.x etc.
* **Different mobile network operators** like GSM and CDMA.
* Frequent updates – (like Android- 4.2, 4.3, 4.4, iOS-5.x, 6.x) – with each update a new testing cycle is recommended to make sure no application functionality is impacted.

As with any application, Mobile application testing is also very important, as the clientele is usually in millions for a certain product – and a product with bugs is never appreciated. It often results in monetary losses, legal issue and irreparable brand image damage.

### Basic Difference Between Mobile and Desktop Application Testing:

**Few obvious aspects that set mobile app testing apart from the desktop testing**

* On the desktop, the application is tested on a central processing unit. On a mobile device, the application is tested on handsets like Samsung, Nokia, Apple and HTC.
* Mobile device screen size is smaller than desktop.
* Mobile devices have less memory than desktop.
* Mobiles use network connections like 2G, 3G, 4G or WIFI where desktop use broadband or dial-up connections.
* The automation tool used for desktop application testing might not work on mobile applications.

### Types of Mobile App Testing:

**To address all the above technical aspects, the following types of testing are performed on Mobile applications.**

* **[Usability testing](http://www.softwaretestinghelp.com/usability-testing-guide/" \o "Usability testing)**– To make sure that the mobile app is easy to use and provides a satisfactory user experience to the customers
* **Compatibility testing**– Testing of the application in different mobiles devices, browsers, screen sizes and OS versions according to the requirements.
* **Interface testing**– Testing of menu options, buttons, bookmarks, history, settings, and navigation flow of the application.
* **Services testing**– Testing the services of the application online and offline.
* **Low-level resource testing**: Testing of memory usage, auto-deletion of temporary files, local database growing issues known as low-level resource testing.
* **[Performance testing](http://www.softwaretestinghelp.com/introduction-to-performance-testing-loadrunner-training-tutorial-part-1/" \o "Performance testing)**– Testing the performance of the application by changing the connection from 2G, 3G to WIFI, sharing the documents, battery consumption, etc.
* **Operational testing**– Testing of backups and recovery plan if a battery goes down, or data loss while upgrading the application from a store.
* **[Installation tests](http://www.softwaretestinghelp.com/software-installationuninstallation-testing/" \o "Unstallation uninstallation testing)–** Validation of the application by installing /uninstalling it on the devices.
* **[Security Testing](http://www.softwaretestinghelp.com/category/security-testing/" \o "Security testing)**– Testing an application to validate if the information system protects data or not.

### Mobile Application Testing Strategy

The Test strategy should make sure that all the quality and performance guidelines are met. A few pointers in this area:

1. **Selection of the devices**–Analyze the market and choose the devices that are widely used. (This decision mostly relies on the clients. The client or the app builders consider the popularity factor of certain devices as well as the marketing needs for the application to decide what handsets to use for testing.)
2. **Emulators –**The use of these is extremely useful in theinitial stages of development, as they allow quick and efficient checking of the app. The emulator is a system that runs software from one environment to another environment without changing the software itself. It duplicates the features and works on the real system.

**Types of Mobile Emulators**

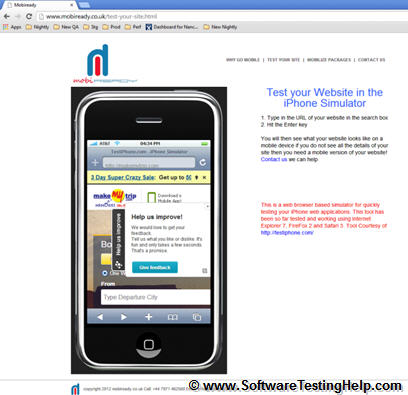
* Device Emulator- provided by device manufacturers
* Browser Emulator- simulates mobile browser environments.
* Operating systems Emulator- Apple provides emulators for iPhones, Microsoft for Windows phones and Google Android phones

**List of few free and easy to use mobile device emulators**

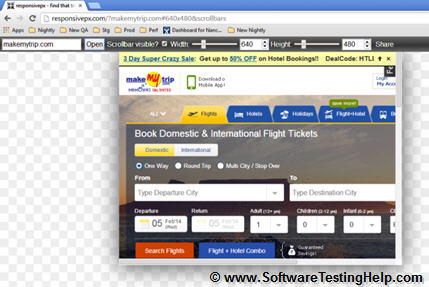
**i.** **[Mobile Phone Emulator](http://www.mobilephoneemulator.com/" \o "http://www.mobilephoneemulator.com/)** – Used to test handsets like iPhone, Blackberry, HTC, Samsung etc.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/mobile-device-emulator-2.jpg)

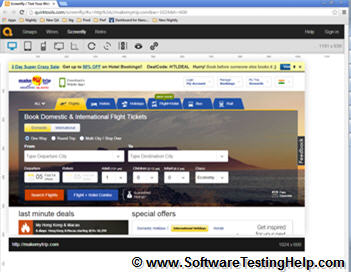
**ii. [MobiReady](http://ready.mobi/launch.jsp?locale=en_EN" \o "http://ready.mobi/launch.jsp?locale=en_EN)** – With this, not only can we test the web app, we can also check the code.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/mobile-device-emulator-3.jpg)

1. **[Responsivepx](http://responsivepx.com/" \o "http://responsivepx.com/)** – It checks the responses of the web pages, appearances and functionality of the websites.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/mobile-device-emulator-4.jpg)

**iv. [Screenfly](http://quirktools.com/screenfly/" \o "http://quirktools.com/screenfly/)** – It is a customizable tool and used to test websites under different categories.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/mobile-device-emulator-5.jpg)

**3)** After a satisfactory level of development is complete for the mobile app, you could move to test on the **physical devices** for more real-life scenarios based testing.

**4) Consider cloud computing based testing:** [Cloud computing](http://en.wikipedia.org/wiki/Cloud_computing" \o "Cloud Computing) is basically running devices on multiple systems or networks via the Internet where applications can be tested, updated and managed. For testing purposes, it creates the web-based mobile environment on a simulator to access the mobile app.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/cloud-based-mobile-testing.jpg)

**Pros:**

* Backup and recovery- Cloud computing automatically takes back up of your data from remote location making recovery and restoring of data easily. And also, the storage capacity is unlimited.
* Clouds can be accessed from different devices and anywhere.
* Cloud computing is cost-efficient, easy to use, maintain and update.
* Fast and quick deployment.
* Web-based interface.
* Can run the same script on several devices in parallel.

**Cons**

* **Less control**– Since the application runs on the remote or third-party environment, the user has limited control and access over the functions.
* **Internet connectivity issues**– the setup is on the Internet. Network issues affect the availability and functioning
* **Security and privacy Issues**– Cloud computing is an Internet computing and nothing on the Internet is completing secure, so chances of data hacking are more.

**5) [Automation vs. Manual testing](http://www.softwaretestinghelp.com/manual-and-automation-testing-challenges/" \o "Manual vs automation testing)**

* If the application contains new functionality, test it manually.
* If the application requires testing once or twice, do it manually.
* Automate the scripts for regression test cases. If regression tests are repeated, automated testing is perfect for that.
* Automate the scripts for complex scenarios which are time-consuming if executed manually.

**Two kinds of automation tools are available to test mobile apps:**

**Object-based mobile testing tools**– automation by mapping elements on the device screen into objects. This approach is independent of screen size and mainly used for Android devices.

* Eg:- Ranorex, jamo solution

**Image-based mobile testing tools**– create automation scripts based on screen coordinates of elements.

* Eg:- Sikuli, Egg Plant, RoutineBot

**6) Network** **configuration** is also the necessary part of mobile testing. It’s important to validate the application on different networks like 2G, 3G, 4G or WIFI.

### Test Cases for Testing a Mobile App

In addition to functionality based test cases, Mobile application testing requires special test cases which should cover following scenarios.

* **Battery usage**– It’s important to keep a track of battery consumption while running application on the mobile devices.
* **The speed of the application-**the response time on different devices, with different memory parameters, with different network types etc.
* **Data requirements**– For installation as well as to verify if the user with the limited data plan will able to download it.
* **Memory requirement**– again, to download, install and run
* **The functionality of the application**– make sure application is not crashing due to network failure or anything else.

**Download Some Sample Test Cases for Testing Mobile Applications:**

=> **[Download Mobile app sample test cases](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/03/Mobile-Application-Sample-Test-Cases.xls" \o "Mobile testing test cases)**

### Typical activities and proceedings in Testing Mobile Application

The scope of the testing depends on a number of requirements to be checked or the extent of changes made to the app. If the changes are few, a round of **sanity** testing will do. In case of major and/or complex changes, a **full regression** is recommended.

**An example application testing project**: ILL (International Learn Lab) is an application designed to help admin, publisher to create websites in collaboration. Using a web browser, instructors choose from a set of features to create a class that meets their requirements.

**Mobile Testing process:**

**Step #1. Identify the [types of testing](http://www.softwaretestinghelp.com/types-of-software-testing/" \o "Testing types)**: As an ILL application is applicable for browsers, so it’s mandatory to test this application on all supported browsers using different mobile devices. We need to do **usability, functional** and **compatibility** testing on different browsers with the **combinations** of **manual** and **automation** test cases.

**Step #2.** **Manual and Automated testing:** The methodology followed for this project is Agile with the iteration of two weeks. Every two weeks dev. team releases a new build to testing team and testing team will run their test cases on QA environment. Automation team creates scripts for the set of basic functionality and runs the scripts that help determine if the new build is stable enough to test. The Manual testing team will test the new functionality.

[JIRA](http://www.softwaretestinghelp.com/atlassian-jira-tutorial-1/" \o "JIRA test management tool Tutorials) is used for writing of acceptance criteria; maintaining of test cases and logging /re-verification of defects. Once the iteration gets over, **iteration** **planning** meeting held where dev. The team, product owner, business analyst, and QA team discuss **what went well** and **what needs to improve**.

**Step #3. [Beta Testing](http://www.softwaretestinghelp.com/what-is-alpha-testing-beta-testing/" \o "Beta testing):** Once the regression testing is completed by the QA team, the build moves into UAT. User Acceptance Testing is done by the client. They re-verify all the bugs to make sure every bug was fixed and the application is working as expected on every approved browser.

**Step #4. Performance test:** Performance testing team tests the performance of the web app using JMeter scripts and with different the loads on the application.

**Step #5. [Browser testing](http://www.softwaretestinghelp.com/best-cross-browser-testing-tools-to-ease-your-browser-compatibility-testing-efforts/" \o "Cross browser testing):** The web app gets tested across multiple browsers- both using different simulation tools as well as physically using real mobile devices.

**Step #6. Launch plan:** After every 4th week, the testing moves into staging, where a final round of end to end testing on these devices is performed to make sure the product is ready for production. And then, it goes Live!

### Conclusion

Designing the right test strategy, choosing the right mobile simulators, devices and mobile testing tools can make sure that we have 100% test coverage and help us include security, usability, performance, functionality and compatibility based tests into our test suites.

Well, this has been our effort to fulfil multiple requests from our readers on a mobile application testing guide.

**About Author:**This is a guest post by Nancy Ratnakar. She is a Senior QA Engineer, in an MNC with more than 5 years of experience in QA and mobile testing. With extensive experience in STLC, she is an expert handling projects from initial requirement stage to the final release.

***Please let us know how we did in the comments.  Also, share your experiences if you are working or have worked on this kind of mobile testing. Your questions and suggestions are most welcome!***

#### JSON

[JSON](http://www.json.org/) (or JavaScript Object Notation) is a lightweight, easy and popular way to exchange data. jQuery is not the only tool for manipulating and interfacing with JSON; it’s just my and many others’ preferred method.

A lot of the services we use everyday have JSON-based APIs: Twitter, Facebook and Flickr all send back data in JSON format.

# Fetching JSON data from REST APIs

<https://cran.r-project.org/web/packages/jsonlite/vignettes/json-apis.html>

This section lists some examples of public HTTP APIs that publish data in JSON format. These are great to get a sense of the complex structures that are encountered in real world JSON data. All services are free, but some require registration/authentication. Each example returns lots of data, therefore not all output is printed in this document.

## Github

Github is an online code repository and has APIs to get live data on almost all activity. Below some examples from a well known R package and author:

hadley\_orgs <- fromJSON("https://api.github.com/users/hadley/orgs")

hadley\_repos <- fromJSON("https://api.github.com/users/hadley/repos")

gg\_commits <- fromJSON("https://api.github.com/repos/hadley/ggplot2/commits")

gg\_issues <- fromJSON("https://api.github.com/repos/hadley/ggplot2/issues")

Paste the above url in the url and test with Rest Console:

### JSON Webservices

GeoNames offers most webservices in XML and JSON format. JSON has the advantage that is can be used to directly access the geonames webservice from javascript code, whereas using XML the browser will throw a security exception if an xml call to another webserver is made.

All JSON services accept an optional parameter 'callback' for a javascript function call and a parameter 'formatted=true' to format the output with linefeeds and indentation. The latter is useful to view the JSON result in a browser but should not be used in production usage (waste of bandwith).

JSON Examples :  
[Placename autocomplete](http://www.geonames.org/export/ajax-postalcode-autocomplete.html)   
[full text search on google maps](http://www.geonames.org/maps/json-googlemaps-example.html)

#### Places

##### Cities and Placenames

Webservice Type : REST   
Url : api.geonames.org/citiesJSON?  
Parameters :   
north,south,east,west : coordinates of bounding box   
callback : name of javascript function (optional parameter)   
lang : language of placenames and wikipedia urls (default = en)  
maxRows : maximal number of rows returned (default = 10)  
  
Result : returns a list of cities and placenames in the bounding box, ordered by relevancy (capital/population). Placenames close together are filterered out and only the larger name is included in the resulting list.  
  
Example : <http://api.geonames.org/citiesJSON?north=44.1&south=-9.9&east=-22.4&west=55.2&lang=de&username=demo>   
  
This service is also available in XML output :   
Example : <http://api.geonames.org/cities?north=44.1&south=-9.9&east=-22.4&west=55.2&username=demo>

##### Wikipedia Fulltext Search

Webservice Type : XML or JSON   
Url : api.geonames.org/wikipediaSearch?  
api.geonames.org/wikipediaSearchJSON?  
Parameters : q : place name ([urlencoded utf8](http://forum.geonames.org/gforum/posts/list/8.page))  
title : search in the wikipedia title (optional)  
lang : language code, supported languages are de,en,es,fr,it,nl,pl,pt,ru,zh (default = en)  
maxRows : maximal number of rows returned (default = 10)  
Result : returns the wikipedia entries found for the searchterm as xml document   
Example <http://api.geonames.org/wikipediaSearch?q=london&maxRows=10&username=demo> 

**Example:**

<http://www.geonames.org/export/ws-overview.html>

**JSON**, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML.

**Benefit:**

JSON stands for JavaScript Object Notation, and it's a text format that makes it easy to share data between devices like clients and servers.

Because it is smaller and easier to **convert into a data** **structure**, it's a great alternative to other formats **like XML**.

One of the advantages to using JSON is how easy it is to read. JSON uses minimal formatting--really just a few special characters in addition to the data. Another advantage to JSON is that it's super **easy to parse**.

**This means that JSON data will take less space and load faster into your web applications**. Plus, **parsing an XML object can be complicated and time consuming**, whereas JSON is easily mapped into a JavaScript object and so it takes less time to process.

**REST:**

<http://www.javatpoint.com/soap-vs-rest-web-services>

REST is an **architectural style**.

REST stands for **REpresentational State Transfer**.

REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.

REST **uses URI to expose business logic**.

**JAX-RS** is the java API for RESTful web services.

REST does not define too much standards like SOAP.

REST **requires less bandwidth** and resource than SOAP.

RESTful web services **inherits security measures** from the underlying transport.

REST **permits different** data format such as **Plain text**, **HTML**, **XML**, **JSON** etc.

REST **more preferred** than SOAP.

**REST** describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP). REST does not contain an additional messaging layer and focuses on design rules for creating stateless services. A client can access the resource using the unique [URI](http://searchsoa.techtarget.com/definition/URI) and a representation of the resource is returned. With each new resource representation, the client is said to transfer state. While accessing RESTful resources with HTTP protocol, the URL of the resource serves as the resource identifier and GET, PUT, DELETE, POST and HEAD are the standard HTTP operations to be performed on that resource.

**What is API:**

API stands for Application Programming Interface, i.e. **API is the way for an application to interact with certain system/application/library/etc.**

In layman's terms, I've always said an **API is like a translator between two people who speak different languages. In software,** data can be consumed or distributed using an API (or translator) so that two different kinds of software can communicate. Good software has a strong translator (API) that follows rules and protocols for security and data cleanliness.

**Where it is used**

An example, You are buying an item in online through your credit card. You will provide credit card details and press continue button. It will tell you whether your information is correct or not. To provide these results, there are lot of things in the background.

The application will send your credit card **details to a remote application which will validate your information and send the result back your application. API is used in this scenario**.

**API Vs Web Services:**

Web services are a specific subset of APIs.   
  
APIs can have many forms, whether they communicate over the Web, over other Internet protocols, or even inter-process on the same machine, or inside the same machine and process.

A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may use any style for communication.

**Webservice:**  
A Web Service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network"  
  
Clearly, both are means of communications. The diference is that Web Service almost always involves communication over network and HTTP is the most commonly used protocol. Web service also uses SOAP, REST, and XML-RPC as a means of communication. While an API can use any means of communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.   
So, you can say that-

Web Service. A web service is a collection of APIs working together to perform a particular task.

Web service can be accessed using a transport protocol. HTTP is a far more popular transport protocol to send a request and get a response to and forth from a web service. Using a web service does requires us to be online in the first place.

1.Web Service is an API wrapped in HTTP.  
2. All Web Services are API but APIs are not Web Services.  
3. Web Service might not perform all the operations that an API would perform.  
4. A Web Service needs a network while an API doesn't need a network for its operation.

**Benefit of API Testing :**

**Test for Core Functionality**  
The first major advantage of **API testing is access to the application without a user interface.**

**Time Effective**

Reduced Testing Costs

### Reduced Technical Debt

### Earlier Remediation

## Reduce Risks

### Web Services Example:

**For example: MAP**

<http://maps.googleapis.com/maps/api/staticmap?center=Sydney,NSW&zoom=14&size=400x400&sensor=false>

### 

**Most often-used types of web service:**

* SOAP (Simple Object Access Protocol)
* XML-RPC
* JSON-RPC
* **REST**

### What is REST:

**REST means REpresentational State Transfer; it is an architecture that generally runs over HTTP.** The REST style emphasizes the interactions between clients and services, which are enhanced by having a limited number of operations. **REST is an alternative to SOAP (Simple Object Access Protocol) and instead of using ;XML for request REST uses simple URL in some cases.** Unlike SOAP, RESTFUL applications uses HTTP build in headers to carry meta-information.

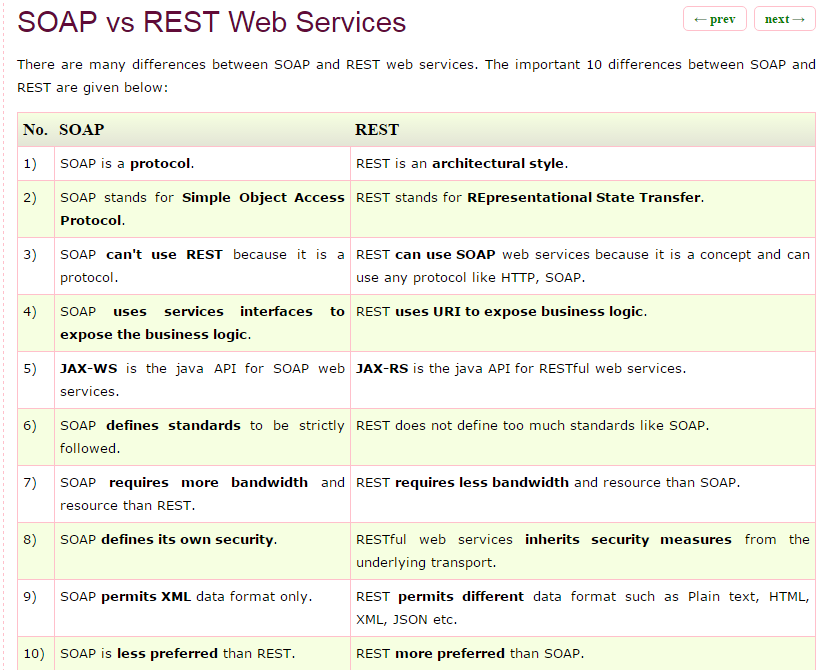
**REST provides a lighter weight alternative. Instead of using XML to make a request**

REST can use four different HTTP 1.1 verbs (GET, POST, PUT, and DELETE) to perform tasks

You can find REST-based Web services that output the data in Command Separated Value (CSV), JavaScript Object Notation (JSON) and Really Simple Syndication (RSS).

REST API supports both XML and JSON format. It is usually preferred for [mobile](http://www.guru99.com/mobile-testing.html) and web apps as it makes app work faster and smoother.

[JSON](http://www.json.org/) (or JavaScript Object Notation) is a lightweight, easy and popular way to exchange data. jQuery is not the only tool for manipulating and interfacing with JSON; it’s just my and many others’ preferred method.

A lot of the services we use everyday have JSON-based APIs: Twitter, Facebook and Flickr all send back data in JSON format.

# Getting Started with Agile Scrum Methodology: The Complete Guide for Software Developers and Testers

**Introduction**:

This is the guide for software developers and testers to understand and start working on the very famous **Agile SCRUM methodology for software development and testing**. Learn the basic but important terminologies used in Agile Scrum process along with a real example of the complete process.

SCRUM is a process in agile methodology which is a combination of Iterative model and incremental model.

One of the major handicaps of the traditional [water-fall model](http://www.softwaretestinghelp.com/what-is-sdlc-waterfall-model/) was that – until the first phase is complete, the application does not move to the other phase. And if by chance there are some changes in the later stage of the cycle, it becomes very challenging to implement those changes, as it would involve revisiting the earlier phases and redoing the changes.

**Some of the key characteristics of SCRUM include:**

* Self-organized and focused team
* No huge requirement documents, rather have very precise and to the point stories.
* Cross functional team works together as a single unit.
* Close communication with the user representative to understand the features.
* Has definite time line of max 1 month.
* Instead of doing the entire “thing” at a time, Scrum does a little of everything at a given interval
* Resources capability and availability are considered before committing any thing.

To understand this methodology well, it’s important to understand the key terminologies in SCRUM.

**[How to Deliver High Value Software Features in a Short Time Period using Agile Scrum Process](http://www.softwaretestinghelp.com/how-to-deliver-high-value-software-features-in-a-short-time-period-using-agile-scrum-process/)**

### Important SCRUM Terminologies:

**1. Scrum Team**

Scrum team is a team comprising of 7 with + or – two members. These members are a mixture of competencies and comprise of developers, testers, data base people, support people etc. along with the product owner and a scrum master. All these members work together in close collaboration for a recursive and definite interval, to develop and implement the said features.

SCRUM team sitting arrangement plays a very important role in their interaction, they never sit in cubicles or cabins; but a huge table.

**2. Sprint**

**Sprint is a predefined interval or the time frame in which the work has to be completed and make it ready for review or ready for production deployment**. This time box usually lies between 2 weeks to 1 month. In our day to day life when we say that we follow 1-month Sprint cycle, it simply means that we work for one month on the tasks and make it ready for review by the end of that month.

**3. Product Owner**

**The product owner is the key stakeholder or the lead user of the application to be developed.**

The product owner is the person who represents the customer side. He/she have the final authority and should always be available for the team. He/she should be reachable in case any one has any doubts that need clarification. It is important for the product owner to understand and not to assign any new requirement in the middle of the sprint or when the sprint has already started.

**4. Scrum Master**

**Scrum Master is the facilitator of the scrum team**. He/she make sure that the scrum team is productive and progressive. In case of any impediments, scrum master follows up and resolves them for the team. SCRUM Master is the mediator between the PO and the team. He / She keeps the PO informed about the progress of the Sprint. If there are any roadblocks or concerns for the team, discuss with the PO and get them resolved. Like team’s Daily Standup, a standup of the SCRUM Master with the PO happens every day.

**Recommended read => [How To Be a Good Team Mentor, Coach and a True Team-Defender in an Agile Testing World?](http://www.softwaretestinghelp.com/how-to-be-a-good-team-mentor-coach-and-a-true-team-defender-in-an-agile-testing-world-the-inspiration/)**

**5. Business Analyst (BA)**

A BA plays a very important role in SCRUM. This person is responsible for getting the requirement finalized and drafted in the requirement docs (based on which the user stories are created). If there are any ambiguities in User Stories / Acceptance criteria, he/she is the one approached by the technical (SCRUM) team who then takes it up to the PO else if possible resolves on his own. In large scale projects there may be more than 1 BA but in small scale projects, the SCRUM Master may be acting as the BA as well. It is a good practice to have a BA when a project kick starts.

**6. User Story**

User stories are nothing but the requirements or feature which has to be implemented. In scrum, we don’t have those huge requirements documents, rather the requirements are defined in a single paragraph, typically having the format as:

As a <User / type of user>  
I want to <Some achievable goal / target>  
To achieve <some result or reason of doing the thing>

**For example:**

As an Admin I want to have a password lock in case user enters incorrect password for consecutive 3 times to restrict unauthorized access.

There are some characteristics of user stories which should be adhered. The user stories should be short, realistic, could be estimated, complete, negotiable and testable. A user story is never altered or changed in the middle of the Sprint. It is the responsibility of the SCRUM Master and the BA (if applicable) to make sure that the PO has drafted the User Stories correct with the proper set of the Acceptance Criteria”. If any changes which will impact the sprint release are to be made, then such stories are pulled out of the sprint or they are done as per the hours available.

Every user story has an acceptance criterion which should be well defined and understood by the team. Acceptance criteria details down the user story provides the supported documents. It helps to further refine the user story. Anybody from the team can write down the acceptance criteria. Testing team base their test cases/conditions on these acceptance criteria.

**7. Epics**

Epics are equivocal user stories or we can say these are the user stories which are not defined and are kept for future sprints. Just try to relate it with life, imagine you are going for a vacation. Since you are going next week, you have everything in place like your hotel bookings, sightseeing, travelers check etc. But what about your vacation plan for next year? You have only a vague idea that you may go to XYZ place, but you have no detailed plan.

An Epic is just like you next year’s vacation plan, where you just know that you may want to go, but where, when, with whom, all these details you have no idea at this point of time.

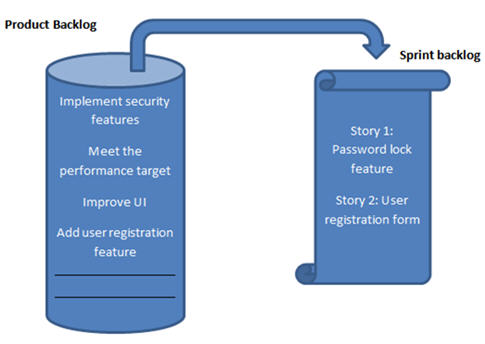
In a similar way, there are features which required to be implemented in future whose details are not yet known. Mostly feature begins with an Epic and then broken down to stories which could be implemented.

**8. Product Backlog**

Product backlog is a kind of bucket or source where all the user stories are kept. This is maintained by the Product owner. Product backlog can be imagined as a wish list of the product owner who prioritizes it as per business needs. During planning meeting (see next section), one user story is taken from the product backlog, the team does the brainstorming, understands it and refine it and collectively decides which user stories to take, with the intervention of the product owner.

**9. Sprint Backlog**

Based on the priority, user stories are taken from the Product Backlog one at a time. The Scrum team brainstorms on it determines the feasibility and decides on the stories to work on a particular sprint. The collective list of all the user stories which the scrum team works on a particular sprint is called s Sprint backlog.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/07/Sprint-Backlog.jpg)

**10. Story Points:**

Story points are quantitative indication of the complexity of a user story. Based on the story point, estimation and efforts for a story are determined. A story point is relative and is not absolute. To make sure that our estimate and efforts are correct, it’s important to check that the user stories are not big. Precise and smaller is the user story, accurate will be the estimation.

Each and every user story is assigned a story point based on the Fibonacci series (1, 2, 3, 5, 8, 13&21). Higher is the number, complex is the story.

**To be precise**

* If you give 1 / 2 / 3 story point it means the story is small and of low complexity.
* If you give points as 5 / 8, it is a medium complex and
* 13 and 21 are highly complex.

Here complexity consists of both development plus testing effort

To decide a story point, brainstorming happens with in the scrum team and the team collectively decides a story point. It may happen that development team gives a story point of 3 to a particular story, because for them it may be 3 lines of code change, but testing team gives 8 story point because they feel this code change will affect larger modules so testing effort would be larger. Whatever story point you are giving, you have to justify it. So in this situation, brainstorming happens and the team collectively agrees to one story point.

Whenever you are deciding on a story point, keep the below factors in mind:

* Dependency of the story with other application/module,
* Skill set of the resource
* Complexity of the story
* Historical learning,
* Acceptance criteria of the user story

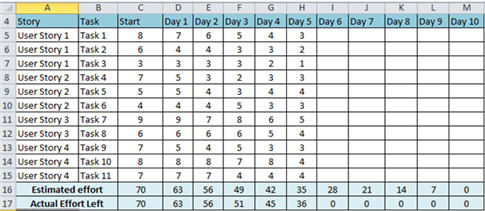
If you are not aware of a particular story, don’t size it.  
Whenever a story is = or > 8 points, it is broken down in 2 or more stories.

**11. Burn down chart**

Burn down chart is a graph which shows the estimated v/s actual effort of the scrum tasks.

It is a tracking mechanism by which for a particular sprint; day to day tasks are tracked to check whether the stories are progressing towards the completion of the committed story points or not.

**Example: To understand this, check the below figure:**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/07/Burn-Down-chart-1.jpg)

I have assumed:

* 2 weeks Sprint ( 10 days)
* 2 resources actual working on the sprint.

“Story”-> Column shows the user stories taken for the sprint.

“Task” -> Column shows the list of task associated with the user story.

“Effort” -> Column shows the effort. Now; this measure is the total effort to complete the task. It does not depict the effort put in by any specific individual.

“Day 1 – Day 10” -> – Column(s) shows the hours which are left to complete the story. Please see that the hour is NOT the hour which is already done BUT the hours which are still left.

“Estimated Effort” -> is the total of the effort. For the “Start” it is simply the sum of the entire individual task: SUM (C5: C15)

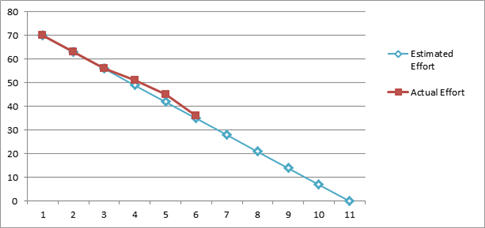
A total number of effort that has to be completed in 1 day is 70 / 10 = 7. So at the end of day 1, the effort should reduce to 70 – 7 = 63. In a similar way, it is calculated for all the days till day 10, when estimated effort should be 0 (Row 16)

“Actual Effort Left” -> as the name suggests, is the effort actually left to complete the story. It may also happen that the actual efforts increases or decreases then the estimated one.

You can use the in build functions and Chart in Excel to create this burn down chart.

**Burn Down Chart steps would be:**

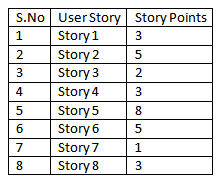
1. Enter all the stories ( Column A5 – A15)
2. Enter all the Tasks ( Column B5 – B15)
3. Enter the Days ( Day 1 – Day 10 )
4. Enter the starting efforts (Sum the tasks C5 – C15 )
5. Apply formula to calculate the “Estimated Efforts” for each day (Day 1 to Day 10). Enter the formula at D15 (C16-$C$16/10) and drag it for all the days.
6. For each day, enter the actual efforts. Enter the formula at D17 (SUM (D5:D15)) for summing up the actual efforts left, and drag it for all the other days.
7. Select it and create the chart as follows:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/07/Burn-Down-chart-2.jpg)

**12. Velocity**

A total number of story point which a scrum team archives in a sprint, is called Velocity. The Scrum team is judged or referenced by its velocity. Having said that, it should be kept in mind that the objective here is NOT achieving the maximum story points, but to have quality deliverable, respecting scrum team’s comfort level.

**For example**: For a particular sprint: total number of user stories are 8 having story points as

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/07/Scrum-Velocity.jpg)

So here the velocity will be the sum of the story points = 30

**Definition of Done:**

A Sprint is marked as Done when all the stories are completed, all dev, research, QA tasks are marked ‘Completed’, all bugs are fixed-closed else the ones that can be done later (like not completely related or are less important) are pulled out and added in the backlog, the code reviews and unit testing is completed, the estimated hours have met the actual hours put up in the tasks and most importantly a successful demo has been given to the PO and the stakeholders.

### Activities done in SCRUM Methodology:

**#1: Planning meeting**

A planning meeting is the starting point of Sprint. It is the meeting where the entire scrum team gather, the SCRUM Master selects a user story based on the priority from the product back log and the team brain storms on it. Based on the discussion, the scrum team decides the complexity of the story and sizes it as per the Fibonacci series. The team identifies the tasks along with the efforts (in hours) which would be done to complete the implementation the user story.

Many a time planning meeting is preceded by a “Pre-Planning meeting”. It’s just like a home work which the scrum team does before they sit for the formal planning meet. The team tries to write down the dependencies or other factors which they would like to discuss in the planning meet.

**#2: Execution of sprint tasks**

As the name suggests, these are the actual work done by the scrum team to accomplish their task and take the user story into the “Done” state.

**#3: Daily Standup**

During the sprint cycle, every day the scrum team meets for, not more than 15 minutes (could be a stand up call, recommended to have during the beginning of the day) and state 3 points:

1. What did the team member did yesterday
2. What did the team member plan to do today
3. Any impediments (roadblocks)

It is the Scrum master who facilitates this meeting. In case, any team member is facing any kind of difficulties, the scrum master follows up to get it resolved. In Standups, the board is also reviewed and it itself shows the progress of the team.

**#4: Review meeting**

At the end of every sprint cycle, the SCRUM team meets again and demonstrates implemented user stories to the product owner. The product owner may cross verify the stories as per its acceptance criteria. It’s again the responsibility of the Scrum master to preside over this meeting. Also in the SCRUM tool, the Sprint is closed and the tasks that are marked done.

**#5: Retrospective meeting**

The retrospective meeting happens after the review meeting. The SCRUM team meets and discusses & document the following points:

* What went well during the Sprint (Best practices)
* What did not went well in the Sprint
* Lessons learnt
* Action Items.

The Scrum team should continue to follow the best practice, ignore the “not best practices” and implement the lessons learned during the consequent sprints. The retrospective meeting helps to implement the continuous improvement of the SCRUM process.

### How the Process is done? An example!

Having read about the technical jargons of SCRUM; let me try to demonstrate the whole process with an example.

**Step #1**: Let’s have a SCRUM team of 9 people comprising of 1 product owner, 1 Scrum master, 2 testers, 4 developers and 1 DBA.

**Step #2**: The Sprint is decided to follow 4 weeks cycle. So we have 1-month Sprint starting 5th June to 4th of July.

**Step #3**: The Product owner has the prioritized list of user stories in the product backlog.

**Step #4:** The team decides to meet on 4th June for the “Pre Planning” meeting.

* The product owner takes 1 story from the product backlog, describes it and leaves it to the team to brainstorm on it.
* The entire team discusses and communicates directly to the product owner to have clear understood of the user story.
* In a similar way, various other user stories are taken. If possible team can go ahead and size the stories as well.

After all the discussion, Individual team member go back to their work stations and

* Identify their individual tasks for each story.
* Calculate the exact number of hours on which they will be working. How the member concludes these hours; let’s check that

Total number of working hours = 9  
Minus 1 hour for break, minus 1 hour for meetings, minus 1 hour for mails, discussions, troubleshooting etc.  
So the actual working hours = 6  
A total number of working days during the Sprint = 21 days.  
Total number of hours available = 21\*6 = 126  
The member is on leave for 2 days = 12 hours (This varies for each member, some may take leave and some may not.)  
Number of actual hours = 126 – 12 = 114 hours.

This means that the member will actually available for 114 hours for this sprint. So he will break down his individual sprint task in such a way that total of 114 hours is reached.

**Step #5**: On 5th of June the entire Scrum team meets for the “Planning Meeting”.

* Final verdict of the user story from the product backlog is done and the story is moved to the Sprint back log.
* For each story, each team member declares their identified tasks, if required can have a discussion on those tasks, can size or resize it (remember the Fibonacci series!!).
* The Scrum master or the team enter their individual tasks along with their hours for each story in a tool.
* After all the stories are completed, Scrum master notes the initial Velocity and formally starts the Sprint.

**Step #6**: Once the Sprint has started, based on the tasks assigned, each team member starts working on those tasks.

**Step #7**: The team meets daily for 15 minutes and discusses 3 things:

* What did they do yesterday?
* What they plan to do today
* Any impediments (roadblocks)?

**Step #8**: The scrum master tracks the progress on daily basis with the help of “Burn down chart”

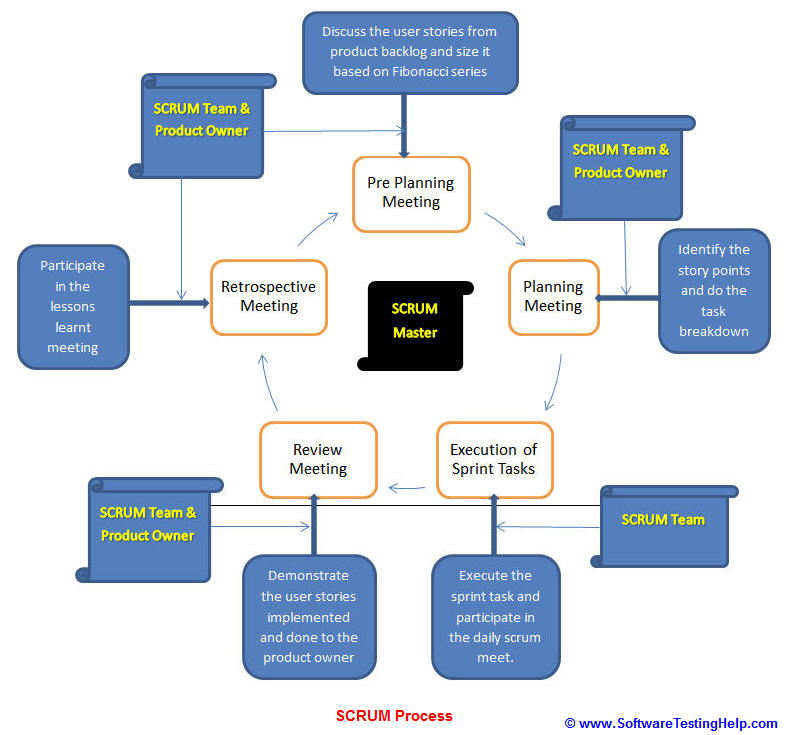
**Step #9**: In case of any impediments, the Scrum master follows up to resolve those.

**Step #10**: On 4th July, the team meets again for the review meeting. A member demonstrates the implemented user story to the product owner.

**Step #11**: On 5th July, Team meets again for the Retrospective, where they discuss

* What went well?
* What did not went well
* Action Items.

**Step #12**: On 6th July, Team again meets for the pre-planning meeting for the next sprint and the cycle continues.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/07/scrum-agile-methodology-process.jpg)

### Tools that can be used for SCRUM activities:

There are many tools available which can be used extensively for tracking the scrum activities. Some of them include:

* [Jira](http://www.softwaretestinghelp.com/atlassian-jira-tutorial-1/)
* [XPlanner](http://xplanner.codehaus.org/" \t "_blank)
* [VersionOne](http://www.versionone.com/" \t "_blank)
* [Sprintometer](http://sprintometer.com/" \t "_blank)
* [ScrumNinja](http://scrumninja.com/scrum-software" \t "_blank)

### Conclusion:

In the beginning, people may face some difficulty to adopt this methodology, but through practice, you will see SCRUM doing wonders. It keeps the resources very focused and you can actually see your application growing.

Many a time’s organization encourages team to have few hours as self-study and development as a scrum task and allocate few hours. During the review, team members also demonstrate what they have studied, and sometimes present any tool or application which they have developed. I personally appreciate this approach as it gives the people a chance to enhance their knowledge and also present their skills. (Well, I have [learnt Selenium](http://www.softwaretestinghelp.org/selenium-training-online-course/) through this approach only! :)

**How to automate cascading drop downs.**

**public** **class** CascadingDropDown\_visibility  {

**public** WebDriver driver;

**public** WebDriverWait wait;

     @Test

**public** **void** cddTest() **throws** Throwable {

*//Explicit wait*

                wait = **new** WebDriverWait(driver, 10);

*//Drop down 1*

                WebElement dd1= driver.findElement(By.id("ctl00\_SampleContent\_DropDownList1"));

*//Drop down 2*

                WebElement dd2= driver.findElement(By.id("ctl00\_SampleContent\_DropDownList2"));

*//Drop down 3*

                WebElement dd3= driver.findElement(By.id("ctl00\_SampleContent\_DropDownList3"));

*//Select a value from Dropdown 1*

**Select s1 = new Select(dd1);**

                s1.selectByVisibleText("Audi");

*//Wait until a option loads in second dropdown*

                wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//select[@i='ctl00\_SampleContent\_DropDownList2']/option[2]")));

              Select s2 = **new** Select(dd2);

                s2.selectByVisibleText("A6");

*//Wait until a option loads in Third dropdown*

                wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//select[@id='ctl00\_SampleContent\_DropDownList3']/option[2]")));

                Select s3 = **new** Select(dd3);

                s3.selectByVisibleText("Cyan");

                Thread.sleep(5000);

        }

        @BeforeTest

**public** **void** beforeTest() {

                driver = **new** FirefoxDriver();

                driver.get("http://www.asp.net/AjaxLibrary/AjaxControlToolkitSampleSite/CascadingDropDown/CascadingDropDown.aspx");

*//driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);*

        }

        @AfterTest

**public** **void** afterTest() {

                driver.quit();

        }

 }

**You have saved some information, how to make sure that it gets saved with the help of Automation**

**Add mysql-connector-java-5.1.21.jar**

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.sql.Connection;

//import com.mysql.jdbc.Connection;

**public** **class** jdbcconnection {

**public** **static** **void** main(String[] args) **throws** SQLException{

// **TODO** Auto-generated method stub

String host="localhost";

String port="3306";

Connection con=DriverManager.*getConnection*("jdbc:mysql://"+host+":"+port+"/employeeportal", "root", "root");

Statement s = con.createStatement();

ResultSet rs =s.executeQuery("select \* from emp");

**while**(rs.next())

{

System.***out***.print(rs.getString("name"));

System.***out***.print(rs.getString("id"));

System.***out***.print(rs.getString("location"));

System.***out***.print(rs.getString("age"));

System.***out***.print("\n");

}

}

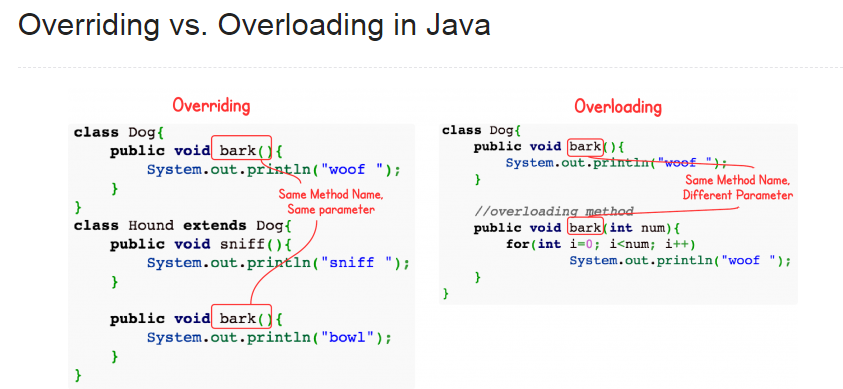
}

**What is the most common exception in Selenium**

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

* NoSuch**Element**Exception
* NoSuch**Window**Exception
* NoSuch**Frame**Exception
* NoSuch**Session**Exception
* NoAlertPresentException
* InvalidSelectorException
* Element**NotVisible**Exception
* Element**NotSelectable**Exception
* **TimeoutException**
* StaleElementReferenceException

**Method overloading and overriding**



* *Overloading* occurs when two or more methods in one class have the same method name but different parameters
* *Overriding* means having two methods with the same method name and parameters (i.e., *method signature*). One of the methods is in the parent class and the other is in the child class. Overriding allows a child class to provide a specific implementation of a method that is already provided its parent class.
* **Polymorphism applies to overriding**, not to overloading.
* **Overriding is a run-time concept** while overloading is a compile-time concept.
* At run-time, a Hound is created and assigned to dog. The JVM knows that dog is referring to the object of Hound, so it calls the bark() method of Hound. This is called **Dynamic Polymorphism**.
* In java, method overloading can't be performed by changing return type of the method only. *Return type can be same or different* in method overloading. But you must have to change the parameter.

class OverloadingExample{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;}

}

//Overridding

class Animal{

void eat(){System.out.println("eating...");}

}

class Dog extends Animal{

void eat(){System.out.println("eating bread...");}

}

**3 ok buttons with same id and other attributes, 5. how you click them**

String cssSelectorOfSameElements="input[type='submit'][id='button']";

List<WebElement> a=driver.findElements(By.cssSelector(cssSelectorOfSameElements)) ;

a.get(0).click();

//a.get(1).click();

//a.get(2).click();

selenium.click("xpath=(//input[@type='checkbox'][position()=2]");

selenium.click("xpath=(//input[@type='checkbox'][position()=4]");

Each time id gets generated differently. So to handle this situation we use some JavaScript functions.

XPath: //button[starts-with(@id, 'continue-')]

Selenium.click(”xpath.(//input[@type=’checkbox’][position()=2”]

**Xpath in Selenium**

**Absolute xpath:**

html/body/div[1]/section/div[1]/div/div/div/div[1]/div/div/div/div/div[3]/div[1]/div/h4[1]/b

**Relative xpath:** //\*[@class='featured-box']//\*[text()='Testing']

**Syntax for XPath**:

Xpath=//tagname[@attribute='value']

**Using XPath Handling complex & Dynamic elements in Selenium**

**1) Basic XPath:**

Xpath=//input[@name='uid']

Xpath=//input[@type='text']

Xpath= //label[@id='message23']

Xpath= //input[@value='RESET']

Xpath=//\*[@class='barone']

Xpath=//a[@href='http://demo.guru99.com/']

Xpath= //img[@src='//cdn.guru99.com/images/home/java.png']

**2) Contains():**

**used when the value of any attribute changes dynamically**

Xpath=//\*[contains(@type,'sub')]

Xpath=//\*[contains(@name,'btn')]

Xpath=//\*[contains(@id,'message')]

Xpath=//\*[contains(text(),'here')]

Xpath=//\*[contains(@href,'guru99.com')]

**3) Using OR & AND:**

Xpath=//\*[@type='submit' or @name='btnReset']

Xpath=//input[@type='submit' and @name='btnLogin']

**4) Start-with function:**

Xpath=//label[starts-with(@id,'message')]

**5) Text():**

Xpath=//td[text()='UserID']

**6) XPath axes methods:**

These XPath axes methods are used to find the complex or dynamic elements. Below we will see some of these methods.

**a) Following:**

Xpath=//\*[@type='text']//following::input

**For particular:**

Xpath=//\*[@type='text']//following::input[1]

**b) Ancestor:**

Xpath=//\*[text()='Enterprise Testing']//ancestor::div

**c) Child:**

Xpath=//\*[@id='java\_technologies']/child::li

**d) Preceding:**

Xpath=//\*[@type='submit']//preceding::input

**e) Following-sibling:**

xpath=//\*[@type='submit']//following-sibling::input

**f) Parent:**

Xpath=//\*[@id='rt-feature']//parent::div

**g) Self:**

Xpath =//\*[@type='password']//self::input

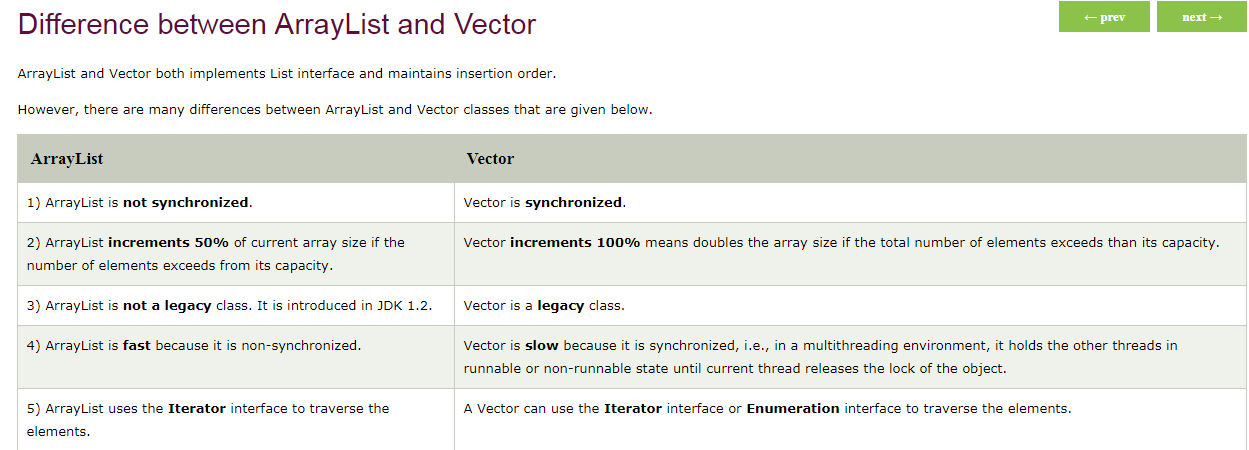
**h) Descendant:**

Xpath=//\*[@id='rt-feature']//descendant::a

**XPath Axes are the methods used to find dynamic elements, which otherwise not possible to find by normal XPath method**

**XPath expression select nodes or list of nodes on the basis of attributes like ID , Name, Classname, etc. from the XML document** .

**ArrayList and Vector**



### Example of Java ArrayList

Let's see a simple example where we are using ArrayList to store and traverse the elements.

**import** java.util.\*;

**class** TestArrayList21{

**public** **static** **void** main(String args[]){

  List<String> al=**new** ArrayList<String>();//creating arraylist

  al.add("Sonoo");//adding object in arraylist

  al.add("Michael");

  al.add("James");

  al.add("Andy");

  //traversing elements using Iterator

  Iterator itr=al.iterator();

**while**(itr.hasNext()){

   System.out.println(itr.next());

  }     }    }

### Example of Java Vector

Let's see a simple example of a Java Vector class that uses the Enumeration interface.

import java.util.\*;

class TestVector1{

public static void main(String args[]){

Vector<String> v=new Vector<String>();//creating vector

v.add("umesh");//method of Collection

v.addElement("irfan");//method of Vector

v.addElement("kumar");

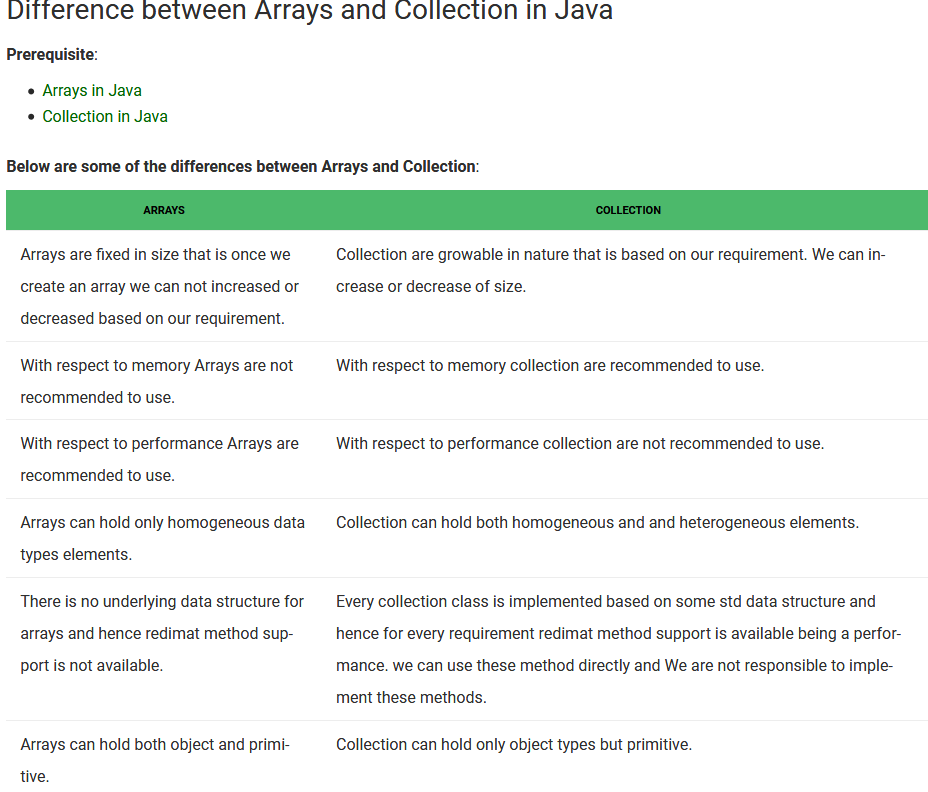
//traversing elements using Enumeration

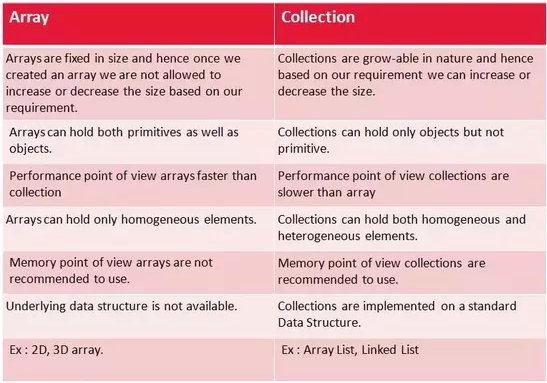
Enumeration e=v.elements();

while(e.hasMoreElements()){

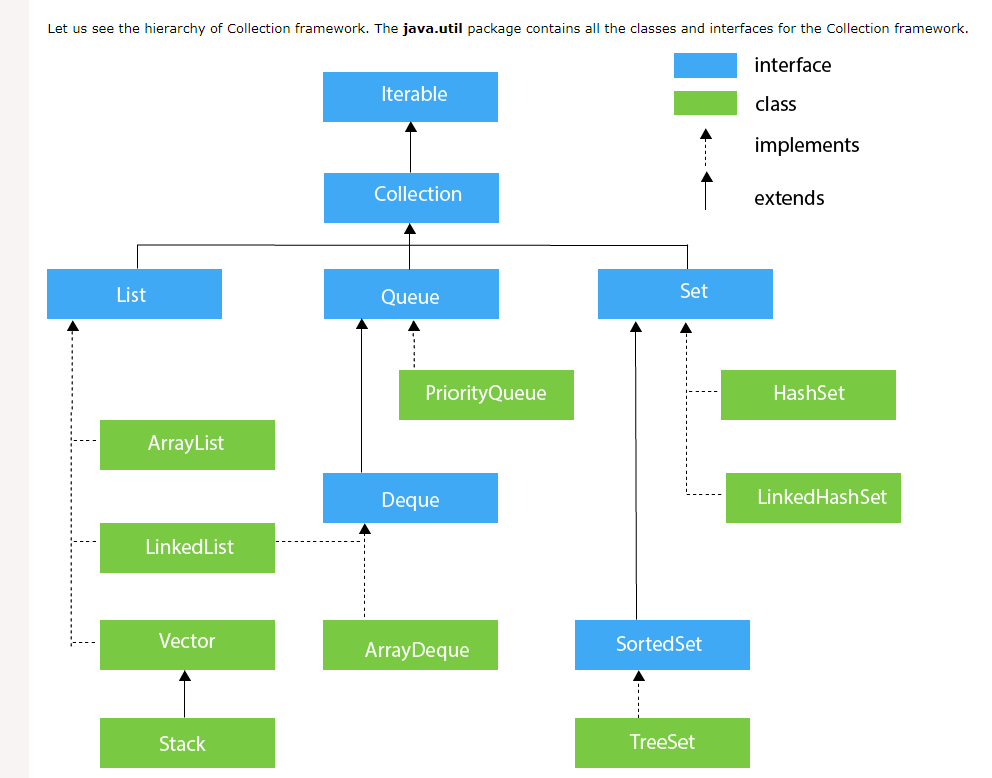
System.out.println(e.nextElement());  }}}

**Arrays and Collection**





**COLLECTIONS:**



The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

### Collections class:

A Collection is a group of individual objects represented as a single unit. Java provides Collection Framework which defines several classes and interfaces to represent a group of objects as a single unit.

The Collection interface (java.util.Collection) and Map interface (java.util.Map) are the two main “root” interfaces of Java collection classes.

Need for Collection Framework : Before Collection Framework (or before JDK 1.2) was introduced, the standard methods for grouping Java objects (or collections) were Arrays or Vectors or Hashtables. All of these collections had no common interface.

Accessing elements of these Data Structures was a hassle as each had a different method (and syntax) for accessing its members:

### Iterator interface

### Iterator interface provides the facility of iterating the elements in a forward direction only.

## Iterable Interface

### The Iterable interface is the root interface for all the collection classes. The Collection interface extends the Iterable interface and therefore all the subclasses of Collection interface also implement the Iterable interface.

## Collection Interface

The Collection interface is the interface which is implemented by all the **classes** in the collection framework. It declares the methods that every collection will have. In other words, we can say that the Collection interface builds the foundation on which the collection framework depends.

### Some of the methods of Collection interface are Boolean add ( Object obj), Boolean addAll ( Collection c), void clear(), etc. which are implemented by all the subclasses of Collection interface.

## List Interface

List interface is the child interface of Collection interface. It inhibits a list type data structure in which we can store the **ordered** collection of objects. It can have **duplicate** values.

List interface is implemented by the **classes ArrayList, LinkedList, Vector, and Stack**.

## ArrayList

The ArrayList class implements the List interface. It uses a dynamic array to store the **duplicate** element of different data types. The ArrayList class maintains the insertion order and is **non-synchronized**. The elements stored in the ArrayList class can be randomly accessed.

## LinkedList

LinkedList implements the Collection interface. It uses a doubly linked list internally to store the elements. It can store the **duplicate** elements. It maintains the insertion order and is **not synchronized**. In LinkedList, the manipulation is **fast** because no shifting is required.

## Vector

Vector uses a dynamic array to store the data elements. It is similar to ArrayList. However, It is **synchronized** and contains many methods that are not the part of Collection framework.

## Stack

The stack is the subclass of Vector. It implements the last-in-first-out data structure, i.e., Stack. The stack contains all of the methods of Vector class and also provides its methods like boolean push(), boolean peek(), boolean push(object o), which defines its properties.

## Queue Interface

Queue interface maintains the first-in-first-out order. It can be defined as an **ordered** list that is used to hold the elements which are about to be processed. There are various classes like PriorityQueue, Deque, and ArrayDeque which implements the Queue interface.

## PriorityQueue

The PriorityQueue class implements the Queue interface. It holds the elements or objects which are to be processed by their priorities. PriorityQueue doesn't allow **null** values to be stored in the queue.

## Deque Interface

Deque interface extends the Queue interface. In Deque, we can remove and add the elements from both the side. Deque stands for a double-ended queue which enables us to perform the operations at both the ends.

## ArrayDeque

ArrayDeque class implements the Deque interface. It facilitates us to use the Deque. Unlike queue, we can add or delete the elements from both the ends.

ArrayDeque is faster than ArrayList and Stack and has no capacity restrictions.

## Set Interface

Set Interface in Java is present in java.util package. It extends the Collection interface. It represents the **unordered** set of elements which **doesn't allow** us to store the **duplicate** items. We can store at most **one** **null** value in Set. Set is implemented by HashSet, LinkedHashSet, and TreeSet.

## HashSet

HashSet class implements Set Interface. It represents the collection that uses a **hash table for storage**. Hashing is used to store the elements in the **HashSet**. It contains **unique** items.

## LinkedHashSet

LinkedHashSet class represents the LinkedList implementation of Set Interface. It extends the HashSet class and implements Set interface. Like HashSet, It also contains **unique** elements. It maintains the insertion order and permits null elements.

## SortedSet Interface

SortedSet is the alternate of Set interface that provides a total ordering on its elements. The elements of the SortedSet are arranged in the **increasing** (ascending) **order**. The SortedSet provides the additional methods that inhibit the natural ordering of the elements.

## TreeSet

Java TreeSet class implements the Set interface that uses a **tree for storage**. Like HashSet, TreeSet also contains **unique** elements. However, the access and **retrieval** time of TreeSet is quite **fast**. The elements in TreeSet stored in **ascending order**.

## Map: A Map is useful if you have to search, update or delete elements on the basis of a key. There are two interfaces for implementing Map in java: Map and SortedMap, and three classes: HashMap, LinkedHashMap, and TreeMap. A Map doesn't allow duplicate keys, but you can have duplicate values. HashMap and LinkedHashMap allow null keys and values, but TreeMap doesn't allow any null key or value.

[**HashMap**](https://www.javatpoint.com/java-hashmap)**:** HashMap is the implementation of Map, but it **doesn't** maintain any **order**. It is a part of Java’s collection since Java 1.2. It provides the basic implementation of Map interface of Java. It stores the data in (**Key, Value**) pairs. To access a value one must know its key. HashMap is known as HashMap because it uses a technique called **Hashing**.[Hashing](http://quiz.geeksforgeeks.org/hashing-set-1-introduction/) is a technique of converting a large String to small String that represents the same String. A shorter value helps in indexing and faster searches. HashSet also uses HashMap internally. It internally uses a link list to store key-value pairs already explained in [HashSet](https://www.geeksforgeeks.org/hashset-in-java/" \t "_blank) in detail and further articles.

HashMap is a part of java.util package. HashMap extends an abstract class AbstractMap which also provides an incomplete implementation of Map interface.

It also implements [Cloneable](https://docs.oracle.com/javase/7/docs/api/java/lang/Cloneable.html) and [Serializable](https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html) interface. K and V in the above definition represent Key and Value respectively. **HashMap** **doesn’t** allow **duplicate** **keys** but **allows duplicate values**. That means A single key can’t contain more than 1 value but more than 1 key can contain a single value.

**HashMap** allows **null key** also but only **once** and **multiple null values**.This class makes no guarantees as to the order of the map; in particular, it does not guarantee that the order will remain constant over time. It is roughly **similar** to **HashTable** but is **unsynchronized**.

[**LinkedHashMap**](https://www.javatpoint.com/java-linkedhashmap)**:** LinkedHashMap is the implementation of Map. It **inherits** **HashMap** **class**. It maintains **insertion order.**

[**TreeMap**](https://www.javatpoint.com/java-treemap)**:** TreeMap is the **implementation** of **Map** and **SortedMap**. It **maintains** **ascending** **order**.

**Hashtable:** This class **implements** a **hash** **table**, which maps keys to values. Any **non-null object** can be used as a key or as a value.  
To successfully store and retrieve objects from a hashtable, the objects used as keys must implement the hashCode method and the eq**u**als method.

* It is similar to **HashMap**, but is **synchronised**.
* Hashtable **stores** **key/value** pair **in hash table**.
* In Hashtable we specify an object that is used as a key, and the value we want to associate to that key. The key is then hashed, and the resulting **hash** **code** is used as the **index** at which the value is stored within the table.

……………………………………………………………………………………………………………..

#### 1) What is the Java Collection framework? List down its advantages?

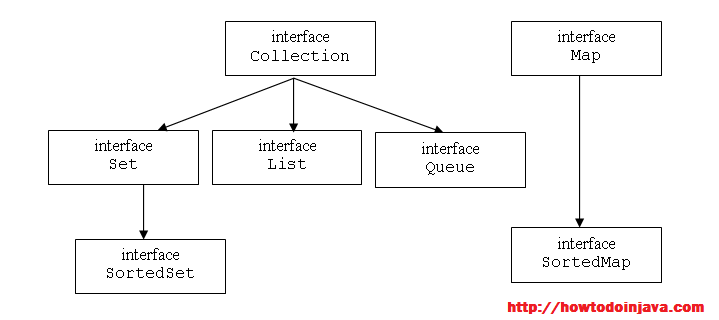
By definition, a collection is **an object that represents a group of objects**. Like in set theory, a set is group of elements. Easy enough !!

Prior to JDK 1.2, JDK has some utility classes such as Vector and HashTable, but there was no concept of Collection framework. Later from JDK 1.2 onwards, JDK felt the need of having a consistent support for reusable data structures. Finally, the collections framework was designed and developed primarily by Joshua Bloch, and was **introduced in JDK 1.2**.

Its most noticeable **benefits of java collections** can be listed as:

* Reduced programming effort due to ready to use code
* Increased performance because of high-performance implementations of data structures and algorithms
* Provides interoperability between unrelated APIs by establishing a common language to pass collections back and forth
* Easy to learn APIs by learning only some top level interfaces and supported operations

#### 2) Explain Collection’s hierarchy?

Java Collection Hierarchy

As shown in above image, collection framework has one interface at top i.e. **Collection**. It is **extended by Set, List and Queue interfaces**. Then there are loads of other classes in these 3 branches which we will learn in following questions.

Remember the signature of Collection interface. It will help you in many question.

|  |
| --- |
| public interface Collection extends Iterable {  //method definitions  } |

Framework also consist of Map interface, which is part of collection framework. but it does not extend Collection interface. We will see the reason in 4th question in this question bank.

#### 3) Why Collection interface does not extend Cloneable and Serializable interface?

Well, simplest answer is “**there is no need to do it**“. Extending an interface simply means that you are creating a subtype of interface, in other words a more specialized behavior and Collection interface is not expected to do what Cloneable and Serializable interfaces do.

Another reason is that not everybody will have a reason to have Cloneable collection because if it has very large data, then every **unnecessary clone operation will consume a big memory**. Beginners might use it without knowing the consequences.

Another reason is that **Cloneable and Serializable are very specialized behavior** and so should be implemented only when required. For example, many concrete classes in collection implement these interfaces. So if you want this feature. use these collection classes otherwise use their alternative classes.

#### 4) Why Map interface does not extend Collection interface?

A good answer to this interview question is “**because they are incompatible**“. Collection has a method add(Object o). Map can not have such method because it need key-value pair. There are other reasons also such as Map supports keySet, valueSet etc. Collection classes does not have such views.

Due to such big differences, Collection interface was not used in Map interface, and it was build in separate hierarchy.

## – List interface questions

#### 5) Why we use List interface? What are main classes implementing List interface?

A java list is a **“ordered” collection of elements**. This ordering is a **zero based index**. It does not care about duplicates. Apart from methods defined in Collection interface, it does **have its own methods** also which are largely to manipulate the collection **based on index location of element**. These methods can be grouped as search, get, iteration and range view. All above operations support index locations.

The main classes implementing List interface are: **Stack, Vector, ArrayList and LinkedList**. Read more about them in java documentation.

#### 6) How to convert an array of String to arraylist?

This is more of a programmatic question which is seen at beginner level. The intent is to check the knowledge of applicant in Collection utility classes. For now, lets learn that there are two utility classes in Collection framework which are mostly seen in interviews i.e. **Collections and Arrays**.

Collections class provides some static functions to perform specific operations on collection types. And Arrays provide utility functions to be performed on array types.

|  |
| --- |
| //String array  String[] words = {"ace", "boom", "crew", "dog", "eon"};  //Use Arrays utility class  List wordList = Arrays.asList(words);  //Now you can iterate over the list |

Please not that this function is not specific to String class, it will return List of element of any type, of which the array is. e.g.

|  |
| --- |
| //String array  Integer[] nums = {1,2,3,4};  //Use Arrays utility class  List numsList = Arrays.asList(nums); |

#### 7) How to reverse the list?

This question is just like above to test your knowledge of **Collections** utility class. Use it **reverse()** method to reverse the list.

|  |
| --- |
| Collections.reverse(list); |

## – Set interface questions

#### 8) Why we use Set interface? What are main classes implementing Set interface?

It **models the mathematical set in set theory**. Set interface is like List interface but with some differences. First, it is **not ordered collection**. So no ordering is preserved while adding or removing elements. The main feature it does provide is “**uniqueness of elements**“. It does not support duplicate elements.

Set also adds a stronger contract on the behavior of the equals and hashCode operations, allowing Set instances to be compared meaningfully even if their implementation types differ. Two Set instances are equal if they contain the same elements.

Based on above reasons, it **does not have operations based on indexes of elements like List**. It only has methods which are inherited by Collection interface.

Main classes implementing Set interface are :**EnumSet, HashSet, LinkedHashSet, TreeSet**. Read more on related java documentation.

#### 9) How HashSet store elements?

You must know that HashMap store key-value pairs, with one condition i.e. keys will be unique. HashSet uses Map’s this feature to ensure uniqueness of elements. In HashSet class, a map declaration is as below:

|  |
| --- |
| private transient HashMap<E,Object> map;    //This is added as value for each key  private static final Object PRESENT = new Object(); |

So **when you store a element in HashSet, it stores the element as key in map and “PRESENT” object as value**. (See declaration above).

|  |
| --- |
| public boolean add(E e) {  return map.put(e, PRESENT)==null;  } |

I will highly suggest you to read this post: [**How HashMap works in java?**](https://howtodoinjava.com/java/collections/how-hashmap-works-in-java/) This post will help you in answering all the HashMap related questions very easily.

#### 10) Can a null element added to a TreeSet or HashSet?

As you see, There is no null check in add() method in previous question. And HashMap also allows one null key, so **one “null” is allowed in HashSet**.

TreeSet uses the same concept as HashSet for internal logic, but uses NavigableMap for storing the elements.

|  |
| --- |
| private transient NavigableMap<E,Object> m;    // Dummy value to associate with an Object in the backing Map  private static final Object PRESENT = new Object(); |

NavigableMap is subtype of SortedMap which does not allow null keys. So essentially,**TreeSet also does not support null keys**. It will throw NullPointerException if you try to add null element in TreeSet.

## – Map interface questions

#### 11) Why we use Map interface? What are main classes implementing Map interface?

Map interface is a special type of collection which is **used to store key-value pairs**. It does not extend Collection interface for this reason. This interface provides methods to add, remove, search or iterate over various views of Map.

Main classes implementing Map interface are:**HashMap, Hashtable, EnumMap, IdentityHashMap, LinkedHashMap and Properties.**

#### 12) What are IdentityHashMap and WeakHashMap?

**IdentityHashMap** is similar to HashMap except that**it uses reference equality when comparing elements**. IdentityHashMap class is not a widely used Map implementation. While this class implements the Map interface, it intentionally violates Map’s general contract, which mandates the use of the equals() method when comparing objects. IdentityHashMap is designed for use only in the rare cases wherein reference-equality semantics are required.

**WeakHashMap** is an implementation of the Map interface **that stores only weak references to its keys**. Storing only weak references allows a key-value pair to be garbage collected when its key is no longer referenced outside of the WeakHashMap. This class is intended primarily for use with key objects whose equals methods test for object identity using the == operator. Once such a key is discarded it can never be recreated, so it is impossible to do a look-up of that key in a WeakHashMap at some later time and be surprised that its entry has been removed.

#### 13) Explain ConcurrentHashMap? How it works?

Taking from java docs:

**A hash table supporting full concurrency of retrievals and adjustable expected concurrency for updates**. This class obeys the same functional specification as Hashtable, and includes versions of methods corresponding to each method of Hashtable. However, even though all operations are thread-safe, retrieval operations do not entail locking, and there is not any support for locking the entire table in a way that prevents all access. This class is fully interoperable with Hashtable in programs that rely on its thread safety but not on its synchronization details.

#### 14) How hashmap works?

The **most important question** which is most likely to be seen in every level of job interviews. You must be very clear on this topic., not only because it is most asked question but also it will open up your mind in further questions related to collection APIs.

Answer to this question is very large and you should read it my post: [**How HashMap works?**](https://howtodoinjava.com/java/collections/how-hashmap-works-in-java/) For now, lets remember that HashMap works **on principle of Hashing**. A map by definition is : “An object that maps keys to values”. To store such structure, **it uses an inner class Entry**:

|  |
| --- |
| static class Entry implements Map.Entry  {  final K key;  V value;  Entry next;  final int hash;  ...//More code goes here  } |

Here key and value variables are used to store key-value pairs. Whole entry object is stored in an array.

|  |
| --- |
| /\*\*  \* The table, re-sized as necessary. Length MUST Always be a power of two.  \*/  transient Entry[] table; |

The index of array is calculated on basis on hashcode of Key object. Read more of linked topic.

#### 15) How to design a good key for hashmap?

Another good question usually followed up after answering how hashmap works. Well, the most important constraint is **you must be able to fetch the value object back in future**. Otherwise, there is no use of having such a data structure. If you understand the working of hashmap, you will find it largely depends on hashCode() and equals() method of Key objects.

So a good key object**must provide same hashCode() again and again**, no matter how many times it is fetched. Similarly, same keys**must return true when compare with equals() method and different keys must return false**.

For this reason,**immutable classes are considered best candidate for HashMap keys**.

#### 16) What are different Collection views provided by Map interface?

Map interface provides 3 views of key-values pairs stored in it:

* key set view
* value set view
* entry set view

All the views can be navigated using iterators.

#### 17) When to use HashMap or TreeMap?

HashMap is well known class and all of us know that. So, I will leave this part by saying that it is used to store key-value pairs and allows to perform many operations on such collection of pairs.

TreeMap is special form of HashMap. **It maintains the ordering of keys** which is missing in HashMap class. This ordering is **by default “natural ordering”**. The default ordering can be override by providing an instance of Comparator class, whose compare method will be used to maintain ordering of keys.

Please note that **all keys inserted into the map must implement the Comparable interface** (this is necessary to decide the ordering). Furthermore, all such keys must be mutually comparable: k1.compareTo(k2) must not throw a ClassCastException for any keys k1 and k2 in the map. If the user attempts to put a key into the map that violates this constraint (for example, the user attempts to put a string key into a map whose keys are integers), the put(Object key, Object value) call will throw a ClassCastException.

## 

#### 18) Difference between Set and List?

The most noticeable differences are :

* Set is **unordered** collection where List is **ordered** collection based on zero based index.
* List allow **duplicate** elements but Set does not allow duplicates.
* List does not prevent inserting null elements (as many you like), but Set will allow only **one null** element.

#### 19) Difference between List and Map?

Perhaps most easy question. **List is collection of elements where as map is collection of key-value pairs**. There is actually lots of differences which originate from first statement. They have**separate top level interface, separate set of generic methods, different supported methods and different views of collection**.

I will take much time hear as answer to this question is enough as first difference only.

#### 20) Difference between HashMap and HashTable?

There are several differences between HashMap and Hashtable in Java:

* Hashtable is **synchronized**, whereas HashMap is not.
* Hashtable does **not allow null** keys or values. HashMap allows **one null key** and any number of null values.
* The third significant difference between HashMap vs Hashtable is that Iterator in the HashMap is a fail-fast iterator while the enumerator for the Hashtable is not.

#### 21) Difference between Vector and ArrayList?

Lets note down the differences:

* All the methods of Vector is **synchronized**. But, the methods of ArrayList is not synchronized.
* Vector is a Legacy class added in first release of JDK. ArrayList was part of JDK 1.2, when collection framework was introduced in java.
* By default, Vector doubles the size of its array when it is re-sized internally. But, ArrayList increases by half of its size when it is re-sized.

#### 22) Difference between Iterator and Enumeration?

Iterators differ from enumerations in three ways:

* Iterators allow the caller to remove elements from the underlying collection during the iteration with its remove() method. You can not add/remove elements from a collection when using enumerator.
* Enumeration is **available in legacy classes** i.e Vector/Stack etc. whereas **Iterator** is available in **all modern collection** **classes**.
* Another minor difference is that Iterator has improved method names e.g. Enumeration.hasMoreElement() has become Iterator.hasNext(), Enumeration.nextElement() has become Iterator.next() etc.

#### 23) Difference between HashMap and HashSet?

**HashMap** is **collection of key-value pairs** whereas **HashSet** is **un-ordered collection of unique elements**. That’s it. No need to describe further.

#### 24) Difference between Iterator and ListIterator?

There are three Differences are there:

* We can use Iterator to traverse Set and List and also Map type of Objects. But List Iterator can be used to traverse for List type Objects, but not for Set type of Objects.
* By using Iterator we can retrieve the elements from Collection Object in forward direction only whereas **List Iterator**, which allows you to **traverse** in **either** **directions** using hasPrevious() and previous() methods.
* **ListIterator** allows you **modify** the list using add() remove() methods. Using **Iterator** you can not add, only remove the elements.

#### 25) Difference between TreeSet and SortedSet?

SortedSet is an **interface** which TreeSet implements. That’ it !!

#### 26) Difference between ArrayList and LinkedList?

* LinkedList store elements within a doubly-linked list data structure. ArrayList store elements within a dynamically resizing array.
* LinkedList allows for constant-time insertions or removals, but only sequential access of elements. In other words, you can walk the list forwards or backwards, but grabbing an element in the middle takes time proportional to the size of the list. **ArrayLists**, on the other hand, allow **random access**, so you can grab any element in constant time. But adding or removing from anywhere but the end requires shifting all the latter elements over, either to make an opening or fill the gap.
* LinkedList has more memory overhead than ArrayList because in ArrayList each index only holds actual object (data) but in case of LinkedList each node holds both data and address of next and previous node.

#### 27) How to make a collection read only?

Use following methods:

* Collections.unmodifiableList(list);
* Collections.unmodifiableSet(set);
* Collections.unmodifiableMap(map);

These methods takes collection parameter and return a new read-only collection with same elements as in original collection.

#### 28) How to make a collection thread safe?

Use below methods:

* Collections.synchronizedList(list);
* Collections.synchronizedSet(set);
* Collections.synchronizedMap(map);

Above methods take collection as parameter and return same type of collection which are synchronized and thread safe.

#### 29) Why there is not method like Iterator.add() to add elements to the collection?

The sole purpose of an Iterator is to enumerate through a collection. All collections contain the add() method to serve your purpose. There would be no point in adding to an Iterator because the **collection may or may not be ordered**. And **add() method can not have same implementation for ordered and unordered collections**.

#### 30) What are different ways to iterate over a list?

You can iterate over a list using following ways:

* Iterator loop
* For loop
* For loop (Advance)
* While loop

#### 31) What do you understand by iterator fail-fast property?

**Fail-fast Iterators fail as soon as they realized that structure of Collection has been changed since iteration has begun**. Structural changes means adding, removing or updating any element from collection while one thread is Iterating over that collection.

Fail-fast behavior is implemented by keeping a modification count and if iteration thread realizes the change in modification count it throws **ConcurrentModificationException**.

#### 32) What is difference between fail-fast and fail-safe?

You have understood fail-fast in previous question. **Fail-safe iterators** are just opposite to fail-fast. **They never fail if you modify the underlying collection on which they are iterating**, because they work on **clone** of Collection instead of original collection and that’s why they are called as **fail-safe iterator**.

Iterator of **CopyOnWriteArrayList** is an example of fail-safe Iterator also iterator written by ConcurrentHashMap keySet is also fail-safe iterator and never throw ConcurrentModificationException.

#### 33) How to avoid ConcurrentModificationException while iterating a collection?

You should first try to **find another alternative iterator which are fail-safe**. For example if you are using List and you can use ListIterator. If it is legacy collection, you can use enumeration.

If above options are not possible then you can use one of three changes:

* If you are using JDK1.5 or higher then you can use ConcurrentHashMap and CopyOnWriteArrayList classes. It is the recommended approach.
* You can convert the list to an array and then iterate on the array.
* You can lock the list while iterating by putting it in a synchronized block.

Please note that last two approaches will cause a performance hit.

#### 34) What is UnsupportedOperationException?

This exception is thrown **on invoked methods which are not supported by actual collection type**.

For example, if you make a read-only list list using “Collections.unmodifiableList(list)” and then call add() or remove() method, what should happen. It should clearly throw UnsupportedOperationException.

#### 35) Which collection classes provide random access of it’s elements?

ArrayList, HashMap, TreeMap, Hashtable classes provide random access to it’s elements.

#### 36) What is BlockingQueue?

**A Queue that additionally supports operations that wait for the queue to become non-empty when retrieving an element, and wait for space to become available in the queue when storing an element.**

BlockingQueue methods come in four forms: one throws an exception, the second returns a special value (either null or false, depending on the operation), the third blocks the current thread indefinitely until the operation can succeed, and the fourth blocks for only a given maximum time limit before giving up.

#### 37) What is Queue and Stack, list down their differences?

**A collection designed for holding elements prior to processing.** Besides basic Collection operations, queues provide additional insertion, extraction, and inspection operations.  
**Queues typically, but do not necessarily, order elements in a FIFO (first-in-first-out) manner.**

**Stack is also a form of Queue but one difference, it is LIFO (last-in-first-out).**

Whatever the ordering used, the head of the queue is that element which would be removed by a call to remove() or poll(). Also note that Stack and Vector are both synchronized.

**Usage:** Use a queue if you want to process a stream of incoming items in the order that they are received.Good for work lists and handling requests.  
Use a stack if you want to push and pop from the top of the stack only. Good for recursive algorithms.

#### 38) What is Comparable and Comparator interface?

In java. all collection which have feature of automatic sorting, uses compare methods to ensure the correct sorting of elements. For example classes which use sorting are TreeSet, TreeMap etc.

**To sort the data elements a class needs to implement Comparator or Comparable interface**. That’s why all Wrapper classes like Integer,Double and String class implements Comparable interface.

**Comparable helps in preserving default natural sorting, whereas Comparator helps in sorting the elements in some special required sorting pattern.** The instance of comparator if passed usually as collection’s constructor argument in supporting collections.

#### 39) What are Collections and Arrays classes?

**Collections and Arrays classes are special utility classes to support collection framework core classes.** They provide utility functions to get read-only/ synchronized collections, sort the collection on various ways etc.

Arrays also helps array of objects to convert in collection objects. Arrays also have some functions which helps in copying or working in part of array objects.

…………………………………………………………………………………………………………

**Explain Try Catch as per your code**

**Try:**

The "try" keyword is used to specify a block where we should **place exception code**. The try block must be **followed by either catch or finally**. It means, we can't use try block alone.

**Catch:**

The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.

**public** **class** JavaExceptionExample{

**public** **static** **void** main(String args[]){

**try**{

      //code that may raise exception

**int** data=100/0;

   }**catch**(ArithmeticException e){System.out.println(e);}

   //rest code of the program

   System.out.println("rest of the code...");

  }

}

**Exception Handling in Java**

The **Exception Handling in Java** is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.

Exception is an abnormal condition. An exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

…………………………………………………………………………………………………………………………………….